

reledmac

Typeset scholarly editions with L^AT_EX*

Maïeul Rouquette[†]

based on the original ledmac by

Peter Wilson

Herries Press

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

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

Abstract

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

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

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

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

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

Contents

1 Introduction	11
1.1 Aim of the package	11
1.2 History	12
1.2.1 edmac	12
1.2.2 ledmac	13
1.2.3 eledmac	14

*This file (**reledmac.dtx**) has version number v2.20.0, last revised 2017/05/04.

[†]maieul at maieul dot net

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

6.3.2 Notes about input encoding with UTF-8 processor	28
6.3.3 Use with <code>\lemma</code> command	29
6.3.4 Customizing	31
6.4 Apparatus of Manuscripts	31
6.4.1 Marking sections of text	32
6.4.2 Layout of the apparatus of manuscripts	32
6.4.3 Settings	33
6.5 Familiar notes	33
6.5.1 Basic use	33
6.5.2 Customizing mark	33
6.5.3 Separator for multiple footnotes	33
6.6 Printing the footnote mark without printing the footnote text	34
6.7 Changing series	34
6.7.1 Create a new series	34
6.7.2 Delete series	34
6.7.3 Series order	34
6.8 Position of critical and familiar footnotes	35
7 Critical apparatus appearance	35
7.1 Notes arrangement in a series	36
7.2 Control line number printing	37
7.2.1 Print line number only at first time	37
7.2.2 Print page number only at first time	37
7.2.3 Arbitrary text before line number	38
7.2.4 Separator for line range	38
7.2.5 Abbreviate line range	38
7.2.6 Disable line number	39
7.2.7 Printing pstart number	39
7.2.8 Printing stanza number	39
7.2.9 Separator between line and subline numbers	39
7.2.10 Separator between page and line numbers	40
7.2.11 Space around number	40
7.2.12 Space around line symbol	40
7.2.13 Space in place of number	40
7.2.14 Boxing line number and line symbol	40
7.3 For endnotes	41
7.4 Arbitrary code around line number	41
7.5 Separator between the lemma and the note	42
7.5.1 For footnotes	42
7.5.2 For endnotes	42
7.6 Font style	43
7.6.1 For line number	43
7.6.2 For the lemma	43
7.6.3 For all notes	43
7.7 Wrapping notes	43
7.7.1 Wrapping lemmas	43

7.7.2 Wrapping contents	44
7.8 Indent of notes content	44
7.9 Arbitrary code at the beginning of notes	44
7.10 Arbitrary code before inserting note	44
7.11 Options for footnotes in columns	45
7.11.1 Alignment	45
7.11.2 Size of the columns	45
7.12 Options for paragraphed footnotes and notes grouped by line	45
7.12.1 Mark separation of notes	45
7.12.2 Ragged text	46
7.13 Options for block of notes	46
7.13.1 Grouping notes by line	46
7.13.2 Text before notes	46
7.13.3 Code before notes	46
7.13.4 Spacing	46
7.13.5 Rule	47
7.13.6 Maximum height	47
7.13.7 Width	48
7.14 Footnotes and the reledpar columns	48
7.15 Endnotes in one paragraph	48
8 Fonts	48
9 Verse	49
9.1 Basic	49
9.2 Define stanza indents	49
9.3 Repeating stanza indents	50
9.4 Manual stanza indent	50
9.5 Stanza breaking	50
9.6 Hanging symbol	51
9.7 Long verse and page break	51
9.8 Content before/after verses	51
9.9 Numbering stanza	52
9.10 Various tools	53
9.11 Notes on empty lines	53
10 Grouping	53
11 Cross referencing	54
11.1 Basic use	54
11.2 Cross-referencing to a critical note	54
11.3 Cross-referencing which return a number in any case	54
11.3.1 Cross-referencing in order to define line number of a critical note	55
11.4 Not automatic cross-referencing	55
11.5 Normal L ^A T _E X cross-referencing	56
11.6 References to start and end lines	56

11.6.1 Reference to main text lines	56
11.6.2 References to lines that are commented on in the apparatus	56
11.6.3 Settings	56
11.7 Compatibility with <code>xr</code> package	58
12 Side notes	58
12.1 Basics	58
12.2 Setting	59
12.2.1 Width	59
12.2.2 Vertical position	59
12.2.3 Distance to the main text	59
12.2.4 Font	59
12.2.5 Separator between notes	59
13 Indexing	60
13.1 Basics	60
13.2 Referring to critical notes	60
13.3 Separator between page and line numbers	60
13.4 Using <code>xindy</code>	61
13.5 Advanced setting	62
14 Glossary	62
14.1 Preamble setting	62
14.2 Commands	62
15 Tabular material	62
16 Sectioning commands	66
16.1 Sectioning commands without line numbers or critical notes	66
16.2 Sectioning commands with line numbering and critical notes	66
16.3 Optimization	67
17 Quotation environments	67
18 Page breaks	67
18.1 Control page breaking	67
18.2 Prevent page break in a long verses	67
19 Miscellaneous	68
19.1 Known and suspected limitations	68
19.1.1 Non-standard geometry	68
19.1.2 <code>floatrow</code> package compatibility	68
19.1.3 ‘No room for a new’	68
19.1.4 Marginal notes	69
19.1.5 Paragraph shape	69
19.1.6 Paragraphed footnotes	69
19.1.7 Use with other packages	70

19.2 Parallel typesetting	71
I Implementation overview	72
II Preliminaries	72
II.1 Links with original edmac	72
II.2 Package declaration	72
II.3 Package options	73
II.4 Loading packages	75
II.5 Compatibility with LuaTeX	75
II.6 Boolean flags	75
II.7 Messages	76
II.8 Gobbling	83
II.9 Miscellaneous commands	83
II.10 Prepare reledpar	83
II.11 Booleans provided by other optional packages which are required in any case	84
III Sectioning commands	84
IV List macros	88
V Line counting	90
V.1 Choosing the system of lineation	90
V.2 Line number margin	91
V.3 Line number initialization and increment	92
V.4 Line number locking	94
V.5 Line number style	95
V.6 Line number printing	95
V.7 Line number counters and lists	96
V.8 Line number locking counter	97
V.9 Line number associated to lemma	97
V.10 Reading the line-list file	101
V.11 Commands within the line-list file	103
V.12 Writing to the line-list file	116
VI Marking text for notes	122
VI.1 \edtext itself	123
VI.2 Substitute lemma	130
VI.3 Substitute line numbers	131
VI.4 Lemma disambiguation	132
VII Paragraph decomposition and reassembly	138
VII.1 Boxes, counters, \pstart and \pend	138
VII.2 Processing one line	144
VII.2.1 General process	144
VII.2.2 Process for “normal” line	145

VII.2.3 Process for line containing \eledsection command	147
VII.2.4 Hooks	147
VII.2.5 Sidenotes and marginal line number initialization	148
VIII Line and page number computation	148
IX Line number printing	152
X Pstart number printing in side	156
XI Restoring footnotes and penalties	157
XI.1 Add insertions to the vertical list	157
XI.2 Penalties	159
XI.3 Printing leftover notes	160
XI.4 Text before notes	161
XII Critical footnotes	162
XII.1 Fonts	162
XII.2 Individual note options	162
XII.3 Notes language	163
XII.4 General survey of the way we manage notes	164
XII.5 General setup	164
XII.6 Footnotes arrangement	165
XII.6.1 User level macro	165
XII.6.2 Normal footnote	165
XII.6.3 Paragraphed footnotes	170
XII.6.4 Columnar footnotes	177
XII.7 Critical notes presentation	184
XII.7.1 Font tools	184
XII.7.2 Pstart number in footnote	185
XII.7.3 Lemma printing	186
XII.7.4 Line number printing	186
XII.7.5 Footnote grouped by line	195
XIII Familiar footnotes	196
XIII.1 Adjacent footnotes	196
XIII.2 Regular footnotes for numbered texts	198
XIII.3 Footnote formats	200
XIII.4 Footnote arrangement	200
XIII.4.1 User level macro	200
XIII.4.2 Normal footnotes	201
XIII.4.3 Two columns footnotes	206
XIII.4.4 Three columns footnotes	209
XIII.4.5 Paragraphed footnotes	211
XIII.5 Wrapping footnote marks in hyperlink	215

XIV Code common to both critical and familiar footnote in normal arrangement	216
XV Footnotes' width for two columns	217
XVI Footnotes' order	218
XVII Footnotes' rule	218
XVIII Specific skip for first series of footnotes	219
XVIII.0.1 Overview	219
XVIII.0.2 User level command	219
XVIII.0.3 Internal commands	220
XIX Endnotes	221
XIX.1 Internal commands	221
XIX.2 User level commands	225
XIX.2.1 Inserting contents to endnotes	225
XIX.2.2 Printing endnotes	225
XX Generate series of notes	232
XX.1 Test if series is still existing	232
XX.2 Init specific to <code>reledpar</code>	232
XX.3 For critical footnotes	233
XX.3.1 Options	233
XX.3.2 Create inserts, needed to add notes in foot	234
XX.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc.	234
XX.3.4 Set standard display	237
XX.4 For familiar footnotes	237
XX.4.1 Options	238
XX.4.2 Create tools for familiar footnotes (<code>\footnotex</code>)	238
XX.5 The endnotes	240
XX.5.1 The auxiliary file	241
XX.5.2 The main macro	241
XX.5.3 Tools	242
XX.5.4 Internal commands	242
XX.5.5 The options	242
XX.6 Init standards series (A,B,C,D,E)	244
XXI Setting series display	244
XXI.1 Change series order	244
XXI.2 Test series order	244
XXI.2.1 Get the first series	245
XXI.3 Series setting	245
XXI.3.1 General way of working	245
XXI.3.2 Tools to set options	245

XXI.3.3 Tools to generate options commands	247
XXI.3.4 Options for critical notes	248
XXI.3.5 Options for familiar notes	250
XXI.3.6 Options for endnotes	251
XXI.4 Hooks for a particular footnote	252
XXI.5 Alias	253
XXII Output routine	253
XXII.1 Extra footnotes output	253
XXII.2 Patching standard output's commands	257
XXIII Cross referencing	259
XXIII.1 Compatibility with xref	274
XXIV Side notes	275
XXV Minipages and such	283
XXVI Indexing	287
XXVI.1 Looking on package order	288
XXVI.2 Auxiliary macros for \edindex	288
XXVI.3 Code specific to \edindexin critical footnotes	289
XXVI.4 Analysis of command in indexed text	290
XXVI.5 Code for the formatted index	291
XXVI.6 Main code	291
XXVI.7 Hyperlink	293
XXVI.8 'innote' and 'notenumber' option of indextols package	295
XXVII Glossaries	296
XXVIII Verse	298
XXVIII.1 Hanging symbol management	298
XXVIII.2 Using & character	299
XXVIII.3 Code category setting	299
XXVIII.4 Stanza count and indent	299
XXVIII.5 Numbering stanza	301
XXVIII.6 Stanza number in note	302
XXVIII.7 Main work	302
XXVIII.8 Restore catcode and penalties	305
XXIX Apparatus of Manuscripts	306
XXIX.1 User level macro	306
XXIX.2 Setting macro	307
XXIX.3 Counters and lists	308
XXIX.4 Auxiliary file macros	308
XXIX.5 Action macro	309
XXIX.6 Inserting footnote	314

XXIX.7 Other	314
XXX Arrays and tables	315
XXX.1 Preamble: macro as environment	315
XXX.2 Tabular environments	318
XXX.2.1 Disabling and restoring commands	318
XXX.2.2 Counters, boxes and lengths	322
XXX.2.3 Tabular typesetting	325
XXX.2.4 Environments	336
XXXI Quotation's commands	337
XXXII Section's title commands	338
XXXII.1 Commands to disable some feature	338
XXXII.2 General overview	338
XXXII.3 \beforeeledchapter command	339
XXXII.4 Auxiliary commands	339
XXXII.5 Patching standard commands	340
XXXII.6 Main code of \eledxxx commands	345
XXXII.7 Macros written in the auxiliary file	348
XXXIII Page breaking or no page breaking depending of specific lines	350
XXXIV Long verse: prevents being separated by a page break	352
XXXV Tools for hyperref package	352
XXXVI Compatibility with eledmac	353
Appendix A Things to do when changing versions	356
Appendix A.1 Migrating from edmac to ledmac	356
Appendix A.2 Migration from ledmac to eledmac	357
Appendix A.3 Migration to eledmac 1.5.1	358
Appendix A.4 Migration to eledmac 1.12.0	358
Appendix A.5 Migration to eledmac 17.1	359
Appendix A.6 Migration to eledmac 1.21.0	359
Appendix A.6.1 \Xledsetnormalparstuff and \ledsetnormalparstuff	359
Appendix A.6.2 Endnotes	359
Appendix A.7 Migration to eledmac 1.22.0	359
Appendix A.8 Migration to eledmac 1.23.0	359
Appendix A.9 Migration from eledmac to reledmac	360
Appendix A.9.1 Risk of 'no room for a new'	360
Appendix A.9.2 Multiple indices with memoir	360
Appendix A.9.3 Deprecated commands and options	360
Appendix A.9.4 \renewcommand replaced by command	361
Appendix A.9.5 Commands the names of which have been changed . . .	361
Appendix A.9.6 Endnotes	363

Appendix A.9.7 Z Series	363
Appendix A.9.8 Internal commands	363
Appendix A.10 Migration to <code>reledmac</code> 2.1.0	363
Appendix A.11 Migration to <code>reledmac</code> 2.1.3	363
Appendix A.12 Migration to <code>reledmac</code> 2.3.0	363
Appendix A.13 Migration to <code>reledmac</code> 2.4.0	364
Appendix A.14 Migration to <code>reledmac</code> 2.5.0	364
Appendix A.15 Migration to <code>reledmac</code> 2.7.0	364
Appendix A.16 Migration to <code>reledmac</code> 2.7.2	364
Appendix A.17 Migration to <code>reledmac</code> 2.8.0	364
Appendix A.18 Migration to <code>reledmac</code> 2.13.1	364
Appendix A.19 Migration to <code>reledmac</code> 2.18.0	365
References	366
Index	366
Change History	414

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

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).

combination enables you to recover to some extent the text of each line as ASCII code, facilitating the creation of concordances, an *index verborum*, etc.

As of 1994, we were pleased to be able to say that edmac was being used for the real-life book production of several interesting editions, such as the Latin texts of Euclid's *Elements*,² an edition of the letters of Nicolaus Copernicus,³ Simon Bredon's *Arithmetica*,⁴ a Latin translation by Plato of Tivoli of an Arabic astrolabe text,⁵ a Latin translation of part II of the Arabic *Algebra* by Abū Kāmil Shujā' b. Aslam,⁶ the Latin *Rithmarchia* of Werinher von Tegernsee,⁷ a middle-Dutch romance epic on the Crusades,⁸ a seventeenth-century Hungarian politico-philosophical tract,⁹ an anonymous Latin compilation from Hungary entitled *Sermones Compilati in Studio Generali Quinqueecclesiensi in Regno Ungarie*,¹⁰ the collected letters and papers of Leibniz,¹¹ Theodosius's *Spherics*, the German *Algorismus* of Sacrobosco, the Sanskrit text of the *Kāśikāvṛtti* of Vāmana and Jayāditya,¹² and the English texts of Thomas Middleton's collected works.

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.

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

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

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

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

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

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

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

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

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

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

¹²Being prepared at Poona and Lausanne Universities.

1.2.3 eledmac

Important changes were put in version 1.0, to make ledmac more easily extensible (see 7 p. 35). 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. 357.

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. 360.

1.3 Bibliography

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

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

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

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

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

3 Compatibility warning

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

Please read 7.13.6 p. 47.

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

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

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

4 Options

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

4.1 Specific features

draft underlines lemmas in the main text.

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

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

eledmac-compat help to migrate from `eledmac` to `reledmac` (see Appendix A.9.5 p. 361).

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

nopbinverse prevents page break within verse environment.

noquotation by default, the quotation environment is redefined within numbered text. You can disable this redefinition with `noquotation` (see 17 p. 67).

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

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

xindy and **xindy+hyperref** select xindy as the index processor (13.4 p. 61).

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. 66).

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. 53). However, such features use up computer memory, at the expense of other processing needs. So if you do not need this feature, use `noledgroup` option. This should make `reledmac` faster.

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

5 Text lines and paragraphs numbering

5.1 Text lines numbering

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

```
\beginnumbering
Text
\endnumbering
```

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

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

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

5.2 Paragraphs

5.2.1 Basics

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

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

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

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

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

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

The lines of this paragraph are
not numbered.

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

5.2.2 Automatically producing `\pstart ... \pend`

`\autopar` You can use `\autopar` to avoid the nuisance of this paragraph markup and still have

every paragraph automatically numbered. The scope of the `\autopar` command needs to be limited by keeping it within a group, as follows:

```
\begingroup
  \beginnumbering
  \autopar

  A paragraph of numbered text.

  Another paragraph of numbered
  text.

  \endnumbering
\endgroup
```

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

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

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

Note that a `\noindent` will be automatically added before this argument, and, consequently, a `\parskip` will be inserted. You can use a second optional argument, in brackets, to not have this `\noindent`.

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

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

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

This feature is also useful when typesetting verses (see 9 p. 49) or `reledpar` (see 19.2 p. 71).

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

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

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

`\AtEveryPend`

Note that a `\noindent` will be inserted before the argument, and, consequently, a `\parskip`. You can use the starred version of `\AtEveryPstart` and `\AtEveryPend` to no insert the `\noindent`.

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

`\AtStartEveryPstart`
`\AtEndEveryPend`

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

```
\AtStartEveryPstart{*}
```

Instead of manually doing

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

5.2.5 Numbering paragraphs (`\pstart`)

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

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

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

On each `\beginnumbering` the numbering restarts.

`\sidepstartnumtrue`

With the `\sidepstartnumtrue` command, the number of `\pstart` will be printed 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 \LaTeX or \XeTeX , 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 \LaTeX .

`\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 \LaTeX may reach its memory limit. There are two solutions to this.

The first solution is to get a larger \LaTeX 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:

```
\linenumberlist
```

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

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

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

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

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

5.3.3 Setting lineation reset

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

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

5.3.4 Setting line number margin

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

`\linenummargin{left}`

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

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

5.3.5 Other settings

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

5.4 Changing the line numbers

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

5.4.1 Sublineation

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

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

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

You can change the separator between line number and subline number either by using `\Xsublinesep` without any optional argument (7.2.9 p. 39) or by using `\Xsublinesepside`. But in the second case, it will change the separator only for line numbers in the margins, not in the footnotes.

5.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 49.

`\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>}`
`\sublinenumberstyle` to change the numbering style. `<style>` must be one of:

Alph Uppercase letters (A ... Z).

alph Lowercase letters (a ... z).

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

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

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

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

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

5.4.5 Skipping and hiding number

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

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

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

5.5 Executing code at each line

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

6 Apparatus commands

6.1 Terminology

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

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

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

So :

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

6.2 Critical notes

6.2.1 The lemma

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

```
\edtext{<lemma>}{<commands>}
```

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

For example:

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

1 Smith] Jones C, D.

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

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

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

1 Smith] Jones C, D.

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

However, `\edtext` cannot handle overlapping but unnested notes—for example, one note covering lines 10–15, and another covering 12–18; an `\edtext` that starts in the `<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

`\Bfootnote`

`\Cfootnote`

`\Dfootnote`

`\Efootnote`

just below the main text, followed by the rest in turn, down to the E notes at the bottom. These are the main macros that you will use to construct the critical apparatus of your text.

If you need more series of critical notes, please look at 6.7.1 p. 34.

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. 7.2.5 p. 38).
- `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. 42).
- `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` **Inserting endnotes** The package also maintains five separate series of endnotes.

`\Bendnote` If you do not need the endnotes facility, you should use `noend` option when loading `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` `\Aendnote[<option>]{<text>}`.

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

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 7.2.5 p. 38).
- `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. 42).
- `linangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

`\doendnotesbysection` However, you may want to print the endnotes of one given series covering the first numbered section, then the endnotes of another given series covering the first numbered section, then the endnotes of the first given series covering the second numbered section,

then the endnotes of the second given series covering the second numbered section, and so forth. In this case, use `\doendnotesbysection{⟨s⟩}`. For each value of `⟨s⟩`, the first call of the command will print the notes for the first series, the second call will print the notes for the second series etc. For example, do:

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

Note that by default inside endnotes no separator is used between the lemma and the content. However you can use the `\Xendlemmaseparator` macro to define one (7.5.2 p. 42).

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

`toendnotes`
`Xtoendnotes`

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

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

For example:

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

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

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

However, in some cases, you may want to write a once-expanded¹⁴ version of the `⟨content⟩`, that is the version where the commands are expanded on the first level. This can be, for example, to get a counter value. Use the starred version in this case. For example:

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

¹⁴The expansion mechanism of \TeX is a quite complex problem, but fundamental. We have no place to explain it fully here. Read introduction to \TeX to understand well.

6.2.4 Paragraph in critical apparatus

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

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

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

6.2.5 Change lemma and line number

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

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

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

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

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

This command does not change the marginal line numbers in any way; it just changes the numbers passed to the notes. Its use comes in situations that `\edtext` has trouble dealing with for whatever reason. If you need notes for overlapping passages that aren't nested, for instance, you can use `\lemma` and `\linenum` to generate such notes despite the limitations of `\edtext`. If the `⟨lemma⟩` argument to `\edtext` is extremely long, you may run out of memory; here again you can specify a note with an abbreviated lemma using `\lemma` and `\linenum`. The numbers used in `\linenum` need not be entered manually; you can use the 'x-' symbolic cross-referencing commands below (11 p. 54) to compute them automatically.

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

6.2.6 Changing the names of commands for critical apparatus

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

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

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

6.3 Disambiguation of identical words in the apparatus

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

6.3.1 Basic use

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

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

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

6.3.2 Notes about input encoding with UTF-8 processor

If you use UTF-8 processor, like `XYLaTeX` or `LuaLaTeX`, 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)

¹⁵We use `\newcommand` and `\newcommandx` instead of classical `\let` command because the `edtabular` environments have to modify the notes definition, and we need to use the newest definition of notes. Read the handbook of `xargs` to know more about `\newcommandx`.

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

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

- If you use \XeTeX , add this line in your preamble: `\XeTeXinputnormalization 1`.
- If you use \LuaTeX , use the `uninormalize` package of Michal Hoftich¹⁶ 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`:

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

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

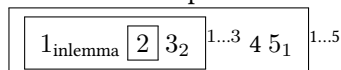
¹⁶<https://github.com/michal-h21/uninormalize>.

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

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

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

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



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

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

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

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

Here, the corresponding code:

```

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

```

1	A (1) A (2) A (3) A (4) A (5)
<hr style="width: 100%;"/>	
1	A ¹ ...A ⁵] level 1
1	A ¹ ...A ³] level 2
1	A ² (2)] level 3

6.3.4 Customizing

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

```

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

```

6.4 Apparatus of Manuscripts

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

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

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

6.4.1 Marking sections of text

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

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

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

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

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

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

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

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

6.4.2 Layout of the apparatus of manuscripts

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

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

6.4.3 Settings

As the apparatus of manuscripts technically consists of first-level critical notes (‘A’ series), any setting available for critical notes can be applied (7 p. 35). However, the following *additional* commands are available.

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

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

`\setmsdataposition` If you want the manuscript apparatus to be on the same level of critical footnotes as the other apparatuses, for each line, reledmac will first insert the manuscript apparatus, then the other footnotes. You can change it using:
`\msdataposition{regular-msdata}`
 And restore the default behaviour using `\msdataposition{msdata-regular}`

6.5 Familiar notes

6.5.1 Basic use

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

6.5.2 Customizing mark

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

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

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

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

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

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

There are similar command triples for the other series.

6.5.3 Separator for multiple footnotes

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

`\multfootsep` `\multfootsep` is used as the separator between footnote markers. Its default definition is:

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

6.6 Printing the footnote mark without printing the footnote text

`\footnoteXmark` If in certain cases, you can't directly use `\footnoteX`; for example, when using `\uline`
`\footnoteXtext` command of the `ulem` package. You need to print the footnote mark first, then call the
footnote text to be inserted.
For all $\langle X \rangle$ command, `reledmac` provides a `\footnote\langle X \rangle mark` command and a
`\footnote\langle X \rangle text` command, equivalent to standard \TeX 's command `\footnotemark`
and `\footnotetext`. For example, to use with `\uline`, do:

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

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

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

6.7 Changing series

6.7.1 Create a new series

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

6.7.2 Delete series

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

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

6.7.3 Series order

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

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

6.8 Position of critical and familiar footnotes

`\fnpos` `\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}
```

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

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

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

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

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

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

7 Critical apparatus appearance

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

When a length, noted $\langle l \rangle$, is used, it can be stretchable: `a plus b minus c`. The final length m is calculated by \TeX to have: $a - c \leq m \leq a + b$. If you use some relative unit¹⁷, it will be relative to font size of the footnote, except for commands concerning the place kept by the notes — including blank space.

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

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

There is also name convention:

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

7.1 Notes arrangement in a series

`\Xarrangement`
`\arrangementX`

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

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

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

- `paragraph` formats all of the footnotes of a series as a single paragraph; if you use this arrangement, you are strongly encouraged to read 19.1.6 p. 69.
- `twocol` formats them as separate paragraphs, but in two columns;
- `threecol`, in three columns.
- `normal`, restore normal arrangement.

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

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

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

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

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

7.2 Control line number printing

7.2.1 Print line number only at first time

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

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

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

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

7.2.2 Print page number only at first time

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

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

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

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

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

In any case, you can use:

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

7.2.3 Arbitrary text before line number

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

7.2.4 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 `\Xlinerangeseparator[⟨s⟩]{⟨text⟩}`, and with `\Xendlinerangeseparator[⟨s⟩]{⟨text⟩}` for critical endnotes.

7.2.5 Abbreviate line range

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

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. 25.

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

7.2.6 Disable line number

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

7.2.7 Printing pstart number

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

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

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

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

7.2.8 Printing stanza number

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

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

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

7.2.9 Separator between line and subline numbers

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

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

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

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

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

7.2.10 Separator between page and line numbers

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

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

7.2.11 Space around number

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

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

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

`\Xnonbreakableafternumber` By default, the space defined by `\Xafternumber` is breakable. With `\Xnonbreakableafternumber[$\langle s \rangle$]{ $\langle l \rangle$ }` it becomes nonbreakable.

7.2.12 Space around line symbol

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

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

`\Xendbeforesymmlinenum` `\Xendbeforesymmlinenum` and `\Xendaftersymmlinenum` are the equivalents of `\Xbeforesymmlinenum` and `\Xaftersymmlinenum` for the endnotes.

7.2.13 Space in place of number

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

`\Xendinplaceofnumber` `\Xendinplaceofnumber[$\langle s \rangle$]{ $\langle l \rangle$ }` is the same, for critical endnotes.

7.2.14 Boxing line number and line symbol

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

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

`\Xboxsymmlinenum` `\Xboxsymmlinenum[$\langle s \rangle$]{ $\langle l \rangle$ }` is the same as `\Xboxlinenum` but for the line number symbol.

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

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

L to align left (default value);

R to align right;

C to center.

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

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

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

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

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

7.3 For endnotes

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

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

7.4 Arbitrary code around line number

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

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

7.5 Separator between the lemma and the note

7.5.1 For footnotes

<code>\Xlemmaseparator</code>	By default, in a footnote, the separator between the lemma and the note is a right bracket (<code>\rbracket</code>) ¹⁸ . You can use <code>\Xlemmaseparator[⟨s⟩]{⟨Xlemmaseparator⟩}</code> to change it. The optional argument can be used to specify the series in which it is used. Note that there is a non-breakable space between the lemma and the separator, but a breakable space between the separator and the following text.
<code>\Xbeforelemmaseparator</code>	Using <code>\Xbeforelemmaseparator[⟨s⟩]{⟨l⟩}</code> you can add some space between lemma and separator. If your lemma separator is empty, this space won't be printed. The default value is 0 em.
<code>\Xafterlemmaseparator</code>	Using <code>\Xafterlemmaseparator[⟨s⟩]{⟨l⟩}</code> you can add some space between separator and note. If your lemma separator is empty, this space will not be printed. The default value is 0.5 em.
<code>\Xnolemmaseparator</code>	You can suppress the lemma separator, using <code>\Xnolemmaseparator[⟨s⟩]</code> , which is simply a alias of <code>\Xlemmaseparator[⟨s⟩]{}</code> .
<code>\Xinplaceoflemmaseparator</code>	With <code>\Xinplaceoflemmaseparator[⟨s⟩]{⟨l⟩}</code> you can add a space if no lemma separator is printed. The default value is 1 em.

7.5.2 For endnotes

<code>\Xendlemmaseparator</code>	By default, there is no separator inside endnotes between the lemma and the content of the note. You can use <code>\Xendlemmaseparator[⟨s⟩]{⟨Xendlemmaseparator⟩}</code> to change this. The optional argument can be used to specify the series in which it is used. A common value of <code>⟨Xendlemmaseparator⟩</code> is <code>\rbracket</code> . Note that there is a non-breakable space between the lemma and the separator, but a breakable space between the separator and the following text.
<code>\Xendbeforelemmaseparator</code>	Using <code>\Xendbeforelemmaseparator[⟨s⟩]{⟨l⟩}</code> you can add some space between the lemma and the separator. If your lemma separator is empty, this space won't be printed. The default value is 0 em.
<code>\Xendafterlemmaseparator</code>	Using <code>\Xendafterlemmaseparator[⟨s⟩]{⟨l⟩}</code> you can add some space between 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.
<code>\Xendinplaceoflemmaseparator</code>	With <code>\Xendinplaceoflemmaseparator[⟨s⟩]{⟨l⟩}</code> you can add some space if you chose to remove the lemma separator. The default value is 0.5 em.

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

7.6 Font style

7.6.1 For line number

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

7.6.2 For the lemma

<code>\lemmadisablefontselection</code>	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 <code>\Xlemmadisablefontselection[⟨s⟩]</code> command allows to disable it for a specific series.
<code>\lemmadisablefontselection</code>	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 <code>\Xendlemmadisablefontselection[⟨s⟩]</code> to disable it for a specific series.
<code>\Xlemmafont</code> <code>\Xendlemmafont</code>	Use <code>\Xlemmafont[⟨s⟩]{⟨cmd⟩}</code> to apply a \TeX font command to the lemma. For example, to have boldface lemma: <code>\Xlemmafont{\bfseries}</code> <code>\Xendlemmafont[⟨s⟩]{⟨cmd⟩}</code> is the same for endnotes.

7.6.3 For all notes

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

7.7 Wrapping notes

7.7.1 Wrapping lemmas

<code>\Xwraplemma</code>	<code>\Xwraplemma[⟨s⟩]{⟨cmd⟩}</code> is used to wrap, in the footnote, the lemma in a \TeX com-
--------------------------	---

mand. For example, with the `bidir` 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[\langle s \rangle]{\langle cmd \rangle}` is used to wrap the footnote contents — excluding the lemma — in a \LaTeX command.

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

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

`\wrapcontentX` `\wrapcontentX[\langle s \rangle]{\langle cmd \rangle}` 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[\langle s \rangle]` to enable indentation.

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

`\Xhangindent` For critical notes NOT paragraphed you can define an indent with `\Xhangindent[\langle s \rangle]{\langle l \rangle}`, 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[\langle s \rangle]{\langle l \rangle}`, 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[\langle s \rangle]{\langle l \rangle}`, 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[\langle s \rangle]{\langle code \rangle}` is to be used at the beginning of the critical footnotes.

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

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

7.10 Arbitrary code before inserting note

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

`beforeinsertingX`

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 `bidirectional`, 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. 43)).

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 \TeX . 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 and notes grouped by line

7.12.1 Mark separation of notes

`\Xafternote` You can add some horizontal space after a note by using `\Xafternote[⟨s⟩]{⟨l⟩}` (for `\afternoteX`

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
`\parafootsepX` 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 Grouping notes by line

`Xgroupbyline` If you do not use `\Xarrangement{paragraph}`, you may want to group all the critical footnotes related to the same line in the same paragraph. In this case, use `\Xgroupbyline[⟨series⟩]`.

In many cases, you might like to use it in combination with `\Xnumberonlyfirstinline` (7.2.1 p. 37).

`\Xgroupbylineseparetwolines` Note that the `\Xafternote` and `\Xparafootsep` settings are used to determine space and content between footnotes (7.12 p. 45). Suppose you have two notes on line 1 which overlap lines 1 and 2. This last note will be printed, if you use `\Xgroupbyline` in the same group as the previous one. In the case you want that note to be distinct, you must use both `\Xgroupbyline` and `\Xgroupbylineseparetwolines[⟨s⟩]`.

In many cases, you might like to use it in combination with `\Xnumberonlyfirstintwolines` (7.2.1 p. 37)

7.13.2 Text before notes

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

7.13.3 Code before notes

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

7.13.4 Spacing

`\Xbeforenotes` You can change the vertical space before the rule of the critical notes with `\Xbeforenotes[⟨s⟩]{⟨l⟩}`.

The default value is 1.2em plus .6em minus .6em.

Be careful, the standard \LaTeX 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 \LaTeX footnote rule, which is used by `reledmac`, decreases 3pt. These 3pt are not changed by this command.

`\Xprenotes` You can set the space before the first series of critical notes printed on each page and set a different amount of space for each subsequent series on the page. You can do it with `\Xprenotes{⟨l⟩}`. The default value is 0pt. You can disable this feature by setting the length to 0pt.

`\prenotesX` You can set the space before the first printed (in a page) series of familiar notes to be different from the space before other series. The default value is 0pt. You can do this with `\prenotesX{⟨l⟩}`. You can disable this feature by setting the length to 0pt.

7.13.5 Rule

`\Xafterrule` You can change the vertical space printed after the rule of the critical notes with `\Xafterrule[⟨s⟩]{⟨l⟩}`. The default value is 0pt.

Be careful, the standard \LaTeX 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 \LaTeX footnote rule, which is used by `reledmac`, adds 2.6pt. These 2.6pt are not changed by this command.

7.13.6 Maximum height

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

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

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

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

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

7.13.7 Width

`\Xwidth` `\Xwidth[⟨s⟩]{⟨l⟩}` sets the total width of critical footnotes. `\widthX[⟨s⟩]{⟨l⟩}` does the same for familiar footnotes.
`\widthX` `⟨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. 45.

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.

`\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@lemmafонт` We will briefly discuss `\select@lemmafонт` 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. 51.

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

9.3 Repeating stanza indents

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

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

is like

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

Be careful: the feature is changed in eledmac 1.5.1. See Appendix A.3 p. 358.

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

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

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

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

9.4 Manual stanza indent

```
\stanzaindent
\stanzaindent*
```

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

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

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

9.5 Stanza breaking

```
\setstanzapenalties
```

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

The command

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

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

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

9.6 Hanging symbol

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

`\sethangingsymbol`

`\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. 67 for further details.

9.8 Content before/after verses

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

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

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

`\AtEveryStanza`

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

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

<code>\AtStartEveryStanza</code>	<ul style="list-style-type: none"> • Use <code>\AtStartEveryStanza</code> to automatically add content at the beginning of stanza (in the same paragraph). • <code>&</code> can be replaced by <code>\newverse</code> with two optional arguments (in brackets). The first will be printed after the current verse, the second before the next verse. A <code>\noindent</code> is automatically inserted before the contents of these optional arguments. Use a third and fourth optional argument to not add these <code>\noindents</code> (to add content respectively after the current verse / before the next verse). • Use <code>\AtEveryPend{<arg>}</code> to automatically add content after verses (including the final one) and <code>\AtEveryPstart{<arg>}</code> to automatically add content before verses (including the first one). • <code>\&</code> can take an optional argument (in brackets). Its content will be printed after the stanza.
<code>\AtEveryStopStanza</code>	<ul style="list-style-type: none"> • Use <code>\AtEveryStopStanza</code> to automatically add content after the end of stanzas (not in the same paragraph). <p>Note that a <code>\noindent</code> will be inserted before the argument, and, consequently, a <code>\parskip</code>. You can use the starred version of <code>\AtEveryStopStanza</code> to avoid this <code>\noindent</code>.</p>
<code>\AtStartEveryStanza</code>	<ul style="list-style-type: none"> • Use <code>\AtStartEveryStanza</code> to automatically add content at the end of stanza (in the same paragraph).

9.9 Numbering stanza

<code>\numberstanzatrue</code>	If you want to automatically number stanzas, use <code>\numberstanzatrue</code> . In this case, the line number will restart at each <code>\stanza</code> .
<code>\numberstanzafalse</code>	
	If you want to disable this feature again, use <code>\numberstanzafalse</code> .
	You can use this feature in combination with <code>\Xstanza</code> (7.2.8 p. 39).
<code>\thestanza</code>	You can redefine <code>\thestanza</code> to change the aspect of stanza number. Default value is:
	<pre> \renewcommand{\thestanza}{% \textbf{\arabic{stanza}}% } </pre>
<code>\stanzanumwrapper</code>	<p>You can change the value of the stanza counter with the usual commands of \LaTeX.</p> <p>You can redefine <code>\stanzanumwrapper</code> in order to modify the way the stanza number is inserted in the flow of text. Default value is:</p> <pre> \newcommand{\stanzanumwrapper}[1]{% \flagstanza{#1}% } </pre>

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. 51).

10 Grouping

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

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

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

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

`ledgroupsize` The `ledgroupsize` environment is similar to `ledgroup` except that you must specify a width for the environment, as with a `minipage`.
`\begin{ledgroupsize}[⟨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.¹⁹

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

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

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

The `\edlabel` command works by writing macros to `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. 56).

11.3 Cross-referencing which return a number in any case

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

`\xlineref` However, there are situations in which you will want `reledmac` to return a number
`\xsublineref` without displaying any warning messages about undefined labels or the like: if you want
`\xpstartref` to use the reference in a context where `ℒTEX` is looking for a number, such a warning will lead to a complaint that the number is missing. This is the case for references used within the argument to `\linenum`, for example (see 6.2.5 p. 27).

¹⁹More precisely, you should stick to characters in the `TEX` categories of “letter” and “other”.

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

It automatically calls `\linenum` (q.v., 6.2.5 p. 27 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.

For example, one might use the following:

```
\beginnumbering

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

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

\endnumbering
```

11.4 Not automatic cross-referencing

`\edmakelabel`

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

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

11.5 Normal \TeX cross-referencing

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

11.6 References to start and end lines

11.6.1 Reference to main text lines

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

`\edlabelS` Use `\edlabelS{<label>}` to mark the start line of the passage.
`\edlabelE` Use `\edlabelE{<label>}` to mark the end the end line of the passage. These two
commands just create 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 clas-
sical `\edlabel`, the `<label>` could be use with `\Seref` and `\Serefwithpage`.
`\Seref` The main utility is to use them with three other commands. `\Seref{<label>}` will
make a cross-reference printed as a reference in critical footnotes.
`\Serefwithpage` `\Serefwithpage` will make a cross-reference printed as a reference in critical end-
notes.
`\Serefonlypage` `\Serefonlypage` will make a cross-reference printed only with page number.

11.6.2 References to lines that are commented on in the apparatus

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

`\applabel` If you use `\applabel{<label>}` inside the second argument of a `\edtext`, `reledmac`
will add a `\edlabel` at the beginning and end of the marked passage. The label at the
beginning of the passage will have the title `<label>:start`, while the label at the end
will have the title `<label>:end`.
If you use `\linenum` (6.2.5 p. 27) to refer to these labels, `reledmac` will use your line
settings to refer to the passage.
`\appref` You can also use `\appref{<label>}` and `\apprefwithpage{<label>}` to refer to these
`\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 do not use `\setapprefprefixmore`, the argument of `\setapprefprefixsingle` will be used in any case.

`\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 `\XendbeforepagenumberSerefonly` (see 11.6.3 p. 57). So if you use `\Xendbeforepagenumber` without any optional argument, it will override this setting.

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

These commands are the following:

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

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

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

- `\Xendafterpagenumber` (not for `\Serefonlypage`)

- `\Xendbeforepagenumber`
- `\Xendlineflag` (for `reledpar`), enabled by default.
- `\Xendlineprefixmore`
- `\Xendlineprefixsingle`
- `\Xendlinerangeseparator`
- `\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. 25).

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

11.7 Compatibility with `xr` package

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

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

12.2 Setting

12.2.1 Width

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

12.2.2 Vertical position

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

12.2.3 Distance to the main text

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

12.2.4 Font

`\ledlsnotefontsetup` These macros specify how the sidenote texts are to be typeset. The initial definitions are:
`\ledrsnotefontsetup` `\newcommand*{\ledlsnotefontsetup}{\raggedleft\footnotesize}% left`
 `\newcommand*{\ledrsnotefontsetup}{\raggedright\footnotesize}% right`

These can of course be changed to suit.

12.2.5 Separator between notes

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

13 Indexing

13.1 Basics

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

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

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

13.2 Referring to critical notes

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

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

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

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

13.3 Separator between page and line numbers

`\pagelinesep` The page & line number combination is written as `page\pagelinesep line`, where the default definition is `\newcommand{\pagelinesep}{-}` so that an item on page 3,

line 5 will be noted as being at 3–5. You can renew `\pagelinesep` to get a different separator.

- is the default separator used by the MAKEINDEX program.
- You can reconfigure it, this example defines a colon as separator:

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

However, you also have to configure your `.ist` index style file. For example if you use `:` as separator²⁰.

```
page_compositor ":"
```

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

13.4 Using xindy

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

`pagenumber–linenumber`

An example of such a file is provided in the “examples” folder. Read the `xindy` handbook to learn how to use it.²¹

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.²²

²⁰For further detail, you can read <http://tex.stackexchange.com/a/32783/7712>.

²¹Or, for people who read French, read <http://geekographie.maieul.net/174>.

²²These are the recommended lines to provide the best possible compatibility between `hyperref` and `xindy`, even without using `reledmac`.

13.5 Advanced setting

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

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

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

14 Glossary

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

14.1 Preamble setting

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

- Or set `.glossaries`:

```
\glsSetCompositor{-}
```
- Or set `reledmac`:

```
\renewcommand{\pagelinesep}{-}
```

In this case, the above will have consequences for your use of `\edindex` and you should set your `.ist` file (13.3 p. 60).

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
```

```

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

```

1	2	3
a	bb	ccc
AAA	BB	C

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

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

For example:

```

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

```

produces the following parallel pair of verses.

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

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

```

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

```

1	2	3	4
	$F + G + C$		
a	bb	ccc	dddd

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

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          &      & l    & co   & ghweropjklmnbvcxys \\
1          & 2    & 3    & \edrowfill{4}{5}{\hrulefill} & \\
\end{tabularr}
```

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

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

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

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

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

1	2	3	4
a	$\underbrace{\hspace{1em}}$		d
A	B	C	D

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

```

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

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

For example:

```

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

```

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

`\edvertline` The macro `\edvertline{<height>}` draws a vertical line `<height>` 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}

```

a	b	C	d	
v	w	x	y	
m	n	o	p	
k		L	cvb	

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

16 Sectioning commands

16.1 Sectioning commands without line numbers or critical notes

The standard sectioning commands (`\chapter`, `\section` etc.) can be used inside numbered text. In this case, you must call them as an optional argument of `\pstart` (5.2.3 p. 18):

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

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

16.2 Sectioning commands with line numbering and critical notes

You have to use the following commands:

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

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

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

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

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

16.3 Optimization

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

17 Quotation environments

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

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

18 Page breaks

18.1 Control page breaking

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

`\ledpb`
`\lednopb`

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

These commands have effect only at the second run.

`\ledpbsetting` These two commands take effect at the beginning of line in which they are called. For example, if you call `\ledpb` at l. 444, then l. 443 will be at the p. *n*, and the l. 444 at the p. *n* + 1. However, you can change the behavior and decide they will have effect after the end of the line, adding `\ledpbsetting{after}` at the beginning of your file (better: in your preamble). With the previous example, l. 444 will be on p. *n* and l. 445 will be on p. *n* + 1.

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

18.2 Prevent page break in a long verses

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

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

19 Miscellaneous

<code>\extensionchars</code>	When the package assembles the name of the auxiliary file for a section, it prefixes <code>\extensionchars</code> 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 <code>\renewcommand{\extensionchars}{!}</code> , then you would get temporary files called <code>jobname. !1</code> , <code>jobname. !2</code> , etc.
<code>\ifledfinal</code>	The package can take options. The option ‘final’, which is the default is for final typesetting; this sets <code>\ifledfinal</code> to TRUE. The other option, ‘draft’, may be useful during earlier stages and sets <code>\ifledfinal</code> to FALSE.
<code>\showlemma</code>	<p>The lemma within the text is printed via <code>\showlemma{lemma}</code>. Normally, or with the ‘final’ option, the definition of <code>\showlemma</code> is:</p> <pre>\newcommand*\showlemma[1]{#1}</pre> <p>so it just produces its argument. With the ‘draft’ option it is defined as</p> <pre>\newcommand*\showlemma[1]{\textit{#1}}</pre> <p>so that its argument is typeset in an italic font, which may make it easier to check that all lemmas have been treated.</p> <p>If you would prefer some other style, you could put something like this in the preamble:</p> <pre>\ifledfinal\else \renewcommand*\showlemma[1]{\textbf{#1}}% or simply ...[1]{#1} \fi</pre>

19.1 Known and suspected limitations

19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.13.6 p. 47 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. 67 in order to know which are the available options.

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

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

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

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

19.1.4 Marginal notes

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

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

19.1.5 Paragraph shape

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

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

19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.1 p. 36, and described in more detail on XII.6.3 p. 173, really is a nuisance if that is something you need to do. There are some possible solutions, described by Michael Downes, but this area remains unsatisfactory.

If you use more than one series of paragraphed notes, it may happen, in some particular cases, that only the footnote rule, with no accompanying footnotes, be printed. In this case use `reledmac` package option `nopenalties` which should solve the problem,

but also may produce widow or orphan lines. For the time being, we have no solution of this problem.

`\footfudgefiddle`

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

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

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

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

Finally, in many cases you should use `\Xmaxhnotes` and / or `\maxhnotesX` (7.13.6 p. 47), 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 this²³ 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{...}}}
```

²³Reported by Dirk-Jan Dekker in the CTT thread ‘Incompatibility of “color” package’ on 2003/08/28.

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

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

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

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

19.2 Parallel typesetting

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

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

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}[2017/05/04 v2.20.0 typesetting critical editions]
4 %

```

II.3 Package options

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

```

\ifnocritical@
\if@noeled@sec
\ifnoend@
\ifnofamiliar@
\ifnoledgroup@
\ifparapparatus@
\ifnoquotation@
\iflednopbinverse
\ifparledgroup
\ifwidthliketwocolumns
\ifxindy@
\ifxindyhyperref@
\ifeledmaccompat@

```

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

```

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}

```

And now, the options of `reledmac`.

```

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

```

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.

```

78 \ProcessOptionsX*\relax
79
80 %

```

II.4 Loading packages

Loading package `xargs` to declare commands with optional arguments. `Etoolbox` is also used to make code clearer - for example, in dynamic command names (which can replace `\csname` etc.). Use `suffix` to declare commands with a starred version, `xstring` to work with strings, `ifluatex` and `ifxetex` to test if `LuaTeX` or `XYTeX` is running, and `ragged2e` to manage ragged justification for paragraphed notes.

```

81 \RequirePackage{xargs}
82 \RequirePackage{etoolbox}
83 \@ifl@t@r\fmtversion{2015/10/01}
84 {\ifboolexpr{not test{\@ifl@t@r\fmtversion{2016/03/31}} or (test{\
85   ifdefstring{\fmtversion}{2016/03/31}} and test {\ifnumless{\patch@level
86   }{3}})}}}%
87   {\PackageWarning{reledmac}{You are using a LaTeX version older than
88   2016/03/31 patch 3.%
89   \MessageBreak You are strongly encouraged to use a newer version.}}}%
90   {}%
91 }%
92 \RequirePackage{suffix}
93 \RequirePackage{xstring}
94 \RequirePackage{ifluatex}
95 \RequirePackage{ragged2e}
96 \RequirePackage{ifxetex}%
97 %

```

II.5 Compatibility with LuaTeX

Here, we enable some primitives for `LuaTeX`.

```

98 \ifx\directlua\undefined\else%
99   \directlua{tex.enableprimitives("",{"texdir","pardir","bodydir"})}
100 \fi
101 %

```

II.6 Boolean flags

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

```

102 \newif\ifl@dmemoir
103 \@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
104
105 %

```

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

```

106 \newif\if@ledgroup%
107 %

```

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

```

108 \newif\ifl@imakeidx
109 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value
110 %

```

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

```

111 \newif\ifl@indextools%
112 \@ifpackageloaded{indextools}{%
113   \l@indextoolstrue%
114   \l@imakeidxtrue%
115   \let\imki@wrindexentry\indtl@wrindexentry%
116 }{}%
117 %

```

False is the default value. We consider indextools as a variant of imakeidx. That is why we set \ifl@imakeidx to true. We also let \imki@wrindexentry to \indtl@wrindexentry.

\ifl@footmisc Define a flag if the footmisc package has been loaded.

```

118 \newif\ifl@footmisc
119 \@ifpackageloaded{footmisc}{\l@footmisctrue}{}%False is the default value
120 %

```

\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.

```

121 \ifdef{\if@RTL}{\newif\if@RTL}
122 %

```

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

```

123 \newif\if@firstlineofpage%
124 \newif\if@firstlineofpageR%
125 %

```

II.7 Messages

All the messages are grouped here as macros. This saves T_EX's memory when the same message is repeated and also lets them be edited easily.

\reledmac@warning Write a warning message.

```

126 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
127 %

```

`\reledmac@error` Write an error message.

```
128 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
129 %
```

```
\led@err@NumberingStarted30 \newcommand*\led@err@NumberingStarted}{%
d@err@NumberingNotStarted31 \reledmac@error{Numbering has already been started}{\@ehc}}
NumberingShouldHaveStarted32 \newcommand*\led@err@NumberingNotStarted}{%
133 \reledmac@error{Numbering was not started}{\@ehc}}
134 \newcommand*\led@err@NumberingShouldHaveStarted}{%
135 \reledmac@error{Numbering should already have been started}{\@ehc}}
136 %
```

```
d@err@edtextoutsidestart37 \newcommand*\led@err@edtextoutsidestart}{%
138 \reledmac@error{\string\edtext\space outside numbered paragraph (\...pstart
\pend)}{\@ehc}}%
139 %
```

```
\led@mess@NotesChanged40 \newcommand*\led@mess@NotesChanged}{%
141 \typeout{reledmac reminder: }%
142 \typeout{ The number of the footnotes in this section
143 has changed since the last run.}%
144 \typeout{ You will need to run LaTeX two more times
145 before the footnote placement}%
146 \typeout{ and line numbering in this section are
147 correct.}}
148 %
```

```
\led@mess@SectionContinued49 \newcommand*\led@mess@SectionContinued}[1]{%
150 \message{Section #1 (continuing the previous section)}}
151 %
```

```
d@err@LineationInNumbered52 \newcommand*\led@err@LineationInNumbered}{%
153 \reledmac@error{You can't use \string\lineation\space within
154 a numbered section}{\@ehc}}
155 %
```

```
\led@warn@BadLineation56 \newcommand*\led@warn@BadLineation}{%
led@warn@BadLinenummargin57 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp58 \newcommand*\led@warn@BadLinenummargin}{%
led@warn@BadSublockdisp59 \reledmac@warning{Bad \string\linenummargin\space argument}}
160 \newcommand*\led@warn@BadLockdisp}{%
161 \reledmac@warning{Bad \string\lockdisp\space argument}}
162 \newcommand*\led@warn@BadSublockdisp}{%
163 \reledmac@warning{Bad \string\sublockdisp\space argument}}
164 %
```

```

\led@warn@NoFile65 \newcommand*{\led@warn@NoFile}[1]{%
166 \reledmac@warning{File `#1' not found}}
167 %

\led@warn@LineFileObsolete68 \newcommand*{\led@warn@Obsolete}[1]{%
169 \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
170 %

\led@warn@BadAdvancelineSubline71 \newcommand*{\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine72 \reledmac@warning{\string\advanceline\space produced a sub-line
173 number less than zero.}}
174 \newcommand*{\led@warn@BadAdvancelineLine}{%
175 \reledmac@warning{\string\advanceline\space produced a line
176 number less than zero.}}
177 %

\led@warn@BadSetline78 \newcommand*{\led@warn@BadSetline}{%
\led@warn@BadSetlinenum79 \reledmac@warning{Bad \string\setline\space argument}}
180 \newcommand*{\led@warn@BadSetlinenum}{%
181 \reledmac@warning{Bad \string\setlinenum\space argument}}
182 %

\led@err@PstartNotNumbered83 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart84 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered85 numbered section %
\led@err@PendNoPstart86 (\string\...beginnumbering\string\endnumbering)}{\@ehc}}%
\led@err@AutoparNotNumbered87 \newcommand*{\led@err@PstartInPstart}{%
\led@err@NumberingWithoutPstart88 \reledmac@error{\string\pstart\space encountered while another
189 \string\pstart\space was in effect}{\@ehc}}
190 \newcommand*{\led@err@PendNotNumbered}{%
191 \reledmac@error{\string\pend\space must be used within a
192 numbered section}{\@ehc}}
193 \newcommand*{\led@err@PendNoPstart}{%
194 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
195 \newcommand*{\led@err@AutoparNotNumbered}{%
196 \reledmac@error{\string\autopar\space must be used within a
197 numbered section}{\@ehc}}
198 \newcommand*{\led@err@NumberingWithoutPstart}{%
199 \reledmac@error{\string\beginnumbering...\string\endnumbering\space
without \string\pstart}{\@ehc}}%
200 %

```

```

\led@warn@BadAction201 \newcommand*{\led@warn@BadAction}{%
202   \reledmac@warning{Bad action code, value \next@action.}}
203   %

\led@warn@DuplicateLabel204 \newcommand*{\led@warn@DuplicateLabel}[1]{%
\ppLabelOutSecondArgEdtext205   \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined206   \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined207 }%
208   \newcommand*{\led@warn@AppLabelOutSecondArgEdtext}[1]{%
209     \reledmac@warning{\string\applabel\space outside of the second argument
of an \string\edtext\space `#1' on page \thepage.}}%
210   \newcommand*{\led@warn@RefUndefined}[1]{%
211     \G@refundefinedtrue%
212     \reledmac@warning{Reference `#1' on page \thepage\space undefined.%
213       Using `000'.}%
214     \@latex@warning{Reference `#1' undefined\on@line}%
215   }%
216   \newcommand*{\led@warn@pairRefUndefined}[1]{%
217     \G@refundefinedtrue%
218     \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \thepage\
space undefined.
219       Using `??'.}%
220     \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
221   }
222   %

\led@warn@NoMarginpars223 \newcommand*{\led@warn@NoMarginpars}{%
224   \reledmac@warning{You can't use \string\marginpar\space in numbered text
}}
225   %

\led@warn@BadSidenotemargin226 \newcommand*{\led@warn@BadSidenotemargin}{%
227   \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
228   %

\led@warn@NoIndexFile229 \newcommand*{\led@warn@NoIndexFile}[1]{%
230   \reledmac@warning{Undefined index file #1}}
231   %

\led@warn@SeriesStillExist232 \newcommand{\led@warn@SeriesStillExist}[1]{%
233   \reledmac@warning{Series #1 is still existing !}%
234 }%
235   %

```

```

\led@err@BadAction236 \newcommand*{\led@err@StanzaIndentNotDefined}{%
237 \reledmac@error{You have not defined the indentation for the line \number
\stanza@count}{\@ehc}}%
238 %

```

```

\led@err@ManySidenotes239 \newcommand{\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes240 \ifledRcol{%
\led@err@ManyRightnotes241 \reledmac@warning{\itemcount@space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
242 \else%
243 \reledmac@warning{\itemcount@space sidenotes on line \the\line@num\
space p. \the\page@num}%
244 \fi%
245 }%
246 \newcommand{\led@err@ManyLeftnotes}{%
247 \ifledRcol{%
248 \reledmac@warning{\itemcount@space leftnotes on line \the\line@numR\
space p. \the\page@numR}%
249 \else%
250 \reledmac@warning{\itemcount@space leftnotes on line \the\line@num\
space p. \the\page@num}%
251 \fi%
252 }%
253 \newcommand{\led@err@ManyRightnotes}{%
254 \ifledRcol{%
255 \reledmac@warning{\itemcount@space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
256 \else%
257 \reledmac@warning{\itemcount@space rightnotes on line \the\line@num\
space p. \the\page@num}%
258 \fi%
259 }%
260 %

```

```

\led@err@TooManyColumns261 \newcommand*{\led@err@TooManyColumns}{%
\led@err@UnequalColumns262 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn263 \newcommand*{\led@err@UnequalColumns}{%
\led@err@HighEndColumn264 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns265 in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
266 \newcommand*{\led@err@LowStartColumn}{%
267 \reledmac@error{Start column is too low}{\@ehc}}
268 \newcommand*{\led@err@HighEndColumn}{%
269 \reledmac@error{End column is too high}{\@ehc}}
270 \newcommand*{\led@err@ReverseColumns}{%
271 \reledmac@error{Start column is greater than end column}{\@ehc}}
272 %

```



```

endnotes@outsidenumbering 73 \newcommand{\led@err@toendnotes@outsidenumbering}{%
274 \reledmac@error{\string\toendnotes\space and related commands must be
called inside a numbered text (\string\...beginnumbering\string\endnumbering
)}{\@ehc}%
275 }%
276 %

err@EdtextWithoutFootnote 77 \newcommand{\led@err@EdtextWithoutFootnote}{%
278 \reledmac@error{edtext without Xfootnote. Check syntax}{\@ehc}%
279 }%
280 %

FootnoteNotInSecondArgEdtext 81 \newcommand{\led@err@FootnoteNotInSecondArgEdtext}[1]{%
282 \reledmac@error{#1 footnote outside of the second argument of an edtext.
Check syntax}{\@ehc}%
283 }%
284 %

error@PackageAfterEledmac 85 \newcommand{\led@error@PackageAfterEledmac}[1]{%
286 \reledmac@error{#1 must be loaded before reledmac}{\@ehc}%
287 }%
288 %

error@fail@patch@@makecol 89 \newcommand{\led@error@fail@patch@@makecol}{%
290 \reledmac@error{Fail to patch \string@makecol\space command}{\@ehc}%
291 }%
292 %

error@fail@patch@@reinserts 93 \newcommand{\led@error@fail@patch@@reinserts}{%
294 \reledmac@error{Fail to patch \string@reinserts\space command}{\@ehc}%
295 }%
296 %

error@fail@patch@@docclearpage 97 \newcommand{\led@error@fail@patch@@docclearpage}{%
298 \reledmac@error{Fail to patch \string@docclearpage\space command}{\@ehc}%
299 }%
300 %

error@fail@patch@@iiiminipage 01 \newcommand{\led@error@fail@patch@@iiiminipage}{%
302 \reledmac@error{Fail to patch \string@iiiminipage\space command}{\@ehc}%
303 }%
304 %

```

```

\led@error@fail@patch@endminipage05 \newcommand{\led@error@fail@patch@endminipage}{%
306   \reledmac@error{Failed to patch the \string\endminipage\space command}{\
    @ehc}%
307 }%
308 %

```

```

\led@error@fail@patch@endminipage09 \newcommand{\led@error@fail@patch@makeindex}{%
310   \reledmac@error{Failed to patch the \string\makeindex\space command}{\
    @ehc}%
311 }%
312 %

```

```

\led@warn@edinde@outsidenumbering13 \newcommand{\led@warn@edinde@outsidenumbering}{%
314   \reledmac@warning{\string\edindex\space called outside of \string\
    ...beginnumbering\string\endnumbering. \MessageBreak Automatically switched
    to \string\index.}%
315 }%
316 %

```

```

\led@warning@hsizeX@deprecated17 \newcommand{\led@warning@hsizeX@deprecated}{%
318   \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
    widthX\space instead.}%
319 }%
320 %

```

```

\led@warning@Xhsize@deprecated21 \newcommand{\led@warning@Xhsize@deprecated}{%
322   \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
    Xwidth\space instead.}%
323 }%
324 %

```

```

\led@warning@msdatawithoutstop25 \newcommand{\led@warning@msdatawithoutstop}{%
326   \reledmac@warning{\string\msdata\space without corresponding \string\
    stopmsdata}%
327 }%
328 %

```

```

\led@warning@preXnotes@deprecated29 \newcommand{\led@warning@preXnotes@deprecated}{%
330   \reledmac@warning@preXnotes@deprecated%
331 }%
332 %

```

II.8 Gobbling

Here, we define some commands which gobble their arguments.

```
\@gobblethree33 \providecommand*\@gobblethree}[3]{}
\@gobblefour34 \providecommand*\@gobblefour}[4]{}
\@gobbleseven35 \providecommand*\@gobbleseven}[7]{}
336 %
```

II.9 Miscellaneous commands

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

```
337 \ifl@dfinal
338   \newcommand*\showlemma[1]{#1}
339 \else
340   \newcommand*\showlemma[1]{\underline{#1}}
341 \fi
342
343 %
```

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

Initialize it as `\empty`.

```
344 \let\linenumberlist=\empty
345
346 %
```

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

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

```
347 \newcount\@l@tempcnta \newcount\@l@tempcntb
348 %
```

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
349 \ifp@rtedL \newif\ifl@dpairing
350 \l@dnumprintstartsl \newif\ifl@dpaging%
351 \l@dnumprintstartsl \newif\ifl@dprintingpages%
352 \l@dnumprintstartsl \newif\ifl@dprintingcolumns%
353 \l@dnumprintstartsl \newif\ifp@rtedL
```

```

354 \newcount\l@dnumpstartsL
355 %

```

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

```

356 \newif\ifledRcol
357 \newif\ifledRcol@
358 %

```

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

```

359 \newif\ifnumberingR
360 %

```

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

```

361 \newif\ifXnote@%
362 %

```

II.11 Booleans provided by other optional packages which are required in any case

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

```

363 \providebool{indtl@innote}%
364 \providebool{indtl@notenumber}%
365 %

```

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.

```

366 \newcount\section@num
367 \section@num=0
368 \let\extensionchars=\empty
369 %

```

`\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.

```

370 \newif\ifnumbering
371 %

```

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

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

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

```

372 \newcommand*{\beginnumbering}{%
373   \ifnumbering
374     \led@err@NumberingStarted
375   \endnumbering
376 \fi
377 \global\numberingtrue
378 \global\advance\section@num \@ne
379 \initnumbering@reg
380 \message{Section \the\section@num }%
381 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
382 \l@dend@stuff
383 \setcounter{pstart}{1}
384 \ifl@dpairing
385   \global\l@dnumstartsL \z@
386   \global\pst@rtedLfalse
387 %

```

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.

```

388 \else
389 \begingroup
390 \global\@afterindenttrue%In order to reestablish normal feature if the \
beginning was not here
391 \initnumbering@quote
392 \ifwidthliketwocolumns%
393 \csuse{setwidthliketwocolumns@\columns@position}%
394 \csuse{setpositionliketwocolumns@\columns@position}%
395 \fi%
396 \fi
397 \gdef\eled@sections@{}%
398 \if@noeled@sec\else%
399 \makeatletter%
400 \InputIfFileExists%
401 {\l@auxdir\jobname.eledsec\the\section@num}%
402 {}%
403 {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@num}}%
404 \makeatother%
405 \immediate\openout\eled@sectioning@out=\l@auxdir\jobname.eledsec\the\
section@num\relax%
406 \fi%
407 }
408 \newcommand*\initnumbering@reg{%
409 \global\pst@rtedLfalse
410 \global\l@dnumstartsL \z@
411 \global\absline@num \z@
412 \gdef\normal@page@break{}
413 \gdef\l@prev@pb{}
414 \gdef\l@prev@nopb{}
415 \global\line@num \z@
416 \global\subline@num \z@
417 \global\@lock \z@
418 \global\sub@lock \z@
419 \global\sublines@false
420 \global\let\next@page@num=\relax
421 \global\let\sub@change=\relax
422 \resetprevline@
423 \resetprevpage@num
424 \global\stopmsdata@inserted@true%
425 \global\let\@msdata@list\relax%
426 \global\csundef{\@msdata@\add@msd@c @data}%
427 }
428
429 %

```

\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.

```

430 \def\endnumbering{%
431   \ifnumbering
432     \global\numberingfalse
433     \normal@pars
434     \ifnum\l@dnumstart=L=0%
435       \led@err@NumberingWithoutPstart%
436     \fi%
437     \ifl@dpairing
438       \global\pst@rtedLfalse
439     \else
440       \ifx\insertlines@list\empty\else
441         \global\noteschanged@true
442       \fi
443       \ifx\line@list\empty\else
444         \global\noteschanged@true
445       \fi
446     \fi
447     \ifnoteschanged@
448       \led@mess@NotesChanged
449     \fi
450   \else
451     \led@err@NumberingNotStarted
452   \fi
453   \autoparfalse
454   \if@noeled@sec\else%
455     \immediate\closeout\eled@sectioning@out%
456   \fi%
457   \ifl@dpairing\else
458     \global\l@dnumstart=L=\z@%
459   \endgroup
460 \fi
461 }
462 %

```

\pausenumbering The \pausenumbering macro is just the same as \endnumbering, but with the **\resumenumbering** \ifnumbering flag set to true, to show that numbering continues across the gap.²⁴

```

463 \newcommand{\pausenumbering}{%
464   \ifautopar\global\autopar@pausetrue\fi%
465   \endnumbering\global\numberingtrue}
466 %

```

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

²⁴Peter Wilson's thanks to Wayne Sullivan, who suggested the idea behind these macros.

that no check is made by `\resumenumbering` to ensure that `\pausenumbering` was actually invoked.

```

467 \newcommand*{\resumenumbering}{%
468   \ifnumbering
469     \ifautopar@pause\autopar\fi
470     \global\pstartedittrue
471     \global\advance\section@num \@ne
472     \led@mess@SectionContinued{\the\section@num}%
473     \line@list@stuff{\jobname.\extensionchars\the\section@num}%
474     \l@dend@stuff
475     \ifl@dpairing\else%
476       \begingroup%
477       \initnumbering@quote%
478       \ifwidthliketwocolumns%
479         \csuse{setwidthliketwocolumns@}\columns@position}%
480         \csuse{setpositionliketwocolumns@}\columns@position}%
481       \fi%
482     \fi%
483     \ifcontinuousnumberingwithcolumns%
484       \ifdefined\line@numR%
485         \ifnum\line@numR>\line@num%
486           \expandafter\setlinenum\expandafter{\the\line@numR}%
487         \fi%
488         \ifnum\last@page@numR>\last@page@num%
489           \global\last@page@num=\last@page@numR%
490         \fi%
491       \fi%
492     \fi%
493   \else
494     \led@err@NumberingShouldHaveStarted
495     \endnumbering
496     \beginnumbering
497   \fi}
498
499
500 %

```

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 \TeX 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.

```
501 \newcommand*\list@create}[1]{%
502   \global\let#1=\empty%
503 }%
504 %
```

`\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 .

```
505 \newcommand*\list@clear}[1]{%
506   \global\let#1=\empty%
507 }
508 %
```

`\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`.

```
509 \newtoks\led@toksa \newtoks\led@toksb
510 \global\led@toksa={\}
511 \long\def\xright@appenditem#1\to#2{%
512   \global\led@toksb=\expandafter{#2}%
513   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
514   \global\led@toksb={}}
515 %
```

`\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`.

```
516 \long\def\xleft@appenditem#1\to#2{%
517   \global\led@toksb=\expandafter{#2}%
518   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
519   \global\led@toksb={}}
520 %
```

`\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.

```

521 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
522 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
523
524 %

```

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.

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

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

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

```

525 \newif\ifbypage@
526 \newif\ifbypstart@
527 %

```

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@R28 \newif\ifbypage@R
\ifbypstart@R29 \newif\ifbypstart@R
530 %

```

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

```

531 \newcommand*{\lineation}[1]{
532 %

```

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

```

533 \ifnumbering
534 \led@err@LineationInNumbered
535 \else
536 %

```

If the argument is page.

```

537 \def\@tempa{#1}\def\@tempb{page}%
538 \ifx\@tempa\@tempb
539     \global\bypage@true
540     \global\bypstart@false
541     \unless\ifnocritical@%
542         \Xpstart[] [false]%
543     \fi%
544 %

```

If the argument is pstart.

```

545 \else
546     \def\@tempb{pstart}%
547     \ifx\@tempa\@tempb
548         \global\bypage@false
549         \global\bypstart@true
550         \unless\ifnocritical@%
551             \Xpstart%
552         \fi%
553 %

```

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

```

554 \else
555     \def\@tempb{section}
556     \ifx\@tempa\@tempb
557         \global\bypage@false
558         \global\bypstart@false
559         \unless\ifnocritical@%
560             \Xpstart[] [false]%
561         \fi%
562 %

```

In other case, it is an error.

```

563 \else
564     \led@warn@BadLineation
565 \fi
566 \fi
567 \fi
568 \fi}}
569 %

```

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`

```

570 \newcount\line@margin%
571 \newcount\line@margin@columns%Only for parallel typesetting

```

```

572 \line@margin@columns=\m@ne%
573
574 \newcommand*{\linenummargin}[1]{%
575   \l@getline@margin{#1}%
576   \ifnum\@l@tempcntb>\m@ne
577     \ifledRcol
578       \global\line@marginR=\@l@tempcntb
579       \led@warn@setting@in@rightside{\linenummargin}%
580     \else
581       \global\line@margin=\@l@tempcntb
582     \fi
583   \fi}}
584
585 \newcommand*{\l@getline@margin}[1]{%
586   \def\@tempa{#1}\def\@tempb{left}%
587   \ifx\@tempa\@tempb
588     \@l@tempcntb \z@
589   \else
590     \def\@tempb{right}%
591     \ifx\@tempa\@tempb
592       \@l@tempcntb \@ne
593     \else
594       \def\@tempb{outer}%
595       \ifx\@tempa\@tempb
596         \@l@tempcntb \tw@
597       \else
598         \def\@tempb{inner}%
599         \ifx\@tempa\@tempb
600           \@l@tempcntb \thr@@
601         \else
602           \led@warn@BadLinenummargin
603           \@l@tempcntb \m@ne
604         \fi
605       \fi
606     \fi
607   \fi}
608
609 %

```

V.3 Line number initialization and increment

`\c@firstlinenum`
`\c@linenumincrement`

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

```

610 \newcounter{firstlinenum}
611 \setcounter{firstlinenum}{5}

```

```

612 \newcounter{linenumincrement}
613 \setcounter{linenumincrement}{5}
614 %

```

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

```

615 \newcounter{firstsublinenum}
616 \setcounter{firstsublinenum}{5}
617 \newcounter{sublinenumincrement}
618 \setcounter{sublinenumincrement}{5}
619
620 %

```

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

```

\linenumincrement
\firstsublinenum
\sublinenumincrement
621 \newcommand*{\firstlinenum}[1]{%
622 \ifledRcol%
623 \setcounter{firstlinenumR}{#1}%
624 \led@warn@setting@in@rightside{\firstlinenum}%
625 \else%
626 \setcounter{firstlinenum}{#1}%
627 \fi%
628 }
629 \newcommand*{\linenumincrement}[1]{%
630 \ifledRcol%
631 \setcounter{linenumincrementR}{#1}%
632 \led@warn@setting@in@rightside{\linenumincrement}%
633 \else%
634 \setcounter{linenumincrement}{#1}%
635 \fi%
636 }
637 \newcommand*{\firstsublinenum}[1]{%
638 \ifledRcol%
639 \setcounter{firstsublinenumR}{#1}%
640 \led@warn@setting@in@rightside{\firstsublinenum}%
641 \else%
642 \setcounter{firstsublinenum}{#1}%
643 \fi%
644 }
645 \newcommand*{\sublinenumincrement}[1]{%
646 \ifledRcol%
647 \setcounter{sublinenumincrementR}{#1}%
648 \led@warn@setting@in@rightside{\sublinenumincrement}%
649 \else%
650 \setcounter{sublinenumincrement}{#1}%
651 \fi%
652 }
653 }
654

```

655 %

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.

```

656 \newcount\lock@disp
657 \newcommand{\lockdisp}[1]{%
658   \l@dgetlock@disp{#1}%
659   \ifnum\l@dtempcntb>\m@ne
660     \global\lock@disp=\l@dtempcntb
661   \else
662     \led@warn@BadLockdisp
663   \fi}}
664 \newcommand*{\l@dgetlock@disp}[1]{
665   \def\@tempa{#1}\def\@tempb{first}%
666   \ifx\@tempa\@tempb
667     \l@dtempcntb \z@
668   \else
669     \def\@tempb{last}%
670     \ifx\@tempa\@tempb
671       \l@dtempcntb \@ne
672     \else
673       \def\@tempb{all}%
674       \ifx\@tempa\@tempb
675         \l@dtempcntb \tw@
676       \else
677         \l@dtempcntb \m@ne
678       \fi
679     \fi
680   \fi}
681
682 %

```

`\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.

```

683 \newcount\sublock@disp
684 \newcommand{\sublockdisp}[1]{%
685   \l@dgetlock@disp{#1}%
686   \ifnum\l@dtempcntb>\m@ne
687     \global\sublock@disp=\l@dtempcntb
688   \else
689     \led@warn@BadSublockdisp
690   \fi}}

```

```
691
692 %
```

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`.

`\linenumr@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`

```
693 \newcommand*{\linenumberstyle}[1]{%
694   \def\linenumrep##1{\@nameuse{@#1}{##1}}
695 \newcommand*{\sublinenumberstyle}[1]{%
696   \def\sublinenumrep##1{\@nameuse{@#1}{##1}}
697 %
```

Initialise the number styles to arabic.

```
698 \linenumberstyle{arabic}
699 \let\linenumr@p\linenumrep
700 \sublinenumberstyle{arabic}
701 \let\sublinenumr@p\sublinenumrep
702
703 %
```

V.6 Line number printing

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

`\rightlinenum`

`\linenumsep`

`\numlabfont`

`\ledlinenum`

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

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

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

```
704 \newlength{\linenumsep}
705 \setlength{\linenumsep}{1pc}
706 \newcommand*{\numlabfont}{\normalfont\scriptsize}
```

```

707 \newcommand*{\ledlinenum}{%
708   \bgroup%
709   \ifluatex%
710     \texdir TLT%
711   \fi%
712   \numlabfont\linenumrep{\line@num}%
713   \ifsublines@
714     \ifnum\subline@num>0\relax
715       \unskip%
716       \Xsublinesep@side%
717       \sublinenumrep{\subline@num}%
718     \fi
719   \fi%
720   \egroup%
721 }%
722
723 \newcommand*{\leftlinenum}{%
724   \ledlinenum
725   \kern\linenumsep}
726 \newcommand*{\rightlinenum}{%
727   \kern\linenumsep
728   \ledlinenum}
729
730 %

```

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`.

```

731 \newcount\line@num
732 %

```


`\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.

```
733 \newcount\subline@num
734 %
```

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

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

```
735 \newif\ifsublines@
736 %
```

`\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.

```
737 \newcount\absline@num
738 %
```

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.

```
739 \newcount\@lock
740 \newcount\sub@lock
741 %
```

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/cmr/m/n.
```

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

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

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

Any number greater than -1000 is a page-start action: the line number associated with it is the first line on a page, and the action number is the page number. (The cutoff of -1000 is chosen because negative page-number values are used by some

macro packages; we assume that page-number values less than -1000 are not common.) Page-start action codes are added to the list by the `\page@action` macro, which is (indirectly) triggered by the workings of the `\page@start` macro; that macro should always be called in the output routine, just before the page contents are assembled. `Eledmac` calls it in `\pagecontents`.

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

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

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

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

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

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

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

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

Here are the commands to create these lists:

```

742 \list@create{\line@list}
743 \list@create{\insertlines@list}
744 \list@create{\actionlines@list}
745 \list@create{\actions@list}
746
747 %
```

<code>\page@num</code>	We will need some counts while we read the line-list, for the page number and the ending
<code>\endpage@num</code>	page, line, and sub-line numbers. Some of these will be used again later on, when we
<code>\endline@num</code>	are acting on the data in our list macros.
<code>\endsubline@num</code>	
748	<code>\newcount\page@num</code>
749	<code>\newcount\endpage@num</code>
750	<code>\newcount\endline@num</code>
751	<code>\newcount\endsubline@num</code>
752	<code>%</code>
<code>\ifnoteschanged@</code>	If the number of the footnotes in a section is different from what it was during the last
<code>\noteschanged@true</code>	run, or if this is the very first time you've run \LaTeX , on this file, the information from
<code>\noteschanged@false</code>	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 <code>\ifnoteschanged@</code> flag is set if such a change in the number of notes is discovered
	at any point.
753	<code>\newif\ifnoteschanged@</code>
754	<code>%</code>
<code>\resetprevline@</code>	Inside the apparatus, at each note, the line number is stored in a macro called
	<code>\prevlineX</code> , where X is the letter of the current series. This macro is called when
	using <code>\Xnumberonlyfirstinline</code> . This macro must be reset at the same time as the
	line number. The <code>\resetprevline@</code> does this resetting for every series.
<code>\resetprevline@</code>	
55	<code>\newcommand*\resetprevline@{%</code>
756	<code>\def\do##1{\global\csundef{prevline##1}}%</code>
757	<code>\dolistloop{\@series}%</code>
758	<code>}</code>
759	<code>%</code>
<code>\resetprevpage@num</code>	Inside the apparatus, at each note, the page number is stored in a macro called
	<code>\prevpageX@num</code> , where X is the letter of the current series. This macro is called when
	using <code>\Xparafootsep</code> or <code>\parafootsepX</code> . This macro must be reset at the beginning
	of each numbered section. The <code>\resetprevpage@</code> command resets this macro for every
	series.
<code>\resetprevpage@</code>	
60	<code>\newcommand*\resetprevpage@num{%</code>
761	<code>\def\do##1{%</code>
762	<code>\ifcsdef{prevpage##1@num}{%</code>
763	<code>\global\csname prevpage##1@num\endcsname=\z@%</code>
764	<code>\global\csname prevpage##1@numR\endcsname=\z@%</code>
765	<code>}%</code>
766	<code>{}%</code>
767	<code>\ifcsdef{##1prevpage@num}{%</code>

```

768 \global\csname ##1prevpage@num\endcsname=\z@%
769 \global\csname ##1prevpage@numR\endcsname=\z@%
770 }%
771 {}%
772 }%
773 \dolistloop{\@series}%
774 }
775 %

```

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.

```

776 \newread\@inputcheck
777 \newcommand*{\read@linelist}[1]{%
778 \ifledRcol%
779 \list@clearing@regR%
780 \else%
781 \list@clearing@reg%
782 \fi%
783 %

```

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).

```

784 \list@clear{\maxlinesinpar@list}
785 %

```

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

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

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

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

```

786 \get@linelistfile{#1}%
787 \ifcontinuousnumberingwithcolumns
788 \global\page@numR=\page@numR\relax
789 \global\last@page@numR=\last@page@numR\relax

```

```

790     \global\page@num=\page@num\relax
791     \global\last@page@num=\last@page@num\relax
792 \fi
793 \@stopmsd%Security if last \endms{} is forgotten
794 \unless\ifledRcol%Get the last line of the last page
795     \csnumgdef{@lastabsline@forpage@}\the\page@num}{\the\absline@num}%
796     \csnumgdef{@lastline@forpage@}\the\page@num}{\the\line@num}%
797 \else%
798     \csnumgdef{@lastabsline@forpageR@}\the\page@numR}{\the\absline@numR}%
799     \csnumgdef{@lastline@forpageR@}\the\page@numR}{\the\line@numR}%
800 \fi%
801 \endgroup
802 %

```

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.

```

803 \ifledRcol
804     \global\page@numR=\m@ne
805     \ifx\actionlines@listR\empty
806         \gdef\next@actionlineR{1000000}%
807     \else
808         \glp\actionlines@listR\to\next@actionlineR
809         \glp\actions@listR\to\next@actionR
810     \fi
811 \else
812     \global\page@num=\m@ne
813     \ifx\actionlines@list\empty
814         \gdef\next@actionline{1000000}%
815     \else
816         \glp\actionlines@list\to\next@actionline
817         \glp\actions@list\to\next@action
818     \fi
819 \fi
820 }
821 %

```

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

```

822 \newcommand*{\list@clearing@reg}{%
823     \list@clear{\line@list}%
824     \list@clear{\insertlines@list}%
825     \list@clear{\actionlines@list}%
826     \list@clear{\actions@list}%
827     \list@clear{\linesinpar@listL}%
828     \list@clear{\linesonpage@listL}%
829 }%
830 %

```

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

```

831 \newcommand*\get@linelistfile}[1]{%
832   \InputIfFileExists{\l@auxdir#1}{%
833     \global\noteschanged@false
834     \begingroup
835       \catcode`\[=1 \catcode`\]=2
836       \makeatletter \catcode`\^^M=9}{%
837     \led@warn@NoFile{\l@auxdir#1}%
838     \global\noteschanged@true
839     \begingroup}%
840 }
841
842 %

```

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. 19 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.

```

843 \newcommand{\line@list@version}[1]{%
844   \IfStrEq{#1}{\this@line@list@version}%
845   {}%

```

```

846 {\ifledRcol%
847   \led@warn@0obsolete{\jobname.\extensionchars\the\section@num}%
848   \else%
849   \led@warn@0obsoletef{\jobname.\extensionchars\the\section@num}%
850   \fi%
851   \endinput%
852 }%
853 }%
854 %

```

`\@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`.

```

855
856 \newcommand*{\@nl}[2]{%
857   \fix@page{#1}%
858   \ifledRcol%
859   \@nl@regR%
860   \else%
861   \@nl@reg%
862   \fi%
863 }
864 \newcommand*{\@nl@reg}{%
865   \ifx\l@dchset@num\relax \else
866   \advance\absline@num \@ne
867   \set@line@action
868   \let\l@dchset@num=\relax
869   \advance\absline@num \m@ne
870   \advance\line@num \m@ne
871   \fi
872 %

```

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

```

873   \advance\absline@num \@ne
874   \ifx\next@page@num\relax \else
875   \page@action

```



```

876         \let\next@page@num=\relax
877     \fi
878     \ifx\sub@change\relax \else
879         \ifnum\sub@change>\z@
880             \sublines@true
881         \else
882             \sublines@false
883         \fi
884         \sub@action
885         \let\sub@change=\relax
886     \fi
887 %

```

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

```

888     \ifcase\@lock
889     \or
890         \@lock \tw@
891     \or \or
892         \@lock \z@
893     \fi
894     \ifcase\sub@lock
895     \or
896         \sub@lock \tw@
897     \or \or
898         \sub@lock \z@
899     \fi
900 %

```

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

```

901     \ifsublines@
902         \ifnum\sub@lock<\tw@
903             \advance\subline@num \@ne
904         \fi
905     \else
906         \ifnum\@lock<\tw@
907             \advance\line@num \@ne \subline@num \z@
908         \fi
909     \fi}
910
911 %

```

`\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`.

```

912 \newcount\last@page@num
913 \last@page@num=-10000
914
915 \newcommand*{\fix@page}[1]{%
916     \ifledRcol

```

```

917 \ifnum #1=\last@page@numR
918 \else
919 \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
920 \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
921 \ifbypage@R
922 \line@numR \z@ \subline@numR \z@
923 \fi
924 \page@numR=#1\relax
925 \last@page@numR=#1\relax
926 \def\next@page@numR{#1}%
927 \fi
928 \else
929 \ifnum #1=\last@page@num
930 \else
931 \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
932 \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
933 \ifbypage@
934 \line@num \z@ \subline@num \z@
935 \fi
936 \page@num=#1\relax
937 \last@page@num=#1\relax
938 \def\next@page@num{#1}%
939 \listxadd{\normal@page@break}{\the\absline@num}
940 \fi
941 \fi}
942 %

```

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

```

943 \newcommand*{\@pend}[1]{}
944 \newcommand*{\@pendR}[1]{}
945 \newcommand*{\@lopL}[1]{}
946 \newcommand*{\@lopR}[1]{}
947
948 %

```

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

```

949 \newcommand*{\sub@on}{\ifsublines@
950 \let\sub@change=\relax
951 \else
952 \def\sub@change{1}%
953 \fi}
954 \newcommand*{\sub@off}{\ifsublines@
955 \def\sub@change{-1}%
956 \else

```

```

957 \let\sub@change=\relax
958 \fi}
959
960 %

```

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

```

961
962 \newcommand*{\@adv}[1]{%
963 \ifsublines@
964 \ifledRcol
965 \advance\subline@numR by #1\relax
966 \ifnum\subline@numR<\z@
967 \led@warn@BadAdvancelineSubline
968 \subline@numR \z@
969 \fi
970 \else
971 \advance\subline@num by #1\relax
972 \ifnum\subline@num<\z@
973 \led@warn@BadAdvancelineSubline
974 \subline@num \z@
975 \fi
976 \fi
977 \else
978 \ifledRcol
979 \advance\line@numR by #1\relax
980 \ifnum\line@numR<\z@
981 \led@warn@BadAdvancelineLine
982 \line@numR \z@
983 \fi
984 \else
985 \advance\line@num by #1\relax
986 \ifnum\line@num<\z@
987 \led@warn@BadAdvancelineLine
988 \line@num \z@
989 \fi
990 \fi
991 \fi
992 \set@line@action}
993
994 %

```

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

```

995
996 \newcommand*{\@set}[1]{%
997 \ifledRcol
998 \ifsublines@

```

```

999      \subline@numR=#1\relax
1000    \else
1001      \line@numR=#1\relax
1002    \fi
1003    \set@line@action
1004  \else
1005    \ifsublines@
1006      \subline@num=#1\relax
1007    \else
1008      \line@num=#1\relax
1009    \fi
1010    \set@line@action
1011  \fi}
1012
1013 %

```

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

```

1014
1015 \newcommand*{\l@d@set}[1]{%
1016   \ifledRcol
1017     \line@numR=#1\relax
1018     \advance\line@numR \@ne
1019     \def\l@dchset@num{#1}
1020   \else
1021     \line@num=#1\relax
1022     \advance\line@num \@ne
1023     \def\l@dchset@num{#1}
1024   \fi}
1025 \let\l@dchset@num\relax
1026
1027 %

```

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

```

1028
1029 \newcommand*{\page@action}{%
1030   \ifledRcol
1031     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1032     \xright@appenditem{\next@page@numR}\to\actions@listR
1033   \else
1034     \xright@appenditem{\the\absline@num}\to\actionlines@list
1035     \xright@appenditem{\next@page@num}\to\actions@list
1036   \fi}
1037 %

```

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

```

1038
1039 \newcommand*{\set@line@action}{%
1040   \ifl@edRcol
1041     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1042     \ifsublines@
1043       \l@l@tempcnta=-\subline@numR
1044     \else
1045       \l@l@tempcnta=-\line@numR
1046     \fi
1047     \advance\l@l@tempcnta by -5000\relax
1048     \xright@appenditem{\the\l@l@tempcnta}\to\actions@listR
1049   \else
1050     \xright@appenditem{\the\absline@num}\to\actionlines@list
1051     \ifsublines@
1052       \l@l@tempcnta=-\subline@num
1053     \else
1054       \l@l@tempcnta=-\line@num
1055     \fi
1056     \advance\l@l@tempcnta by -5000\relax
1057     \xright@appenditem{\the\l@l@tempcnta}\to\actions@list
1058   \fi}
1059 %

```

\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.

```

1060
1061 \newcommand*{\sub@action}{%
1062   \ifl@edRcol
1063     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1064     \ifsublines@
1065       \xright@appenditem{-1001}\to\actions@listR
1066     \else
1067       \xright@appenditem{-1002}\to\actions@listR
1068     \fi
1069   \else
1070     \xright@appenditem{\the\absline@num}\to\actionlines@list
1071     \ifsublines@
1072       \xright@appenditem{-1001}\to\actions@list
1073     \else
1074       \xright@appenditem{-1002}\to\actions@list
1075     \fi
1076   \fi}
1077 %

```

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

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

```

1078 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1079
1080 \newcommand*{\do@lockon}{%
1081   \ifx\next\lock@off
1082     \global\let\lock@off=\skip@lockoff
1083   \else
1084     \ifledRcol
1085       \do@lockonR
1086     \else
1087       \do@lockonL
1088     \fi
1089   \fi}
1090
1091
1092 \newcommand*{\do@lockonL}{%
1093   \xright@appenditem{\the\absline@num}\to\actionlines@list
1094   \ifsublines@
1095     \xright@appenditem{-1005}\to\actions@list
1096     \ifnum\sub@lock=\z@
1097       \sub@lock \@ne
1098     \else
1099       \ifnum\sub@lock=\thr@@
1100         \sub@lock \@ne
1101       \fi
1102     \fi
1103   \else
1104     \xright@appenditem{-1003}\to\actions@list
1105     \ifnum\@lock=\z@
1106       \@lock \@ne
1107     \else
1108       \ifnum\@lock=\thr@@
1109         \@lock \@ne
1110       \fi
1111     \fi
1112   \fi}
1113
1114 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1115 \newcommand*{\do@lockoffL}{%
1116   \xright@appenditem{\the\absline@num}\to\actionlines@list
1117   \ifsublines@
1118     \xright@appenditem{-1006}\to\actions@list
1119     \ifnum\sub@lock=\tw@

```

```

1120     \sub@lock \thr@@
1121     \else
1122     \sub@lock \z@
1123     \fi
1124 \else
1125 \xright@appenditem{-1004}\to\actions@list
1126 \ifnum\@lock=\tw@
1127     \@lock \thr@@
1128     \else
1129     \@lock \z@
1130     \fi
1131 \fi}
1132
1133 \newcommand*{\do@lockoff}{%
1134     \ifledRcol
1135     \do@lockoffR
1136     \else
1137     \do@lockoffL
1138     \fi}
1139 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1140 \global\let\lock@off=\do@lockoff
1141
1142 %

```

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

```

1143 \newcommand*{\n@num}{%
1144     \ifledRcol%
1145     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1146     \xright@appenditem{-1007}\to\actions@listR
1147     \else%
1148     \xright@appenditem{\the\absline@num}\to\actionlines@list%
1149     \xright@appenditem{-1007}\to\actions@list%
1150     \fi%
1151 }%
1152
1153 %

```

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

```

1154 \newcommand*{\n@num@stanza}{%
1155     \ifledRcol%
1156     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1157     \xright@appenditem{-1008}\to\actions@listR%
1158     \else%
1159     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1160     \xright@appenditem{-1008}\to\actions@list%
1161     \fi%

```

```
1162 }
1163 %
```

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

`\hidenumberingonleftpage`

`\hidenumberingonrightpage`

```
1164 \newif\ifl@dhidenumber
1165 \newcommand*{\hidenumbering}{%
1166   \ifl@dhidenumber%
1167     \write\linenum@outR{\string\hide@num}%
1168   \else%
1169     \write\linenum@out{\string\hide@num}%
1170   \fi%
1171 }%
1172 \newcommand*{\hide@num}{%
1173   \ifl@dhidenumber%
1174     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1175     \xright@appenditem{-1009}\to\actions@listR%
1176   \else%
1177     \xright@appenditem{\the\absline@num}\to\actionlines@listR%
1178     \xright@appenditem{-1009}\to\actions@listR%
1179   \fi%
1180 }
1181 \newcommand*{\hidenumberingonleftpage}{%
1182   \ifl@dhidenumber%
1183     \write\linenum@outR{\string\hide@num@left}%
1184   \else%
1185     \write\linenum@out{\string\hide@num@left}%
1186   \fi%
1187 }%
1188
1189 \newcommand*{\hide@num@left}{%
1190   \ifl@dhidenumber%
1191     \ifodd\page@numR\else%
1192       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1193       \xright@appenditem{-1009}\to\actions@listR%
1194     \fi%
1195   \else%
1196     \ifodd\page@num\else%
1197       \xright@appenditem{\the\absline@num}\to\actionlines@listR%
1198       \xright@appenditem{-1009}\to\actions@listR%
1199     \fi%
1200   \fi%
1201 }%
1202
1203 \newcommand*{\hidenumberingonrightpage}{%
1204   \ifl@dhidenumber%
1205     \write\linenum@outR{\string\hide@num@right}%
1206   \else%
```



```

1207 \write\linenum@out{\string\hide@num@right}%
1208 \fi%
1209 }%
1210
1211 \newcommand*{\hide@num@right}{%
1212 \ifledRcol%
1213 \ifodd\page@numR%
1214 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1215 \xright@appenditem{-1009}\to\actions@listR%
1216 \fi%
1217 \else%
1218 \ifodd\page@num%
1219 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1220 \xright@appenditem{-1009}\to\actions@list%
1221 \fi%
1222 \fi%
1223 }%
1224
1225 %

```

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

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

```

1226 \newcount\insert@count
1227 %

```

- #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.

```

1228 \newcommand*{\dummy@ref}[2]{#2}
1229 %

```

\@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.

```

1230 \newcommand*{\@ref}[2]{%
1231 \ifledRcol%
1232 \@ref@regR{#1}{#2}%
1233 \else%
1234 \@ref@reg{#1}{#2}%
1235 \fi%
1236 }%

```

```

1237 \newcommand*{\@ref@reg}[2]{%
1238   \global\insert@count=#1\relax
1239   \global\advance\@edtext@level by 1%
1240   \loop\ifnum\insert@count>\z@
1241     \xright@appenditem{\the\absline@num}\to\insertlines@list
1242     \global\advance\insert@count \m@ne
1243   \repeat
1244   %

```

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.

```

1245 \begingroup
1246   \let\@ref=\dummy@ref
1247   \let\@lopL=\gobble
1248   \let\page@action=\relax
1249   \let\sub@action=\relax
1250   \let\set@line@action=\relax
1251   \let\@lab=\relax
1252   \let\@lemma=\relax%
1253   \let\@sw\@gobblethree%
1254   #2
1255   \global\endpage@num=\page@num
1256   \global\endline@num=\line@num
1257   \global\endsubline@num=\subline@num
1258 \endgroup
1259 %

```

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

```

1260 \xright@appenditem%
1261   {\the\page@num|\the\line@num|%
1262    \ifsublines@ \the\subline@num \else 0\fi}%
1263   \the\endpage@num|\the\endline@num|%
1264   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1265 %

```

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

```

1266   \@ref@reg@parse{#2}%
1267 %

```

Decrease edtext level counter.

```

1268   \global\advance\@edtext@level by -1%
1269 }
1270 %

```

\@ref@reg@parse The \@ref@reg@parsearg command parses the second argument of a \@ref or the unique argument of \@ref@later written in the auxiliary fill.

First, create a list which stores every second argument of each \@sw in this lemma, at this level. Also set the boolean about the use of lemma in this edtext level to false.

```

1271 \newcommand{\@ref@reg@parse}[1]{%
1272   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
@edtext@level\endcsname}%
1273   \providebool{lemmacommand@\the\@edtext@level}%
1274   \boolfalse{lemmacommand@\the\@edtext@level}%
1275   %

```

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

```

1276   #1%
1277   %

```

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.

```

1278   \ifnum\@edtext@level>0%
1279   \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1280   \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level
}\}%
1281   \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
1282   \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}
1283   \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1284   \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
1285   \fi%
1286   %
1287 }
1288
1289 %

```

\ref@reg@later This macro is stored in the auxiliary file when using \edtextlater. It is used only to get the correct value for the \sameword tools.

```

1290 \newcommand{\@ref@later}[1]{%
1291   \global\advance\@edtext@level by \@ne%
1292   \ifledRcol%
1293     \@ref@reg@parseR{#1}%
1294   \else%
1295     \@ref@reg@parse{#1}%
1296   \fi%
1297   \global\advance\@edtext@level by -\@ne%
1298 }%
1299 %

```

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`.

```
1300 \newwrite\linenum@out
1301 %
```

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

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

```
1302 \newif\iffirst@linenum@out@
1303 \iffirst@linenum@out@true
1304 %
```

`\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.

```
1305 \newcommand{\this@line@list@version}{6}%
1306 %
```

`\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.

```
1307 \newcommand*{\line@list@stuff}[1]{%
1308 %
```

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

```
1309 \read@linelist{#1}%
1310 %
```

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.

```

1311 \iffirst@linenum@out@
1312   \immediate\closeout\linenum@out%
1313   \global\first@linenum@out@false%
1314   \immediate\openout\linenum@out=\l@auxdir#1\relax%
1315   \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}}%
1316   \ifl@dpaging%
1317     \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}}%
1318   \fi%
1319 \else
1320 %

```

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.

```

1321   \if@minipage%
1322     \leavevmode%
1323     \fi%
1324     \closeout\linenum@out%
1325     \openout\linenum@out=\l@auxdir#1\relax%
1326     \write\linenum@out{\string\line@list@version{\this@line@list@version}}
%
1327   \ifl@dpaging%
1328     \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
%
1329   \fi%
1330 \fi}
1331
1332 %

```

\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.

```

1333 \newcommand*{\new@line}{%
1334   \IfStrEq{\led@pb@setting}{after}%
1335   {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1336     {\xifinlist{\the\absline@num}{\normal@page@break}%
1337       {\numdef{\@next@page}{\c@page+\@ne}%
1338         \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%
1339       }%
1340     {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1341   }%
1342   {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1343 }%
1344 \IfStrEq{\led@pb@setting}{before}%
1345   {\numdef{\next@absline}{\the\absline@num+\@ne}%

```

```

1346 \xifinlist{\next@absline}{\l@prev@nopb}%
1347   {\xifinlist{\the\absline@num}{\normal@page@break}%
1348     {\numgdef{\nc@page}{\c@page+\@ne}%
1349       \write\linenum@out{\string\@nl[\nc@page][\nc@page]}%
1350       }%
1351       {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1352     }%
1353     {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1354   }%
1355   {}%
1356 \IfStrEqCase{\led@pb@setting}{\before}{\relax}{after}{\relax}}{\write\
linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1357 }
1358
1359 %

```

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

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

`\flag@end`

```

1360 \newif\if@noneed@Footnote%
1361
1362 \newcommand*{\flag@start}{%
1363   \ifledRcol%
1364     \edef\next{\write\linenum@outR{%
1365       \string\@ref[\the\insert@countR] []}%
1366     }
1367     \next%
1368     \ifnum\insert@countR<1%
1369       \if@noneed@Footnote\else%
1370         \led@err@EdtextWithoutFootnote%
1371       \fi%
1372     \else%
1373       \edef\next{\write\linenum@out{%
1374         \string\@ref[\the\insert@count] []}%
1375       }
1376       \next%
1377       \ifnum\insert@count<1%
1378         \if@noneed@Footnote\else%
1379           \led@err@EdtextWithoutFootnote%
1380         \fi%
1381       \fi%
1382     }
1383 \newcommand*{\flag@end}{%
1384   \ifledRcol%
1385     \write\linenum@outR[]}%
1386   \else%

```

```

1387 \write\linenum@out{}}%
1388 \fi}%
1389
1390
1391 %

```

`\flag@start@RTL` With Xe_{La}TeX, there is a problem when using RTL: the writing of a command in the numbered auxiliary files (.1, .2 etc) is reversed when the first argument of `\edtext` is typeset in one line, but it is **not** reversed when this first argument is typeset in two lines or more.²⁵

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

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

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

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

```

1392 \newcommand{\flag@start@RTL}{%
1393 \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1394 \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}{%
1395 {000}}%
1396 {}%
1397 {%
1398 \ifnumequal%
1399 {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}{%
1400 {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1401 {\flag@end}%
1402 {\flag@start}%
1403 }%
1404 }%
1405
1406 \newcommand{\flag@end@RTL}{%
1407 \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}%
1408 \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}{%

```

²⁵This problem is caused by the way Xe_{La}TeX manages right-to-left typesetting. David Carlisle explains it on <http://tex.stackexchange.com/a/333373/7712> and provides a potential solution, using `\vadjust`. However in some cases this adds spurious vertical spaces in `reledmac`. That is why we are using the solution explained below.

```

1409 {000}%
1410 {}%
1411 {%
1412 \ifnumequal%
1413 {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1414 {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1415 {\flag@start}%
1416 {\flag@end}%
1417 }%
1418 }%
1419 %

```

`\flag@start@later` `\flag@start@later` and `\flag@end@later`: these send the `\@ref@later` to the line-list file command to the line-list file

```

1420 \newcommand*{\flag@start@later}{%
1421 \ifledRcol%
1422 \write\linenum@outR{\string\@ref@later[]}%
1423 \else%
1424 \write\linenum@out{\string\@ref@later[]}%
1425 \fi%
1426 }%
1427 \newcommand{\flag@end@later}{%
1428 \ifledRcol%
1429 \write\linenum@outR{[]}%
1430 \else%
1431 \write\linenum@out{[]}%
1432 \fi%
1433 }
1434 %

```

`\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.

```

1435
1436
1437 \newcommand*{\startsub}{\dimen0\lastskip
1438 \ifdim\dimen0>0pt \unskip \fi
1439 \ifledRcol \write\linenum@outR{\string\sub@on}%

```



```

1440 \else      \write\linenum@out{\string\sub@on}%
1441 \fi
1442 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1443 \def\endsub{\dimen0\lastskip
1444 \ifdim\dimen0>0pt \unskip \fi
1445 \ifledRcol \write\linenum@outR{\string\sub@off}%
1446 \else      \write\linenum@out{\string\sub@off}%
1447 \fi
1448 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1449
1450 %

```

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

```

1451 \newcommand*{\advanceline}[1]{\leavevmode%
1452 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1453 \else      \write\linenum@out{\string\@adv[#1]}%
1454 \fi}%
1455 }
1456 %

```

\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.

```

1457
1458 \newcommand*{\setline}[1]{%
1459 \leavevmode%
1460 \ifnum#1<\z@
1461 \led@warn@BadSetline
1462 \else
1463 \ifledRcol \write\linenum@outR{\string\@set[#1]}%
1464 \else      \write\linenum@out{\string\@set[#1]}%
1465 \fi
1466 \fi}
1467
1468 %

```

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

```

1469
1470 \newcommand*{\setlinenum}[1]{%
1471 \ifnum#1<\z@
1472 \led@warn@BadSetlinenum
1473 \else
1474 \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1475 \else      \write\linenum@out{\string\l@d@set[#1]} \fi
1476 \fi}
1477
1478 %

```

`\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.

`\endlock`

```

1479
1480 \newcommand*{\startlock}{%
1481   \ifl@dskipnumber \write\linenum@outR{\string\lock@on}%
1482   \else           \write\linenum@out{\string\lock@on}%
1483   \fi}
1484 \def\endlock{%
1485   \ifl@dskipnumber \write\linenum@outR{\string\lock@off}%
1486   \else           \write\linenum@out{\string\lock@off}%
1487   \fi}
1488 %

```

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

`\ifl@dskipversenumber`

```

1489 \l@dskipnumbertrue \newif\ifl@dskipnumber
1490 \l@dskipnumberfalse \newif\ifl@dskipversenumber%
1491 \skipnumbering \newcommand*{\skipnumbering}{%
1492   \leavevmode%
1493   \ifl@dskipnumber%
1494     \ifinstanza%
1495       \write\linenum@outR{\string\n@num@stanza}%
1496     \else%
1497       \write\linenum@outR{\string\n@num}%
1498     \fi%
1499     \advanceline{-1}%
1500   \else%
1501     \ifinstanza%
1502       \write\linenum@out{\string\n@num@stanza}%
1503     \else%
1504       \write\linenum@out{\string\n@num}%
1505     \fi%
1506     \advanceline{-1}%
1507   \fi%
1508 }%
1509
1510 %

```

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

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

VI.1 `\edtext` itself

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

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

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

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

```
1511 \list@create{\end@lemmas}
1512 %
```

`\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.

```
1513 \newcommand{\dummy@edtext}[2]{#1}
1514 %
```

`\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`.

```
1515 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1516 %
```

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

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

`\morenoexpands`

The first class is font-changing macros. We suppress expansion for them by letting them become equal to zero.²⁶ This is done because we want to pass into our notes the generic commands to change to roman or whatever, and not their expansions that will ask for a particular style at a specified size. The notes may well be in a smaller font, so the command should be expanded later, when the note's environment is in effect.

A second sort to turn off includes a few of the accent macros. Most are not a problem: an accent that is expanded to an `\accent` command may be harder to read but it works just the same. The ones that cause problems are: those that use alignments— \TeX seems to

²⁶Since 'control sequences equivalent to characters are not expandable'—*The TeXbook*, answer to Exercise 20.14.

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.

The `\new@series` command also adds `\let\footnote(X)\@gobble` to the end of the `\no@expands` macro for the series $\langle X \rangle$.

(A related problem, not addressed by these two macros, is that of characters whose category code are changed by any of the macros used in the arguments to `\edtext`. Since the category codes are set when the arguments are scanned, macros that depend on changing them will not work. We have most often encountered this with characters that are made ‘active’ within text in some, but not all, of the languages used within the document. One way around the problem, if it takes this form, is to ensure that those characters are *always* active. Within languages that make no special use of them, their associated control sequences should simply return the proper character. A simpler solution is to avoid active characters, using Lua \TeX or Xe \LaTeX .)

```

1517 \newcommand*{\no@expands}{%
1518   \let\select@lemmafnt=0%
1519   \let\startsub=\relax   \let\endsub=\relax
1520   \let\startlock=\relax  \let\endlock=\relax
1521   \let\edlabel=\@gobble
1522   \let\setline=\@gobble  \let\advanceline=\@gobble
1523   \let\sameword\sameword@inedtext%
1524   \let\edtext=\dummy@edtext
1525   \l@dtabnoexpands
1526   \morenoexpands}
1527 \let\morenoexpands=\relax
1528
1529 %
```

\@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.

```

1530 \newcommand{\@tag}{}
1531 %
```

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

```
1532 \newcount\@edtext@level%
1533 \@edtext@level=0%
1534 %
```

`\ifedtext@secondarg@` This boolean is set to TRUE before reading the second argument of a `\edtext`. It is tested on some macro which must be executed only inside a second argument.

```
1535 \newif\ifedtext@secondarg%
1536 %
```

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

```
1537 \newcounter{edtext}
1538 \renewcommand{\theedtext}{\edtxt@\arabic{edtext}}%
1539 %
```

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

```
1540 \newcommand{\edtext}[2]{\leavevmode%
1541 %
```

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

```
1542 \ifnumberedpar%
1543 %
```

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

```
1544 \global\advance\@edtext@level by 1%
1545 %
```

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

We store the value for the current level in a global macro. So we have one macro by level of `\edtext`. That is required, because `\edtext` can contain `\edtext`.

```
1546 \stepcounter{edtext}%
1547 \csxdef{thisedtext@the\@edtext@level}{\theedtext}%
1548 %
```

By default, we do not use `\lemma`

```
1549 \global\@lemmacommand@false%
1550 %
```

```
1551 \begingroup%
1552 %
```

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`.

```

1553     \ifledRcol%
1554     \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1555         {\global\let\sw@inthisedtext\empty}%
1556         {\expandafter\gl@p\csname sw@list@edtextR@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1557     \else%
1558     \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1559         {\global\let\sw@inthisedtext\empty}%
1560         {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1561     \fi%
1562 %

```

`\@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.

```

1563     \global\renewcommand{\@tag}{%
1564         \no@expands #1%
1565     }%
1566 %

```

`\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`.

```

1567     \set@line%
1568 %

```

`\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`.

```

1569     \ifledRcol \global\insert@countR \z@%
1570     \else      \global\insert@count \z@ \fi%
1571 %

```

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.

```

1572 \edtext@secondarg@true%
1573 \ignorespaces #2\relax%
1574 \edtext@secondarg@false%
1575 %

```

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

```

1576 \if@RTL%
1577 \flag@start@RTL%
1578 \else%
1579 \flag@start%
1580 \fi%
1581 %

```

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

```

1582 \if@lemmacommand%
1583 \ifledRcol%
1584 \write\linenum@outR{\string\@lemma}%
1585 \else%
1586 \write\linenum@out{\string\@lemma}%
1587 \fi%
1588 \fi%
1589 %

```

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.

```

1590 \endgroup%
1591 \ifdef{\hypertarget}%
1592 {%
1593 \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}}}%
1594 \showlemma{#1}%
1595 \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}}}%
1596}%
1597{%
1598 \showlemma{#1}%
1599}%
1600 %

```

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.


```

1601 \ifx\end@lemmas\empty \else%
1602 \gl@p\end@lemmas\to\x@lemma%
1603 \x@lemma%
1604 \global\let\x@lemma=\relax%
1605 \fi%
1606 \if@RTL%
1607 \flag@end@RTL%
1608 \else%
1609 \flag@end%
1610 \fi%
1611 %

```

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`.

```

1612 \global\@noneed@Footnotefalse%
1613 \global\advance\@edtext@level by -1%
1614 \global\@lemmacommand@false%
1615 %

```

We also reset `\@beforeinsertofthisedtext`

```

1616 \global\let\@beforeinsertofthisedtext\relax%
1617 %

```

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

```

1618 \else%
1619 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidestart%
1620 \fi%
1621 }%
1622
1623
1624 %

```

`\@beforeinsertofthisedtext` `\@beforeinsertofthisedtext` is an internal macro. `reledmac` or `reledpar` can add in this macro any content required to be executed before doing any `\insert` related to a `\edtext`. Its content is `\let` equal to `\relax` at the end of every `\edtext`.

```

1625 \let\@beforeinsertofthisedtext\relax
1626 %

```

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

```

1627 \newif\ifnumberline
1628 \numberlinetrue
1629 %

```

\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.

```

1630 \newcommand*{\set@line}{%
1631   \ifl@edRcol
1632     \ifx\line@listR\empty
1633       \global\noteschanged@true
1634       \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1635     \else
1636       \gl@p\line@listR\to\@tempb
1637       \xdef\l@d@nums{\@tempb|\edfont@info}%
1638       \global\let\@tempb=\undefined
1639     \fi
1640   \else
1641     \ifx\line@list\empty
1642       \global\noteschanged@true
1643       \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1644     \else
1645       \gl@p\line@list\to\@tempb
1646       \xdef\l@d@nums{\@tempb|\edfont@info}%
1647       \global\let\@tempb=\undefined
1648     \fi
1649   \fi}
1650
1651 %

```

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

```

1652 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1653
1654 %

```

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. 127).

```

1655 \newcommand*{\lemma}[1]{%
1656   \global\@lemmacommand@true%
1657   \global\renewcommand{\@tag}{%
1658     \no@expands #1%
1659   }%
1660   \ignorespaces%
1661 }%
1662 %

```

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

```

1663 \newcommand{\@lemma}{%
1664   \booltrue{lemmacommand@the\@edtext@level}%
1665 }%
1666 %

```

\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.

```

1667 \newif\if@lemmacommand@%
1668 %

```

VI.3 Substitute line numbers

\linenum The \linenum macro can change any or all of the page and line numbers that are passed on to the notes.

As argument \linenum takes a set of seven parameters separated by vertical bars, in the format used internally for \l@d@nums (see V.9 p. 98): the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma. However, you can omit any parameters you do not want to change, and you can omit a string of vertical bars at the end of the argument. Hence \linenum{18|4|0|18|7|1|0} is an invocation that changes all the parameters, but \linenum{|3} only changes the starting line number, and leaves the rest unaltered.

We use \\ as an internal separator for the macro parameters.

```

1669 \newcommand*{\linenum}[1]{%
1670   \xdef\@tempa{#1|||||\\noexpand\\l@d@nums}%
1671   \global\let\l@d@nums=\empty
1672   \expandafter\line@set\@tempa|\\ignorespaces}
1673 %

```

\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.

```

1674 \def\line@set#1|#2\\#3|#4\\{%
1675   \gdef\@tempb{#1}%
1676   \ifx\@tempb\empty
1677     \l@d@add{#3}%
1678   \else
1679     \l@d@add{#1}%
1680   \fi
1681   \gdef\@tempb{#4}%
1682   \ifx\@tempb\empty\else
1683     \l@d@add{|}\line@set#2\\#4\\%
1684   \fi}
1685 %

```

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

```

1686 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1687
1688 %

```

VI.4 Lemma disambiguation

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

The general mechanism is the following:

- **At the first run**, each `\sameword` command increments an `etoolbox` counter the name of which contains the argument of the `\sameword` commands.
- Then this counter, associated with the argument of `\sameword` is stored with the `\@sw` command in the auxiliary file of the current `reledmac` section (the `.1`, `.2...` file).
- **When this auxiliary file is read at the second run**, different operations are achieved:
 1. Get the rank of each `\sameword` in a line (relative rank) from the rank of each `\sameword` in all the numbered section (absolute rank):
 - For each paired `\sameword` argument and absolute line number, a counter is defined. Its value corresponds to the number of times `\sameword{⟨argument⟩}` is called from the beginning of the lineation to the end of the current line. We also store the same data for the preceding absolute line number, if it does not have `\sameword{⟨argument⟩}`.
 - For each `\sameword` having the same argument, we subtract from its absolute rank the number stored for the paired `\sameword` argument and previous absolute line number. Consequently, we obtain the relative rank.

- See the following example which explains how, for same `\sameword`, absolute ranks are transformed to relative ranks.

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

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

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

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

In this example, all lemma information for `\edtext` is framed. The text in parenthesis is the content of critical notes associated to the preceding frame. As you can see, we have two 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 its `\edtext` level. So, in the previous example:
 - Critical notes (A) and (B) are associated with $\{1, 2, 2, 3\}$.
 - Critical note (C) is associated with $\{1, 2, 2, 3, 4, 3\}$.
 - Critical note (D) is associated with $\{5\}$.
 - Critical note (E) is associated with $\{5, 4\}$.
- At the second run, when a critical note is printed:
 - The `\sameword` command is let `\sameword@inedtext`.

- At each call of this `\sameword@inedtext`, we step to the next element of the list associated to the note. Let it be r .
- For the word marked by `\sameword`, we calculate how many time it is called in its line. To do it:
 - * We get the absolute line number of the current `\sameword`. This absolute line number was stored with a list of relative ranks for the current `\edtext`. That means, in the previous example, that if the absolute line number of `\edtext` was 1, that critical notes (A) and (B) were not associated with $\{1, 2, 2, 3\}$ but with $\{(1, 1), (2, 1), (2, 1), (3, 1)\}$. Such a method of knowing the absolute line number associated to a `\sameword` is required because a `\edtext` can overlap many lines, but `\sameword` can't get it.
 - * When reading the auxiliary file, we get the value associated to the pair composed by the current marked word and the current absolute line number. To this value, we subtract the value associated to the pair composed by the current marked word and the previous absolute line number. Let the result be n .
- If $n > 1$, that means the current word appears more than once in its line. In this case, we call `\showwordrank` with the word as the first argument and r as the second argument. If the word is called only once, we just print it.

After theory, implementation.

`\get@sw@txt` As the argument of `\sameword` can contain an active character if we use `inputenc` with `utf8` option instead of native UTF-8 engine, we store its detokenized content in a macro in order to allow the dynamic name of macro with `\csname`.²⁷

Because there is a bug with `\detokenize` and \LaTeX when using non BMP characters²⁸, we detokenize only for non- \LaTeX engines. In any case, in \LaTeX a `\csname` construction can contain UTF-8 characters without a problem, as UTF-8 characters are not managed with category codes, but instead read directly as UTF-8 characters.

```

1689 \newcommand{\get@sw@txt}[1]{%
1690   \ifxetex%
1691     \xdef\sw@txt{#1}%
1692   \else%
1693     \expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize{#1}}%
1694   \fi%
1695 }%
1696 %

```

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

```

1697 \newcommand{\sameword}[2][1,usedefault]{%
1698   \leavevmode%
1699   \get@sw@txt{#2}%
1700 %

```

²⁷See <http://tex.stackexchange.com/q/244538/7712>.

²⁸<http://sourceforge.net/p/xetex/bugs/108/>

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

```
1701 \unless\ifledRcol%
1702 \csnumgdef{sw@sw@txt}{\csuse{sw@sw@txt}+\@ne}%
1703 %
```

Then, write its value to the numbered file.

```
1704 \protected@write\linenum@out{}{\string\@sw{sw@txt}{\csuse{sw@sw@txt-
1705 }}{#1}}%
1706 %
```

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

```
1706 \else%
1707 \csnumgdef{sw@sw@txt}{\csuse{sw@sw@txt}+\@ne}%
1708 \protected@write\linenum@outR{}{\string\@sw{sw@txt}{\csuse{sw@sw@txt-
1709 }}{#1}}%
1710 \fi%
1711 %
```

And print the word.

```
1711 #2%
1712 }%
1713 %
```

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

```
\if@addsw14 \newif\if@addsw%
1715 %
```

\@sw The command printed in the auxiliary files.

```
1716 \newcommand{\@sw}[3]{%
1717 \get@sw@txt{#1}%
1718 \unless\ifledRcol%
1719 %
```

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

```
1720 \csxdef{sw@sw@txt @\the\absline@num @\the\section@num}{#2}%
1721 %
```

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

```
1722 \numdef{\prev@line}{\the\absline@num-1}%
1723 \ifcsundef{sw@sw@txt @\prev@line @\the\section@num}{%
1724 \csnumgdef{sw@sw@txt @\prev@line @\the\section@num}{#2-1}%
1725 }{}%
1726 %
```

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

```
1727 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1728 %
```

And do the same thing for the right side.

```
1729 \else%
1730 \csxdef{sw@sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1731 \numdef{\prev@line}{\the\absline@numR-1}%
1732 \ifcsundef{sw@sw@txt @\prev@line @\the\section@numR @R}{%
1733 \csnumgdef{sw@sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1734 }{}%
1735 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R
}}%
1736 \fi%
1737 %
```

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

```
1738 \@tempcnta=\@edtext@level
1739 \@whilenum{\@tempcnta>0}\do{%
1740 \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1741 {%
1742 \@addswfalse%
1743 \notbool{lemmacommand@\the\@tempcnta}%
1744 {\@addswtrue}%
1745 {\IfStrEq{#3}{inlemma}%
1746 {\@addswtrue}%
1747 {%
1748 \def\do##1{%
1749 \ifnumequal{##1}{\the\@tempcnta}%
1750 {\@addswtrue\listbreak}%
1751 }%
1752 }%
1753 \docsvlist{#3}%
1754 }%
1755 }%
1756 \if@addsw%
1757 \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1758 \ifledRcol%
1759 \xright@appenditem{\the@sw}{\the\absline@numR}\to\@tmp%
1760 \else%
1761 \xright@appenditem{\the@sw}{\the\absline@num}\to\@tmp%
1762 \fi%
1763 \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
1764 \fi%
1765 }%
1766 }%
1767 \advance\@tempcnta by -1%
1768 }%
1769 }
```



```
1770 %
```

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

```
1771 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1772   \get@sw@txt{#2}%
1773   \unless\ifledRcol@%
1774 %
```

Just a precaution.

```
1775   \ifx\sw@list@inedtext\empty%
1776     \def\the@sw{999}%
1777     \def\this@absline{-99}%
1778   \else%
1779 %
```

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

```
1780   \gl@p\sw@list@inedtext\to\@tmp%
1781   \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1782   \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1783   \fi%
1784 %
```

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

```
1785   \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
1786     \numdef{\prev@line}{\this@absline-1}%
1787     \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1788   }%
1789   {\numdef{\sw@atthisline}{0}}%
1790 %
```

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

```
1791   \ifnumgreater{\sw@atthisline}{1}%
1792     {\showwordrank{#2}{\the@sw}}%
1793     {#2}%
1794 %
```

And the same for right side.

```
1795   \else%
1796     \ifx\sw@list@inedtext\empty%
1797       \def\the@sw{999}%
1798       \def\this@absline{-99}%
1799     \else%
1800       \gl@p\sw@list@inedtext\to\@tmp%
1801       \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1802       \edef\this@absline{\expandafter\@secondoftwo\@tmp}%

```

```

1803 \fi%
1804 \ifcsdef{sw@sw@txt @\this@absline @\the\section@numR @R}{%
1805 \numdef{\prev@line}{\this@absline-1}%
1806 \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
1807 }%
1808 {\numdef{\sw@atthisline}{0}}%
1809 \ifnumgreater{\sw@atthisline}{1}%
1810 {\showwordrank{#2}{\the@sw}}%
1811 {#2}%
1812 \fi%
1813 }%
1814 %

```

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

```

1815 \newcommand{\showwordrank}[2]{%
1816 #1\textsuperscript{#2}%
1817 }%
1818 %

```

VII Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than straight onto the vertical list, and when the paragraph ends we slice the paragraph into its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

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

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

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

```

1819 \newbox\raw@text
1820 \newif\ifnumberedpar@
1821 \newcount\num@lines
1822 \newbox\one@line
1823 \newcount\par@line
1824 %

```

`\pstart` \pstart starts the paragraph by clearing the `\inserts@list` list and other relevant variables, and then arranges for the subsequent text to go into the `\raw@text` box.
`\AtEveryPstart` \pstart needs to appear at the start of every paragraph that is to be numbered; the
`\AtStartEveryPstart` \autopar command below may be used to insert these commands automatically.
`\numberpstarttrue` Beware: everything that occurs between `\pstart` and `\pend` is happening within
`\numberpstartfalse` a group; definitions must be global if you want them to survive past the end of the
`\labelpstarttrue` paragraph.
`\labelpstartfalse`
`\thepstart`

```

1825 \if@every@pstart@star@
1826 \newcommand{\AtStartEveryPstart}[1]{%
1827   \ifstrempy{#1}%
1828   {\gdef\@at@start@every@pstart{}}%
1829   {\gdef\@at@start@every@pstart{#1}}%
1830 }%
1831 \def\@at@start@every@pstart{}%
1832
1833 \newif\if@every@pstart@star@%
1834 \newcommand{\AtEveryPstart}[1]{%
1835   \ifstrempy{#1}%
1836   {\gdef\@at@every@pstart{}}%
1837   {\gdef\@at@every@pstart{\noindent#1}}%
1838   \global\@at@every@pstart@star@false%
1839 }%
1840 \WithSuffix\newcommand\AtEveryPstart*[1]{%
1841   \ifstrempy{#1}%
1842   {\gdef\@at@every@pstart{}}%
1843   {\gdef\@at@every@pstart{#1}}%
1844   \global\@at@every@pstart@star@true%
1845 }%
1846 \def\@at@every@pstart{}%
1847
1848 \newcounter{pstart}
1849 \renewcommand{\thepstart}{\bfseries\@arabic@c@pstart}. }
1850 \newif\ifnumberpstart
1851 \numberpstartfalse
1852 \newif\iflabelpstart
1853 \labelpstartfalse
1854 \newcommandx*\pstart[2][1,2,usedefault]{%
1855   \normal@pars%
1856   \ifboolexpr{%
1857     test {\ifstrempy{#1}}%
1858     and test {\ifstrempy{#2}}%
1859   }%
1860   {\@at@every@pstart}%
1861   {%
1862     \ifstrempy{#1}{\noindent#1}%
1863     \ifstrempy{#2}{#2}%
1864   }%
1865   \ifautopar%
1866     \autopar%

```

```

1867 \fi%
1868 \ifluatex%
1869   \edef\l@luatexttextdir@L{\the\textdir}%
1870 \fi%
1871 \@nobraektrue%
1872 \ifnumbering \else%
1873   \led@err@PstartNotNumbered%
1874   \beginnumbering%
1875 \fi%
1876 \ifnumberedpar@%
1877   \led@err@PstartInPstart%
1878   \pend%
1879 \fi%
1880 \list@clear{\inserts@list}%
1881 \global\let\next@insert=\empty%
1882 \begingroup\normal@pars%
1883 \global\advance \l@dnumstartL\@ne
1884 \global\setbox\raw@text=\vbox\bgroup%
1885   \if@nobraek%
1886     \if@afterindent\else%
1887       \noindent%
1888       \global\@afterindenttrue%
1889     \fi%
1890   \fi%
1891   \ifautopar\else%
1892   \ifnumberpstart%
1893     \ifinstanza\else%
1894       \ifsidepstartnum\else%
1895         \thepstart%
1896       \fi%
1897     \fi%
1898   \fi%
1899   \fi%
1900 \numberedpar@true%
1901 \iflabelpstart\protected@edef\@currentlabel%
1902   {\p@pstart\thepstart}
1903 \fi%
1904 \l@dzeropenalties%
1905 \@at@start@every@pstart%
1906 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
1907 }
1908 %

```

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

```

1909 \newcommandx*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
1910   \led@err@PendNotNumbered%
1911   \fi%
1912   \global\l@dskipversenumberfalse%

```

```

1913 \ifnumberedpar@ \else%
1914 \led@err@PendNoPstart%
1915 \fi%
1916 %

```

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.

```

1917 \l@dzeropenalties%
1918 \@at@end@every@pend%
1919 \endgraf\global\num@lines=\prevgraf\egroup%
1920 \global\par@line=0%
1921 %

```

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`.

```

1922 \csnumdef{pstartline}{0}%
1923 \loop\ifvbox\raw@text%
1924 \csnumdef{pstartline}{\pstartline+\@ne}%
1925 \do@line%
1926 \ifbypstart@%
1927 \ifnumequal{pstartline}{1}{%
1928 \bgroup%
1929 \let\leavevmode\relax%
1930 \setline{1}%
1931 \egroup%
1932 \resetprevline@}{}%
1933 \fi%
1934 \repeat%
1935 %

```

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

```

1936 \flush@notes%
1937 \endgroup%
1938 \ignorespaces%
1939 %

```

Increase `pstart` counter.

```

1940 \ifnumberpstart%
1941 \global\pstartnumtrue%
1942 \fi%
1943 \addtocounter{pstart}{1}%
1944 %

```

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

```

1945 \normal@pars%
1946 \ifboolexpr{%
1947   test {\ifstrempy{#1}}%
1948   and test {\ifstrempy{#2}}%
1949 }%
1950 {\at@every@pend}%
1951 {%
1952   \ifstrempy{#1}{\noindent#1}%
1953   \ifstrempy{#2}{#2}%
1954 }%
1955 %

```

Restore standard “nobreak” and “autopar” settings. Normally, `\if@nobreak` is true only immediately after a sectioning command (see `latex.ltx` file). As a `\pstart... \pend` structure can’t contain any sectioning command, we set `\if@nobreak` to false.

```

1956 \@nobreakfalse%
1957 \ifautopar%
1958   \autopar%
1959 \fi%
1960 }
1961 %

```

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.

```

\AtEveryPend62
\at@every@pend63
\ifat@every@pend@star@64 \newif\ifat@every@pend@star@%
1965 \newcommand{\AtEveryPend}[1]{%
1966   \ifstrempy{#1}%
1967   {\gdef\at@every@pend{}}%
1968   {\gdef\at@every@pend{\noindent#1}}%
1969   \global\at@every@pend@star@false%
1970 }%
1971 \WithSuffix\newcommand\AtEveryPend*[1]{%
1972   \ifstrempy{#1}%
1973   {\gdef\at@every@pend{}}%
1974   {\gdef\at@every@pend{#1}}%
1975   \global\at@every@pend@star@true%
1976 }%
1977 \xdef\at@every@pend{}%
1978
1979 %

```

`\AtEndEveryPend` Here a macro to insert automatically any content at the end of `\pend`, in numbered lines.

```

1980 \newcommand{\AtEndEveryPend}[1]{%
1981   \ifstrempy{#1}%
1982     {\xdef\@at@end@every@pend{}}%
1983     {\gdef\@at@end@every@pend{#1}}%
1984 }%
1985 \def\@at@end@every@pend{}%
1986 %

```

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

```

1987 \newcommand*\l@dzero penalties{%
1988   \brokenpenalty \z@ \clubpenalty \z@
1989   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
1990   \postdisplaypenalty \z@ \widowpenalty \z@}
1991 %
1992 %

```

\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.

```

1993 \newif\ifautopar
1994 \autoparfalse
1995 \newcommand*\autopar{%
1996   \ifledRcol
1997     \ifnumberingR \else
1998       \led@err@AutoparNotNumbered
1999       \beginnumberingR

```

```

2000 \fi
2001 \else
2002 \ifnumbering \else
2003 \led@err@AutoparNotNumbered
2004 \beginnumbering
2005 \fi
2006 \fi
2007 \autopartrue
2008 \everypar{\setbox0=\lastbox
2009 \endgraf \vskip-\parskip
2010 \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\
fi\fi
2011 \let\par=\pend}%
2012 \ignorespaces}
2013 %

```

\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.

```

2014 \newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}
2015
2016 %

```

\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.

```

2017 \newif\ifautopar@pause
2018 %

```

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

```

2019 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
2020 \newcommand*{\do@line}{%
2021 {\vbadness=10000
2022 \splittopskip=\z@
2023 \do@linehook
2024 \l@demptyd@ta
2025 \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2026 \unvbox\one@line \global\setbox\one@line=\lastbox
2027 \getline@num
2028 \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{}}
2029 \ifnum\@lock>\@ne
2030 \inserthangingsymboltrue

```



```

2031 \else
2032     \inserthangingsymbolfalse
2033 \fi
2034 \check@pb@in@verse
2035 \ifl@dhidenumber%
2036     \global\l@dhidenumberfalse%
2037     \f@x@l@cks%
2038 \else%
2039     \affixline@num%
2040 \fi%
2041 %

```

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

```

2042 \xifinlist{\the\l@dnumpstartsL}{\eled@sections@@}%
2043     {\print@eledsection}%
2044     {\print@line}%
2045 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{-}
2046 }%
2047 %

```

VII.2.2 Process for “normal” line

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

```

2048 \def\print@line{
2049 %

```

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

```

2050     \affixpstart@num%
2051 %

```

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

```

2052     \hb@xt@ \linewidth{%
2053 %

```

User hook.

```

2054         \do@insidelinehook%
2055 %

```

Left line number

```

2056         \l@dld@ta%
2057 %

```

Prepare text to be inserted before notes.

```

2058         \if@firstlineofpage%
2059             \set@Xtxtbeforenotes%
2060             \global\@firstlineofpagefalse%
2061         \fi%
2062 %

```

Insert footnotes made of manuscripts data and critical footnotes.

```

2063 \ifdefstring{\ms@data@position}{msdata-regular}{%
2064 \insert@msdata%
2065 \add@inserts%
2066 \add@Xgroupbyline%
2067 }{%
2068 \add@inserts%
2069 \add@Xgroupbyline%
2070 \insert@msdata%
2071 }%
2072 %

```

Insert marginal notes.

```

2073 \affixside@note%
2074 %

```

Print left notes.

```

2075 \l@dlsn@te
2076 %

```

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

```

2077 {\ledllfill\hb@xt@ \wd\one@line{\new@line%
2078 %

```

If we use Lua_{TeX} then restore the direction.

```

2079 \ifluatex%
2080 \textdir\l@luatextextdir@L%
2081 \fi%
2082 %

```

Insert, if needed, the hanging symbol.

```

2083 \inserthangingsymbol%
2084 %

```

And so, print the line.

```

2085 \l@dunhbox@line{\one@line}}%
2086 %

```

Right line number

```

2087 \ledrlfill\l@drd@ta%
2088 %

```

Print right notes.

```

2089 \l@drsn@te
2090 }}%
2091 %

```

And reinsert penalties (for page breaking)...

```

2092 \add@penalties%
2093 }
2094 %

```

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.

```

2095 \def\print@eledsection{%
2096   \if@firstlineofpage%
2097     \set@Xtxtbeforenotes%
2098     \global\@firstlineofpagefalse%
2099   \fi%
2100   \ifdefstring{\ms@data@position}{msdata-regular}{}%
2101     \insert@msdata%
2102     \add@inserts%
2103     \add@Xgroupbyline%
2104   }{%
2105     \add@inserts%
2106     \add@Xgroupbyline%
2107     \insert@msdata%
2108   }%
2109   \affixside@note%
2110   \numdef{\temp@}{\l@dnumpstartsL-1}%
2111   \xifinlist{\temp@}{\eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
2112   \@eled@sectioningtrue%
2113   \csuse{eled@sectioning@the\l@dnumpstartsL}%
2114   \@eled@sectioningfalse%
2115   \global\csundef{eled@sectioning@the\l@dnumpstartsL}%
2116   \if@RTL%
2117     \hspace{-3\paperwidth}%
2118     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2119   \else%
2120     \hspace{3\paperwidth}%
2121     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2122   \fi%
2123   \vskip-\baselineskip%
2124 }
2125 %

```

VII.2.4 Hooks

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

\do@insidelinehook

```

2126 \newcommand*{\do@linehook}{}
2127 \newcommand*{\do@insidelinehook}{}
2128 %

```

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

```

2129 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
2130 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
2131
2132 %

```

VII.2.5 Sidenotes and marginal line number initialization

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

```

\l@dcsnotetext
2133 \newcommand*{\l@emptyd@ta}{%
\l@dcsnotetext@l
2134 \gdef\l@dld@ta{}%
\l@dcsnotetext@r
2135 \gdef\l@drd@ta{}%
2136 \gdef\l@dcsnotetext@l{}%
2137 \gdef\l@dcsnotetext@r{}%
2138 \gdef\l@dcsnotetext{}%
2139
2140 %

```

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

```

\l@drsn@te
2141 \newcommand{\l@dlsn@te}{%
2142 \hb@xt@ \z@{\hss\box\l@dldp@rbox\kern\ledlsnotesep}}
2143 \newcommand{\l@drsn@te}{%
2144 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
2145
2146 %

```

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

```

2147 \newcommand*{\ledllfill}{\hfil}
2148 \newcommand*{\ledrlfill}{}
2149
2150 %

```

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.

```

2151 \newcommand*{\getline@num}{%
2152   \global\advance\absline@num \@ne%
2153   \do@actions
2154   \do@ballast
2155   \ifnumberline
2156     \ifsublines@
2157       \ifnum\sub@lock<\tw@
2158         \global\advance\subline@num \@ne
2159       \fi
2160     \else
2161       \ifnum\@lock<\tw@
2162         \global\advance\line@num \@ne
2163         \global\subline@num \z@
2164       \fi
2165     \fi
2166   \fi
2167 }
2168 %

```

\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. 159).

\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.

```

2169 \newcount\ballast@count
2170 \newcounter{ballast}
2171 \setcounter{ballast}{0}
2172 %

```

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.

```

2173 \newcommand*{\do@ballast}{\global\ballast@count \z@
2174   \begingroup
2175     \advance\absline@num \@ne
2176     \ifnum\next@actionline=\absline@num
2177       \ifnum\next@action>-1001\relax
2178         \global\advance\ballast@count by -\c@ballast
2179       \fi
2180     \fi
2181   \endgroup}
2182 %

```

`\do@actions` The `\do@actions` macro looks at the list of actions to take at particular absolute line numbers, and does everything that is specified for the current line.

It may call itself recursively, and to do this efficiently (using \TeX 's optimization for tail recursion), we define a control-sequence called `\do@actions@next` that is always the last thing that `\do@actions` does. If there could be more actions to process for this line, `\do@actions@next` is set equal to `\do@actions`; otherwise it is just `\relax`.

```
2183 \newcommand*{\do@actions}{%
2184   \global\let\do@actions@next=\relax
2185   \ifnum\absline@num<\next@actionline\else
2186   %
```

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.

```
2187   \ifnum\next@action>-1001
2188     \global\page@num=\next@action
2189     \global\@firstlineofpagetrue%
2190     \ifbypage@
2191       \global\line@num=\z@ \global\subline@num=\z@
2192       \resetprevline@
2193     \fi
2194     \add@msdata@firstlineofpage%
2195   %
```

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`.)

```
2196   \else
2197     \ifnum\next@action<-4999
2198       \@l@tempcnta=-\next@action
2199       \advance\@l@tempcnta by -5001
2200       \ifsublines@
2201         \global\subline@num=\@l@tempcnta
2202       \else
2203         \global\line@num=\@l@tempcnta
2204       \fi
2205   %
```

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

```
2206   \else
2207     \@l@tempcnta=-\next@action
2208     \advance\@l@tempcnta by -1000
2209     \do@actions@fixedcode
2210   \fi
2211 \fi
2212 %
```

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.

```

2213 \ifx\actionlines@list\empty
2214 \gdef\next@actionline{1000000}%
2215 \else
2216 \gl@p\actionlines@list\to\next@actionline
2217 \gl@p\actions@list\to\next@action
2218 \global\let\do@actions@next=\do@actions
2219 \fi
2220 \fi
2221 %

```

Make the recursive call, if necessary.

```

2222 \do@actions@next}
2223
2224 %

```

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

```

2225 \newcommand*{\do@actions@fixedcode}{%
2226 \ifcase\@l@dttempcnta
2227 \or% % 1001 = starting sublineation
2228 \global\sublines@true
2229 \or% % 1002 = ending sublineation
2230 \global\sublines@false
2231 \or% % 1003 = starting locking number
2232 \global\@lock=\@ne
2233 \or% % 1004 = ending locking number
2234 \ifnum\@lock=\tw@
2235 \global\@lock=\thr@@
2236 \else
2237 \global\@lock=\z@
2238 \fi
2239 \or% % 1005 = starting locking subnumber
2240 \global\sub@lock=\@ne
2241 \or% % 1006 = ending locking subnumber
2242 \ifnum\sub@lock=\tw@
2243 \global\sub@lock=\thr@@
2244 \else
2245 \global\sub@lock=\z@
2246 \fi
2247 \or% % 1007 = skipping numbering
2248 \l@dskipnumbertrue
2249 \or% % 1008 = skipping numbering in stanza
2250 \l@dskipversenumbertrue%
2251 \or% % 1009 = hiding number
2252 \l@dhidenumbertrue
2253 \or% % 1010 = inserting msdata
2254 \add@msdata%

```

```

2255 \else
2256   \led@warn@BadAction
2257 \fi}
2258
2259
2260 %

```

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.

```

2261 \newcommand*{\affixline@num}{\%
2262 %

```

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).

```

2263 \ifledgroupnotesL\else
2264   \ifnumberline
2265     \ifl@dskipnumber
2266       \global\l@dskipnumberfalse
2267     \else
2268       \ifsublines@
2269         \@l@tempcntb=\subline@num
2270         \ifnum\subline@num>\c@firstsublinenum
2271           \@l@tempcnta=\subline@num
2272           \advance\@l@tempcnta by-\c@firstsublinenum
2273           \divide\@l@tempcnta by\c@sublinenumincrement
2274           \multiply\@l@tempcnta by\c@sublinenumincrement
2275           \advance\@l@tempcnta by\c@firstsublinenum
2276         \else
2277           \@l@tempcnta=\c@firstsublinenum
2278         \fi
2279 %

```


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.

```
2280      \ch@cksub@l@ck
2281 %
```

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

```
2282      \else
2283      \@l@tempcntb=\line@num
2284 %
```

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

```
2285      \ifx\linenumberlist\empty
2286      \ifnum\line@num>\c@firstlinenum
2287      \@l@tempcnta=\line@num
2288      \advance\@l@tempcnta by-\c@firstlinenum
2289      \divide\@l@tempcnta by\c@linenumincrement
2290      \multiply\@l@tempcnta by\c@linenumincrement
2291      \advance\@l@tempcnta by\c@firstlinenum
2292      \else
2293      \@l@tempcnta=\c@firstlinenum
2294      \fi
2295      \else
2296 %
```

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

```
2297      \@l@tempcnta=\line@num
2298      \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2299      \edef\sc@n@list{\def\noexpand\sc@n@list
2300      ###1,\number\@l@tempcnta,###2|\def\noexpand\rem@inder
2301      {####2}}}%
2302      \sc@n@list\expandafter\sc@n@list\rem@inder|
2303      \ifx\rem@inder\empty%
2304      \advance\@l@tempcnta\@ne
2305      \fi
2306 %
```

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

```
2307      \ch@ck@l@ck
2308      \fi
2309 %
```

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

```
2310      \ifnum\@l@tempcnta=\@l@tempcntb
2311      \ifl@dskipversenumber\else
2312 %
```

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`.

```

2313 \l@drd@ta \if@twocolumn
2314         \if@firstcolumn
2315             \gdef\l@dld@ta{\llap{\leftlinenum}}%
2316         \else
2317             \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2318         \fi
2319     \else
2320         \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\line@margin@columns}{\m@ne}}}%
2321             {\@l@tempcntb=\line@margin@columns}%
2322             {\@l@tempcntb=\line@margin}%
2323         \ifnum\@l@tempcntb>\@ne
2324             \advance\@l@tempcntb \page@num
2325         \fi
2326         \ifodd\@l@tempcntb
2327             \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2328         \else
2329             \gdef\l@dld@ta{\llap{\leftlinenum}}%
2330         \fi
2331     \fi
2332 \fi
2333 \fi
2334 %

```

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

```

2335     \f@x@l@cks
2336 \fi
2337 \fi
2338 \fi
2339 }
2340
2341 %

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck`
`\ch@ck@l@ck`
`\f@x@l@cks`

checks subline locking. If it fails, then we disable the line-number display by setting the counters to arbitrary but unequal values.

```

2342 \newcommand*{\ch@cksub@l@ck}{%
2343   \ifcase\sub@lock
2344   \or
2345     \ifnum\sublock@disp=\@ne
2346       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2347     \fi
2348   \or
2349     \ifnum\sublock@disp=\tw@ \else
2350       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2351     \fi
2352   \or
2353     \ifnum\sublock@disp=\z@
2354       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2355     \fi
2356   \fi}
2357 %

```

Similarly for line numbers.

```

2358 \newcommand*{\ch@ck@l@ck}{%
2359   \ifcase\@lock
2360   \or
2361     \ifnum\lock@disp=\@ne
2362       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2363     \fi
2364   \or
2365     \ifnum\lock@disp=\tw@ \else
2366       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2367     \fi
2368   \or
2369     \ifnum\lock@disp=\z@
2370       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2371     \fi
2372   \fi}
2373 %

```

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

```

2374 \newcommand*{\f@x@l@cks}{%
2375   \ifcase\@lock
2376   \or
2377     \global\@lock=\tw@
2378   \or \or
2379     \global\@lock=\z@
2380   \fi
2381   \ifcase\sub@lock
2382   \or
2383     \global\sub@lock=\tw@

```

```

2384 \or \or
2385 \global\sub@lock=\z@
2386 \fi}
2387
2388 %

```

X Pstart number printing in side

In side, the printing of pstart number is running like the printing of line number. There is only some differences:

- The pstarts counter is upgrade in the \pend command. Consequently, the \affixpstart@num command has not to upgrade it, unlike the \affixline@num which upgrades the lines counter.
- To print the pstart number only at the beginning of a pstart, and not in every line, a boolean test is made. The \pstartnum boolean is set to TRUE at every \pend. It is tried in the \leftpstartnum and \rightpstartnum commands. After the try, it is set to FALSE.

```

\leftpstartnum89
\rightpstartnum90 \newif\ifsidepstartnum
\ifsidepstartnum91 \newcommand*{\affixpstart@num}{%
2392 \ifsidepstartnum
2393 \if@twocolumn
2394 \if@firstcolumn
2395 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2396 \else
2397 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2398 \fi
2399 \else
2400 \l@dttempcntb=\line@margin%
2401 \ifnum\l@dttempcntb>\@ne
2402 \advance\l@dttempcntb \page@num
2403 \fi
2404 \ifodd\l@dttempcntb
2405 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2406 \else
2407 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2408 \fi
2409 \fi
2410 \fi
2411 }
2412 %
2413
2414
2415 \newif\ifpstartnum
2416 \pstartnumtrue

```

```

2417 \newcommand*{\leftpstartnum}{
2418     \ifpstartnum\thepstart
2419     \kern\linenumsep\fi
2420     \global\pstartnumfalse
2421 }
2422 \newcommand*{\rightpstartnum}{
2423     \ifpstartnum
2424     \kern\linenumsep
2425     \thepstart
2426     \fi
2427     \global\pstartnumfalse
2428 }
2429 %

```

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.

```

2430 \list@create{\inserts@list}
2431 %

```

`\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`.

```

2432 \newcommand*{\add@inserts}{%
2433     \global\let\add@inserts@next=\relax
2434 %

```

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

```

2435     \ifx\inserts@list\empty \else
2436 %

```

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.

```

2437 \ifx\next@insert\empty
2438 \ifx\insertlines@list\empty
2439 \global\noteschanged@true
2440 \gdef\next@insert{100000}%
2441 \else
2442 \gl@p\insertlines@list\to\next@insert
2443 \fi
2444 \fi
2445 %

```

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 ourselves recursively: there might be another insert for this same line.

```

2446 \ifnum\next@insert=\absline@num
2447 \gl@p\inserts@list\to\@insert
2448 \@insert
2449 \global\let\@insert=\undefined
2450 \global\let\next@insert=\empty
2451 \global\let\add@inserts@next=\add@inserts
2452 \fi
2453 \fi
2454 %

```

Make the recursive call, if necessary.

```

2455 \add@inserts@next}
2456
2457 %

```

`\add@Xgroupbyline` If you use `\Xgroupbyline`, the insertion of the critical footnotes are not made immediately in `\add@inserts`, but the content to be inserted is stored, to be inserted in one block. This insertion in one block is made by `\add@Xgroupbyline`.

```

2458 \newcommand{\add@Xgroupbyline}{%
2459 \unless\ifnocritical{%
2460 \def\do##1{%Looping on the series
2461 \def\do####1{%Looping on the ##1@forinserting command
2462 \ifcsdef{##1@forinserting@####1}{%
2463 \X@beforeinsertion{##1}%
2464 \if@ledgroup%
2465 \global\setbox\@nameuse{mp##1footins}=\vbox%
2466 \else%
2467 \insert\csname ##1footins\endcsname%
2468 \fi%
2469 }%
2470 \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
2471 {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%

```

```

2472     {}%
2473     \if@ledgroup%
2474         \unvbox\@nameuse{mp##1footins}%
2475     \fi%
2476     \X@atbegininsertion{##1}%
2477     \ifcsstring{series@display##1}%
2478         {%
2479             \Xledsetnormalparstuff{##1}%
2480             \rule\z@\splittopskip%
2481         }%
2482     {}%
2483     \csuse{##1@forinserting@####1}%
2484     \strut\par%
2485 }%
2486 \global\csundef{##1@forinserting@####1}%
2487 }%
2488 {}%
2489 }%
2490 \ifcsdef{##1@forinserting}{%
2491     \dolistcslloop{##1@forinserting}%
2492 }{%
2493     \global\csundef{##1@forinserting}%
2494 }%
2495 \dolistloop{\@series}%
2496 \fi%
2497 }%
2498
2499
2500 %

```

XI.2 Penalties

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

In this code, `\num@lines` is the number of lines in the whole paragraph, and `\par@line` is the line we are working on at the moment. The count `\@l@dttempcnta` is used to calculate and accumulate the penalty; it is initially set to the value of `\ballast@count`, which has been worked out in `\do@ballast` above (VIII p. 149). Finally, the penalty is checked to see that it does not go below -10000 .

```

2501 \newcommand*{\add@penalties}{\@l@dttempcnta=\ballast@count
2502     \ifnum\num@lines>\@ne
2503         \global\advance\par@line \@ne
2504         \ifnum\par@line=\@ne
2505             \advance\@l@dttempcnta \clubpenalty
2506         \fi

```

```

2507 \l@tempcntb=\par@line \advance\l@tempcntb \@ne
2508 \ifnum\l@tempcntb=\num@lines
2509   \advance\l@tempcnta \widowpenalty
2510 \fi
2511 \ifnum\par@line<\num@lines
2512   \advance\l@tempcnta \interlinepenalty
2513 \fi
2514 \fi
2515 \ifnum\l@tempcnta=\z@
2516   \relax
2517 \else
2518   \ifnum\l@tempcnta>-10000
2519     \penalty\l@tempcnta
2520   \else
2521     \penalty -10000
2522   \fi
2523 \fi}
2524
2525 %

```

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.

```

2526 \newcommand*{\flush@notes}{%
2527   \@xloop
2528   \ifx\inserts@list\empty \else
2529     \glp\inserts@list\to\@insert
2530     \@insert
2531     \global\let\@insert=\undefined
2532   \repeat}
2533
2534 %

```

\@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.

```
2535 \def\@xloop#1\repeat{%
2536   \def\body{#1\expandafter\body\fi}%
2537   \body}
2538
2539 %
```

XI.4 Text before notes

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

```
2540 \newcommand{\set@Xtxtbeforenotes}{%
2541   \unless\ifnocritical{%
2542     \def\do##1{%
2543       \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2544     }%
2545     \dolistloop{\@series}%
2546   \fi%
2547 }%
2548 %
```

`\insert@Xtxtbeforenotes` `\insert@Xtxtbeforenotes{⟨series⟩}`, called when inserting a note, will insert the text before the note if it is not already inserted. For paragraphed footnotes, it will insert it as a component of the first footnote. For other types of footnotes, it will insert it as a regular footnote.

```
2549 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2550   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2551     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2552     \ifcsvoid{Xtxtbeforenotes@#1}{%
2553       \ifcsstring{series@display#1}{paragraph}%
2554       {\noindent\csuse{Xtxtbeforenotes@#1}}%
2555       {\expandafter\insert\csname#1footins\endcsname%
2556         \bgroup%
2557           \noindent%
2558           \csuse{\csuse{series@display#1}@begin@insert}{#1}%
2559           \strut\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}%
2560         \egroup%
2561       }%
2562     }%
2563   }%
2564 }%
2565 %
```

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.

```

2567 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@@lemmafont#7|}
2568 \def\select@@lemmafont#1/#2/#3/#4|{%
2569     {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2570     \selectfont}
2571
2572 %

```

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.

```

2573 \newcommand*\footnoteoptions@[3]{%
2574     \def\do##1{%
2575         \ifstrequal{#1}{L}{% In Leftside
2576             \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
2577 unexpanded{##1}}}{\to\inserts@list%
2578             \global\advance\insert@count \@ne% Increment the left insert
2579 counter.
2580             }%
2581             \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
2582 unexpanded{##1}}}{\to\inserts@listR%
2583             \global\advance\insert@countR \@ne% Increment the right insert
2584 counter insert.
2585             }%
2586             \notblank{#2}{\docsvlist{#2}}{}}% Parsing all options
2587 }
2588 %

```

XII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua¹TeX is used.

```

2587 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
2588   \ifstrequal{#1}{L}{%
2589     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
inserts@list%Know the dir of lemma
2590     \global\advance\insert@count \@ne%
2591     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@list%Know the dir of lemma
2592     \global\advance\insert@count \@ne%
2593   }%
2594   {%
2595     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
inserts@listR%Know the dir of lemma
2596     \global\advance\insert@countR \@ne%
2597     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@listR%Know the dir of lemma
2598     \global\advance\insert@countR \@ne%
2599   }%
2600 }
2601 %

```

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

```

2602 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2603   \ifstrequal{#1}{L}{%
2604     \if@RTL%
2605       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@list%Know the language used in the lemma
2606       \global\advance\insert@count \@ne%
2607     }%
2608     \else
2609       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@list%Know the language of lemma
2610       \global\advance\insert@count \@ne%
2611     }%
2612     \fi%
2613     \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}{\to\
to\inserts@list%Know the language of lemma
2614     \global\advance\insert@count \@ne%
2615   }%
2616   {%
2617     \if@RTL
2618       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@listR%Know the language of lemma
2619       \global\advance\insert@countR \@ne%
2620     }%
2621     \else
2622       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@listR%Know the language of lemma

```

```

2620 \global\advance\insert@countR \@ne%
2621 \fi
2622 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language\language}}}{\
to\inserts@listR%Know the language of lemma
2623 \global\advance\insert@countR \@ne%
2624 }%
2625 }
2626 %

```

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

`\footplitskips` Some setup code that is common for a variety of the footnotes. The setup is for:

- `\interlinepenalty`.
- `\splittopskip` (skip before last part of notes that flow from one page to another).
- `\splitmaxdepth`.
- `\floatingpenalty`, that is penalty values being added when a long note flows from one page to another. Here, we let it to 0 when we are processing parallel pages in `eledpar`, in order to allow notes to flow from left to right pages and *vice-versa*. Otherwise, we let it to `\@MM`, which is the standard \TeX `\floatingpenalty`.

```

2627 \newcommand*{\footplitskips}{%
2628 \interlinepenalty=\interfootnotelinepenalty
2629 \unless\ifl@dprintingpages%
2630 \floatingpenalty=\@MM%
2631 \fi%

```

```

2632 \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2633 \leftskip=\z@skip \rightskip=\z@skip
2634
2635 %

```

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

```

2636 \let\normalfootnoterule=\footnoterule
2637 %

```

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.

```

2638 \newcommand{\Xarrangement}[2][1,usedefault]{%
2639   \def\do##1{%
2640     \csname Xarrangement@#2\endcsname{##1}%
2641   }%
2642   \ifstrempy{#1}%
2643     {%
2644       \dolistloop{\@series}%
2645     }%
2646     {
2647       \docsvlist{#1}%
2648     }%
2649 }%
2650 %

```

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\vsizel
\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.

```

2651 \newcommand*{\Xarrangement@normal}[1]{%
2652   \csgdef{series@display#1}{normal}
2653   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
2654   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
2655   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
2656   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
2657   \expandafter\let\csname #1footnoterule\endcsname=%
2658                               \normalfootnoterule
2659   \count\csname #1footins\endcsname=1000
2660   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2661   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2662   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2663   %

```

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.

```

2664   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
2665   side only
2666   %

```

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).

```

2666   \ifnoledgroup@else%
2667     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2668     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2669     \count\csname mp#1footins\endcsname=1000
2670     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2671     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2672     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2673   \fi
2674 }
2675 %
2676 %

```

`\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.

```

2677 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2678   \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
2679   \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
2680 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
2681   \X@beforeinsertion{#1}%
2682   \insert\csname #1footins\endcsname{%
2683     \X@atbegininsertion{#1}%
2684     \normalvfootnote@inserted{#1}{#2}%
2685   }%
2686 }%
2687 }%
2688 %

```

\normalvfootnote@inserted The `\normalvfootnote@inserted` macro is expanded to the content to be add to a `\insert` for normal critical footnote.

```

2689 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote@inserted}[2]{%
2690   \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
2691   \csuse{Xnotefontsize@#1}%
2692   \footssplitskip
2693   \ifl@dpairing\ifl@dpaging\else%
2694     \setXnoteswidthliketwocolumns@{#1}%
2695   \fi\fi%
2696   \setXnotespositionliketwocolumns@{#1}%
2697   \spaceskip=\z@skip \xspaceskip=\z@skip%
2698   \csname #1footfmt\endcsname #2{#1}%
2699 }%
2700 %

```

```

\X@beforeinsertion01 \newcommand{\X@beforeinsertion}[1]{%
2702   \if@ledgroup\else%
2703     \insert@Xtxtbeforenotes{#1}%
2704   \fi%
2705   \csuse{Xbeforeinserting@#1}%
2706 }%
2707 %

```

```

\X@atbegininsertion08 \newcommand{\X@atbegininsertion}[1]{%
2709   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2710 }%
2711 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnnormalvfootnote12 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnnormalvfootnote}[2]{%
2713   \iftoggle{Xgroupbyline@#1}{%
2714     \prepare@Xgroupbyline{#1}{#2}{\mpnnormalvfootnote@inserted}%
2715   }%
2716   {%
2717     \global\setbox\@nameuse{mp#1footins}%
2718     \vbox{%
2719       \unvbox\@nameuse{mp#1footins}%
2720       \mpnnormalvfootnote@inserted{#1}{#2}%
2721     }%
2722   }%
2723 }%
2724
2725 %

```

```

\mpnnormalvfootnote@inserted26 \newcommand{\mpnnormalvfootnote@inserted}[2]{%
2727   \noindent\csuse{Xbhooknote@#1}%
2728   \csuse{Xnotefontsize@#1}%
2729   \hsize\columnwidth%
2730   \@parboxrestore%
2731   \color@begingroup%
2732   \csname #1footfmt\endcsname #2{#1}\color@endgroup%
2733 }%
2734 %

```

\normalfootfmt `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 98), 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.

```

2735 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
2736   \Xstorelineinfo{#1}{#4}%
2737   \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}{}%
2738   \hangindent=\csuse{Xhangindent@#4}%
2739   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2740   \nottoggle{Xgroupbyline@#4}{\rule\z@\splittopskip}{}%
2741   {\printlinefootnote{#1}{#4}}%
2742   \print@lemma{#1}{#2}{#4}%
2743   \csuse{Xwrapcontent@#4}{#3}%
2744   \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
2745 }%
2746 %

```

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

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

But if the skip `\Xprenotes@` is greater than 0 pt, it is used instead of `\skip\footins` for the first printed series in one page.

The `\leftskip` and `\rightskip` values are both zeroed here. Similarly, these skips are cancelled in the `\vfootnote` macros for the various types of notes. Strictly speaking, this is necessary only if you are using paragraphed footnotes, but we have put it here and in the other `\vfootnote` macros too so that the behavior of `reledmac` in this respect is general across all footnote types. What this means is that any `\leftskip` and `\rightskip` you specify applies to the main text, but not the footnotes. The footnotes continue to be of width `\hsize`.

```
2747 \newcommand*{\normalfootstart}[1]{%
2748 %
```

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

```
2749 \ifdimequal{0pt}{\Xprenotes@}{}%
2750 {%
2751 \iftoggle{Xprenotes@}{%
2752 \togglefalse{Xprenotes@}%
2753 \skip\csname #1footins\endcsname=%
2754 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterterrule@#1}\relax%
2755 }%
2756 }%
2757 }%
2758 \vskip\skip\csname #1footins\endcsname%
2759 %
```

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. 217 for the general description of the problem.

```
2760 \leftskip0pt \rightskip0pt
2761 \ifl@dpairing\else%
2762 \hsize=\old@hsize%
2763 \fi%
2764 \setXnoteswidthliketwocolumns@{#1}%
2765 \setXnotespositionliketwocolumns@{#1}%
2766 %
```

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

```
2767 \print@Xfootnoterule{#1}%
2768 }%
2769 %
```

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

```

2770 \newcommand*{\normalfootgroup}[1]{%
2771   \csuse{Xhookgroup@#1}%
2772   \unvbox\csname #1footins\endcsname%
2773   \hsize=\old@hsize%
2774   }%
2775
2776 %

```

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

```

2777 \unless\ifnoledgroup@
2778 \newcommand*{\mpnormalfootgroup}[1]{%
2779   \vskip\skip\@nameuse{mp#1footins}
2780   \ifl@dpairing\ifparledgroup%
2781     \leavevmode\marks\parledgroup@{begin}%
2782     \marks\parledgroup@series{#1}%
2783     \marks\parledgroup@type{Xfootnote}%
2784     \fi\fi\normalcolor%
2785     \ifparledgroup%
2786       \ifl@dpairing%
2787       \else%
2788         \setXnoteswidthliketwocolumns@{#1}%
2789         \setXnotespositionliketwocolumns@{#1}%
2790         \print@Xfootnoterule{#1}%
2791       \fi%
2792     \else%
2793       \setXnoteswidthliketwocolumns@{#1}%
2794       \setXnotespositionliketwocolumns@{#1}%
2795       \print@Xfootnoterule{#1}%
2796     \fi%
2797     \setlength{\parindent}{0pt}
2798     \csuse{Xhookgroup@#1}%
2799     \unvbox\csname mp#1footins\endcsname}}
2800 \fi
2801 %

```

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.

```

2802 \newcommand*{\Xarrangement@paragraph}[1]{%
2803   \csgdef{series@display#1}{paragraph}
2804   \expandafter\let\csname #1footstart\endcsname=\parafootstart
2805   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
2806   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
2807   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
2808   \count\csname #1footins\endcsname=1000
2809   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2810   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2811   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2812   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2813   \para@footsetup{#1}
2814   %

```

And the extra setup for minipages.

```

2815 \ifnoledgroup@else
2816   \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
2817   \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
2818   \count\csname mp#1footins\endcsname=1000
2819   \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2820   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2821   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2822   \fi
2823 }
2824 %

```

`\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.

```

2825 \providecommand{\footfudgefiddle}{64}
2826 %

```

`\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`.

```

2827 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}

```

```

2828 \setXnoteswidthliketwocolumns@{#1}%
2829 \ifcsempy{Xwidth@#1}%
2830   {}%
2831   {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
2832   \dimen0=\baselineskip
2833   \multiply\dimen0 by 1024
2834   \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
2835   \csxdef{#1footfudgefactor}{%
2836     \expandafter\strip@pt\dimen0 }}
2837
2838 %

```

\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.

```

2839 \newcommand*{\parafootstart}[1]{%
2840   \rightskip=0pt \leftskip=0pt%
2841   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2842   \ifdimequal{0pt}{\Xprenotes@}{}%
2843   {%
2844     \iftoggle{Xprenotes@}{%
2845       \togglefalse{Xprenotes@}%
2846       \skip\csname #1footins\endcsname=%
2847       \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
2848     }%
2849   }%
2850 }%
2851 \vskip\skip\csname #1footins\endcsname%
2852 \setXnoteswidthliketwocolumns@{#1}%
2853 \setXnotespositionliketwocolumns@{#1}%
2854 \print@Xfootnoterule{#1}%
2855 \let\bidir@RTL@everypar\@empty%
2856 \noindent\leavevmode}
2857 %

```

\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.²⁹

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.³⁰ 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.³¹ Michael's unboxing macro is called `\Xunvzx`: `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.³² 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. 169 above). We need to do this, since `\footfudgefactor` is calculated on the assumption that the notes are `\hsize` wide.

So, finally, here is the modified foot-paragraph code, which sets the footnote in vertical mode so that language and discretionary nodes are included.

```
2858 \newcommand*{\paravfootnote}[2]{%
2859   \csuse{Xbeforeinserting@#1}%
2860   \insert\csname #1footins\endcsname
2861   \bgroup
```

²⁹Michael Downes, 'Line Breaking in `\unhboxed` Text', *TUGboat* **11** (1990), pp. 605–612.

³⁰See *The TeXbook*, p. 455 (editions after January 1990).

³¹Wayne supplied his own macros to do this, but since they were almost identical to Michael's, Peter Wilson have used the latter's `\Xunvzx` macro since it is publicly documented.

³²'Line Breaking', p. 610.

```

2862 \csuse{Xnotefontsize@#1}
2863 \footsplitskips
2864 \setbox0=\vbox{\hsize=\maxdimen%
2865   \let\bidir@RTL@everypar\@empty%
2866   \insert@Xtxtbeforenotes{#1}%
2867   \noindent\csuse{Xhooknote@#1}%
2868   \csname #1footfmt\endcsname #2{#1}}%
2869 \setbox0=\hbox{\Xunvxh{0}{#1}}%
2870 \dp0=0pt
2871 \ht0=\csname #1footfudgefactor\endcsname\wd0
2872 %

```

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

```

2873 \if@RTL\noindent \leavevmode\fi\box0%
2874 \penalty0
2875 \egroup}
2876
2877 %

```

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.

```

2878 \newcommand*{\mpparavfootnote}[2]{%
2879   \global\setbox\@nameuse{mp#1footins}\vbox{%
2880     \unvbox\@nameuse{mp#1footins}%
2881     \csuse{Xnotefontsize@#1}
2882     \footsplitskips
2883     \setbox0=\vbox{\hsize=\maxdimen%
2884       \let\bidir@RTL@everypar\@empty%
2885       \insert@Xtxtbeforenotes{#1}%
2886       \noindent\color@begingroup%
2887       \csuse{Xhooknote@#1}%
2888       \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
2889     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2890     \dp0=\z@
2891     \ht0=\csname #1footfudgefactor\endcsname\wd0
2892     \box0
2893     \penalty0
2894   }}

```

2895
2896 %

\Xunvvh Here is (modified) Michael's definition of `\unvvh`, 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). `\unvvh` cancels these unwanted paragraph-final items using `\unskip` and `\unpenalty`.

```
2897 \newcommand*{\Xunvvh}[2]{%
2898   \setbox0=\vbox{\unvbox#1%
2899     \global\setbox1=\lastbox}%
2900   \unhbox1
2901   \unskip           % remove \rightskip,
2902   \unskip           % remove \parfillskip,
2903   \unpenalty        % remove \penalty of 10000,
2904   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
2905 %
2906 %
```

\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).

```
2907 \newcommand*{\parafootfmt}[4]{%
2908   \Xstorelineinfo{#1}{#4}%
2909   \Xinsertparafootsep{#4}%
2910   \ledsetnormalparstuff@common%
2911   \printlinefootnote{#1}{#4}%
2912   \print@lemma{#1}{#2}{#4}%
2913   \csuse{Xwrapcontent@#4}{#3}%
2914   \penalty-10 }
2915 %
```

Note that in the above definition, the penalty of -10 encourages a line break between notes, so that notes have a slight tendency to begin on new lines. The `\Xinsertparafootsep` command is used to insert the `\Xparafootsep@series` between each note in the *same* page.

\parafootgroup This footgroup code is modelled on the macros in *The TeXbook*, p. 399. The only difference is the `\unpenalty` in `\makehboxofhboxes`, which is there to remove the penalty of 0 which was added to the end of each footnote by `\paravfootnote`.

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

```
2916 \newcommand*{\parafootgroup}[1]{%
2917   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
```

```

2918 \unvbox\csname #1footins\endcsname
2919 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2920 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2921 \makehboxofhboxes
2922 \setbox0=\hbox{\unhbox0 \removehboxes}%
2923 \csuse{Xbhookgroup@#1}%
2924 \csuse{Xnotefontsize@#1}%
2925 \unhbox0\par%
2926 \global\hsize=\old@hsize%
2927 }%
2928
2929 %

```

`\mpparafootgroup` The minipage version.

```

2930 \newcommand*\mpparafootgroup}[1]{%
2931 \setXnoteswidthliketwocolumns@{#1}%
2932 \vskip\skip\@nameuse{mp#1footins}
2933 \ifl@dpairing\ifparledgroup%
2934 \leavevmode\marks\parledgroup@{begin}%
2935 \marks\parledgroup@series{#1}%
2936 \marks\parledgroup@type{Xfootnote}%
2937 \fi\fi\normalcolor
2938 \ifparledgroup%
2939 \ifl@dpairing%
2940 \else%
2941 \setXnoteswidthliketwocolumns@{#1}%
2942 \setXnotespositionliketwocolumns@{#1}%
2943 \print@Xfootnoterule{#1}%
2944 \fi%
2945 \else%
2946 \setXnoteswidthliketwocolumns@{#1}%
2947 \setXnotespositionliketwocolumns@{#1}%
2948 \print@Xfootnoterule{#1}%
2949 \fi%
2950 \unvbox\csname mp#1footins\endcsname
2951 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2952 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2953 \makehboxofhboxes
2954 \setbox0=\hbox{\unhbox0 \removehboxes}%
2955 \csuse{Xbhookgroup@#1}%
2956 \csuse{Xnotefontsize@#1}%
2957 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2958 \unhbox0\par}}
2959
2960 %

```

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


```

\makehboxofhboxes61 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes62 \loop
2963 \unpenalty
2964 \setbox2=\lastbox
2965 \ifhbox2
2966 \setbox0=\hbox{\box2\unhbox0}%
2967 \repeat}
2968
2969 \newcommand*{\removehboxes}{\setbox0=\lastbox
2970 \ifhbox0{\removehboxes}\unhbox0 \fi}
2971
2972 %

```

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

```

\prevpage@num73 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep74 \iflhdRcol{%
2975 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
2976 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2977 {\ifcsequal{prevline#1}{lineinfo@}%
2978 {\ifcseempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
2979 {\csuse{Xparafootsep@#1}}}%
2980 }%
2981 {\csuse{Xparafootsep@#1}}%
2982 }%
2983 }%
2984 \global\csname #1prevpage@numR\endcsname=\page@numR%
2985 \else%
2986 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
2987 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2988 {\ifcsequal{prevline#1}{lineinfo@}%
2989 {\ifcseempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
2990 {\csuse{Xparafootsep@#1}}}%
2991 }%
2992 {\csuse{Xparafootsep@#1}}%
2993 }%
2994 }%
2995 \global\csname #1prevpage@num\endcsname=\page@num%
2996 \fi%
2997 }
2998 %

```

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

`\dosplits`

`\splitoff`

`\@h`

`\@k`

columns, each with a space (#3) between the top baseline and the top of the \vbox. The \rigidbalance macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they do not depend on white space. Note also the extra unboxing in \splittoff, which allows the new \vbox to have its natural height as it goes into the alignment.

The L^AT_EX \line macro has no relationship to the TeX \line. The L^AT_EX 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.

```

2999 \newcount\@k \newdimen\@h
3000 \newcommand*\Xrigidbalance}[3]{%
3001   \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
3002   \rigidbalance{#1}{#2}{#3}%
3003 }%
3004
3005 \newcommand*\rigidbalanceX}[3]{%
3006   \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
3007   \rigidbalance{#1}{#2}{#3}%
3008 }%
3009
3010 \newcommand*\rigidbalance}[3]{%
3011   \setbox0=\box#1 \@k=#2 \@h=#3%
3012   @@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3013     \valign{##\vfil\cr\dosplits}}}%
3014
3015 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splittoff
3016   \global\advance\@k-1\cr\dosplits\fi}
3017
3018 \newcommand*\splittoff{\dimen0=\ht0
3019   \divide\dimen0 by\@k \advance\dimen0 by\@h
3020   \setbox2 \vsplit0 to \dimen0
3021   \unvbox2 }
3022
3023 %

```

Three columns

```

\Xarrangement@threecol 24 \newcommand*\Xarrangement@threecol}[1]{%
3025   \csgdef{series@display#1}{threecol}
3026   \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3027   \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3028   \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
3029   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3030   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3031   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3032   \threecolfootsetup{#1}
3033 %

```

The additional setup for minipages.

```

3034 \ifnoledgroup@else
3035   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3036   \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3037   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3038   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3039   \mpthreecolfootsetup{#1}
3040 \fi
3041 }
3042
3043 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XII.6.2 p. 168 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.

```

3044 \newcommand*{\threecolfootsetup}[1]{%
3045   \count\csname #1footins\endcsname 333
3046   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3047   \multiply\dimen\csname #1footins\endcsname \thr@@}
3048 %

```

`\mpthreecolfootsetup` The setup for minipages.

```

3049 \newcommand*{\mpthreecolfootsetup}[1]{%
3050   \count\csname mp#1footins\endcsname 333
3051   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3052
3053 %

```

`\threecolvfootnote` `\threecolvfootnote` This is the `\vfootnote` command for three-column notes. However, most of the code is deported on `\threecolvfootnote@inserted`. The call to `\Xnotefontsize@<s>` ensures that the `\splittopskip` and `\splitmaxdepth` take their values from the right `\strutbox`: the one used in a footnotes. Note especially

the importance of temporarily reducing the `\hsize` to 0.3 of its normal value. This determines the widths of the individual columns. So if the normal `\hsize` is (say) 10 cm, then each column will be $0.3 \times 10 = 3$ cm wide, leaving a gap of 1 cm spread equally between columns (i.e., .5 cm between each).

The arguments are #1 the note series letter and #1 the full text of the note (including numbers, lemma and text).

```

3054 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnote}[2]{%
3055   \iftoggle{Xgroupbyline@#1}{%
3056     \prepare@Xgroupbyline{#1}{#2}{\threecolvfootnote@inserted}%
3057   }%
3058   {%
3059     \X@beforeinsertion{#1}%
3060     \insert\csname #1footins\endcsname{%
3061       \threecolvfootnote@inserted{#1}{#2}%
3062     }%
3063   }%
3064 }%
3065 %

```

```

\threecolvfootnote@inserted \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
threecolvfootnote@inserted}[2]{%
3067   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3068   \noindent\csuse{Xhooknote@#1}%
3069   \csuse{Xnotefontsize@#1}%
3070   \footplitskips%
3071   \csname #1footfmt\endcsname #2{#1}%
3072 }%
3073 %

```

`\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.

```

3074 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
3075   \threecol@begin@insert{#4}%
3076   \hspace{\parindent}%
3077   \printlinefootnote{#1}{#4}%
3078   \print@lemma{#1}{#2}{#4}%
3079   \csuse{Xwrapcontent@#4}{#3}%
3080   \nottoggle{Xgroupbyline@#4}%
3081   {\strut\par\allowbreak}%
3082   {}%
3083 }%
3084 %

```

`\threecol@begin@insert` The `\threecol@begin@insert` contains code used at the beginning of any `\insert` for critical footnotes in three columns. It is used both by `\threecolfootfmt` and by `\insert@Xtxtbeforenotes`.

```

3085 \newcommand{\threecol@begin@insert}[1]{%
3086   \normal@pars%
3087   \nottoggle{Xgroupbyline@#1}%
3088   {\hspace \csuse{Xhsizethreecol@#1}}%
3089   }%
3090   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3091   \tolerance=5000%
3092   \hangindent=\csuse{Xhangindent@#1}%
3093   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3094   \@tempdima=\parindent%
3095   \csuse{Xcolalign@#1}%
3096   \parindent=\@tempdima%
3097   \strut%
3098 }%
3099 %

```

\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.

```

3100 \newcommand*\threecolfootgroup}[1]{%
3101   \csuse{Xhookgroup@#1}\par%
3102   \splittopskip=\ht\strutbox
3103   \expandafter
3104   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
3105 %

```

\mpthreecolfootgroup The setup for minipages.

```

3106 \newcommand*\mpthreecolfootgroup}[1]{%
3107   \vskip\skip\@nameuse{mp#1footins}
3108   \ifl@dpairing\ifparledgroup%
3109     \leavevmode\marks\parledgroup@{begin}%
3110     \marks\parledgroup@series{#1}%
3111     \marks\parledgroup@type{Xfootnote}%
3112   \fi\fi\normalcolor
3113   \ifparledgroup%
3114     \ifl@dpairing%
3115     \else%
3116       \setXnoteswidthliketwocolumns@{#1}%
3117       \setXnotespositionliketwocolumns@{#1}%
3118       \print@Xfootnoterule{#1}%
3119     \fi%

```

```

3120 \else%
3121 \setXnoteswidthliketwocolumns@{#1}%
3122 \setXnotespositionliketwocolumns@{#1}%
3123 \printXfootnoterule{#1}%
3124 \fi%
3125 \csuse{Xbhookgroup@#1}\par%
3126 \splittopskip=\ht\strutbox
3127 \expandafter
3128 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3129
3130 %

```

Two columns

```

\Xarrangement@twocol 31 \newcommand*\Xarrangement@twocol}[1]{%
3132 \csgdef{series@display#1}{twocol}
3133 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3134 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3135 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3136 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3137 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3138 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3139 \twocolfootsetup{#1}
3140 %

```

The additional setup for minipages.

```

3141 \ifnoledgroup\else
3142 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3143 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
3144 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3145 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3146 \mptwocolfootsetup{#1}
3147 \fi
3148 }
3149
3150 %

```

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

```

\twocolvfootnote
\twocolfootnote@inserted
\twocolfootfmt
\twocolfootgroup
3151 \newcommand*\twocolfootsetup}[1]{%
3152 \count\csname #1footins\endcsname 500
3153 \csxdef{default@#1footins}{500}%
3154 \multiply\dimen\csname #1footins\endcsname \tw@}
3155 %

```

```

3156 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote}[2]{%
3157   \iftoggle{Xgroupbyline@#1}{%
3158     \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3159   }{%
3160     \X@beforeinsertion{#1}%
3161     \insert\csname #1footins\endcsname{%
3162       \twocolvfootnote@inserted{#1}{#2}%
3163     }%
3164   }%
3165 }%
3166 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
twocolvfootnote@inserted}[2]{%
3167   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3168   \noindent\csuse{Xhooknote@#1}%
3169   \csuse{Xnotefontsize@#1}%
3170   \footsplitskips%
3171   \csname #1footfmt\endcsname #2{#1}%
3172 }%
3173 %

3174 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmt}[4]{% 4th
arg is optional, for backward compatibility
3175   \twocol@begin@insert{#4}%
3176   \hspace{\parindent}%
3177   \printlinefootnote{#1}{#4}%
3178   \print@lemma{#1}{#2}{#4}%
3179   \csuse{Xwrapcontent@#4}{#3}%
3180   \nottoggle{Xgroupbyline@#4}%
3181   {\strut\par\allowbreak}%
3182   {}%
3183 }%
3184 %

3185 \newcommand{\twocol@begin@insert}[1]{%
3186   \normal@pars%
3187   \hsize \csuse{Xhsizetwocol@#1}%
3188   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3189   \tolerance=5000%
3190   \hangindent=\csuse{Xhangindent@#1}%
3191   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3192   \@tempdima=\parindent%
3193   \csuse{Xcolalign@#1}%
3194   \parindent=\@tempdima%
3195   \strut%
3196 }%
3197
3198 \newcommand*{\twocolfootgroup}[1]{%
3199   \csuse{Xhookgroup@#1}\par%
3200   \splittopskip=\ht\strutbox
3201   \expandafter

```

```

3202 \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
3203
3204 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3205 \newcommand*\mptwocolfootsetup}[1]{%
3206   \count\csname mp#1footins\endcsname 500
3207   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3208 %

3209 \newcommand*\mptwocolfootgroup}[1]{%
3210   \vskip\skip\@nameuse{mp#1footins}
3211   \ifl@dpairing\ifparledgroup%
3212     \leavevmode\marks\parledgroup@{begin}%
3213     \marks\parledgroup@series{#1}%
3214     \marks\parledgroup@type{Xfootnote}%
3215   \fi\fi\normalcolor
3216   \ifparledgroup%
3217     \ifl@dpairing%
3218     \else%
3219       \setXnoteswidthliketwocolumns@{#1}%
3220       \setXnotespositionliketwocolumns@{#1}%
3221       \print@Xfootnoterule{#1}%
3222     \fi%
3223   \else%
3224     \setXnoteswidthliketwocolumns@{#1}%
3225     \setXnotespositionliketwocolumns@{#1}%
3226     \print@Xfootnoterule{#1}%
3227   \fi%
3228   \csuse{Xbhookgroup@#1}\par%
3229   \splittopskip=\ht\strutbox
3230   \expandafter
3231   \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3232
3233 %

```

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.

Note that these commands are not directly called by `reledmac`, but are enclosed as default value of specific hooks. Consequently, people should not redefine them, but use instead the `\Xlinerangeseparator`, `\Xendlinrangeseparator`, `\Xsublinesep`, `\Xendsublinesep` and `\Xlemmaseparator` macros.

With `polyglossia`, each critical note has a `\footnote@lang` which shows the language of the lemma, and which can be used to switch the bracket from right to left.

```

3234 \def\endashchar{\textnormal{--}}
3235
3236 \newcommand*{\fullstop}{\textnormal{.}}
3237 \def\Xsublinesep@side{\fullstop}
3238
3239 \newcommand*{\rbracket}{\textnormal{%
3240   \csuse{text}\csuse{footnote@lang}}{%
3241     \ifluatex%
3242       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]}{\thinspace
3243     }}%
3244     \else%
3245       \thinspace}%
3246     \fi}%
3247 }
3248
3249 %

```

XII.7.2 Pstart number in footnote

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

```

3250 \newcommand{\printpstart}[0]{%
3251   \ifbool{expr}{bool{1@dpairing} or bool{1@dprintingpages} or bool{
3252     1@dprintingcolumns}}{%
3253     \ifledRcol%
3254       \thepstartR%
3255     \else%
3256       \thepstartL%
3257     \fi%
3258   }{%
3259     \thepstart%
3260   }%
3261 }

```

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).

```

3262 %
3263 \newcommand{\print@lemma}[3]{%
3264   \bgroup%
3265   \nottoggle{Xlemmadisablefontselection@#3}%
3266   {\select@lemmafонт#1|}%
3267   }%
3268   \bgroup%
3269   \csuse{Xlemmafонт@#3}%Deprecated
3270   \csuse{Xwraplemma@#3}{#2}%
3271   \egroup%
3272   \egroup%
3273   \iftoggle{nosep@}{%
3274     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3275     \relax%
3276   }%
3277   {\ifcsemtty{Xlemmaseparator@#3}%
3278     {%
3279       \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3280       \relax%
3281     }%
3282     {%
3283       \nobreak%
3284       \hskip\csuse{Xbeforelemmaseparator@#3}%
3285       \csuse{Xlemmaseparator@#3}%
3286       \hskip\csuse{Xafterlemmaseparator@#3}%
3287       \relax%
3288     }%
3289   }%
3290 }%
3291 %

```

XII.7.4 Line number printing

`\Xstorelineinfo` The `\Xstorelineinfo` macro is used to store some data about line number of the current critical footnote, data which will be reused later for the `\Xnumberonlyfirstinline` and related setting.

#1 footnote specification for the current footnote ; #2 footnote series.

```

3292 \newcommand{\Xstorelineinfo}[2]{%
3293   \l@dp@rsefootspec#1|}%
3294   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3295     \xdef\lineinfo{\l@dparsedstartline - \l@dparsedstartsub - \
3296     \l@dparsedendline - \l@dparsedendsub}%
3297   }%
3298   {%

```

```

3298 \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3299 }%
3300 }%
3301 %

```

\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`.

```

3302 \newcommand{\printlinefootnote}[2]{%
3303   \iftoggle{nonum@}{%Try if the line number must printed for this specific
3304     not (by default, yes)
3305     \hspace{\csuse{Xinplaceofnumber@#2}}%
3306     }%
3307     {%
3308       \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
3309       default, yes)
3310       {%
3311         \hspace{\csuse{Xinplaceofnumber@#2}}%
3312         }%
3313         {%\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
3314         line number must be printed only in the first time.
3315         {%
3316           \ifcsdef{prevline#2}%
3317             {%Be sure the \prevline exists.
3318             \ifcsequal{prevline#2}{\lineinfo@}%Try it
3319             {%
3320               \ifcseempty{Xsymlinenum@#2}%
3321               {%
3322                 \hspace{\csuse{Xinplaceofnumber@#2}}%
3323                 }%
3324                 {\printsymlinefootnotearea{#2}}%
3325                 }%
3326                 {%
3327                   \printlinefootnotearea{#1}{#2}%
3328                   }%
3329                 }%
3330                 \printlinefootnotearea{#1}{#2}%
3331                 }%
3332               }%
3333               {%
3334                 \printlinefootnotearea{#1}{#2}%
3335                 }%
3336                 \csxdef{prevline#2}{\lineinfo@}%
3337               }%
3338             }%

```

```

3339 }%
3340 }%
3341 }
3342 %

```

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

```

3343 \newcommand{\printsymlinefootnotearea}[1]{%
3344   \hspace{\csuse{Xbeforesymlinenum@#1}}%
3345   \csuse{Xnotenumfont@#1}%
3346   \ifdimequal{\csuse{Xboxsymlinenum@#1}}{\z@}%
3347     {\csuse{Xsymlinenum@#1}}%
3348     {\hbox to \csuse{Xboxsymlinenum@#1}%
3349       {\csuse{Xsymlinenum@#1}\hfill}%
3350     }%
3351   \hspace{\csuse{Xaftersymlinenum@#1}}%
3352 }%
3353 %

```

\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.)

```

3354 \newcommand{\printlinefootnotearea}[2]{%
3355   \printXbeforenumber{#2}%
3356   \csuse{Xnotenumfont@#2}%
3357   \boxfootnotenumbers{#1}{#2}%
3358   \printXafternumber{#2}%
3359 }%
3360 %

```

\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.

```

3361 \newcommand{\boxfootnotenumbers}[2]{%
3362   \ifdimequal{\csuse{Xboxlinenum@#2}}{\Opt}{%
3363     \printlinefootnotenumbers{#1}{#2}%
3364   }%
3365   {%
3366     \hbox to \csuse{Xboxlinenum@#2}%
3367     {%
3368       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3369       \printlinefootnotenumbers{#1}{#2}%
3370       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3371     }%
3372   }%
3373 }%
3374 %

```

`\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.

```

3375 \newcommand{\printlinefootnotenumbers}[2]{%
3376   \xdef\@currentseries{#2}%
3377   \ifboolexpr{%
3378     (togl{Xpstart@#2} and bool{numberpstart})%
3379     or togl{Xpstarteverytime@#2}}%
3380   {\printpstart}{}%
3381   \iftoggle{Xstanza@#2}{%
3382     \ifnumberstanza%
3383       \printstanza%
3384       \csuse{Xstanzaseparator@#2}%
3385     \fi%
3386   }{}%
3387   \iftoggle{Xonlypstart@#2}{}{%
3388     \csuse{Xtxtbeforenumber@#2}%
3389     \printlines#1|\ifledRcol@{\@Rlineflag\fi}}%
3390 }%
3391 %

```

`\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.)

```

3392 \newcommand{\printXbeforenumber}[1]{%
3393   \hspace{\csuse{Xbeforenumber@#1}}%
3394 }%
3395 %

```

`\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

```

3396 \newcommand{\printXafternumber}[1]{%
3397   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3398   \hspace{\csuse{Xafternumber@#1}}%
3399 }%
3400 %

```

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. 98: 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@xtwoline` and `\ifl@d@xmorethantwoline` to print a symbol which stands for “and subsequent” when there are two, three or more lines.

```

\ifl@d@pnum01 \newif\ifl@d@pnum
\ifl@d@ssub02 \newif\ifl@d@ssub
\ifl@d@elin03 \newif\ifl@d@elin
\ifl@d@esl04 \newif\ifl@d@esl
\ifl@d@dash05 \newif\ifl@d@dash
\ifl@d@xtwoline06 \newif\ifl@d@xtwoline%
\ifl@d@xmorethantwoline07 \newif\ifl@d@xmorethantwoline%
08 %

```

```

\l@dp@rsefootsspec \l@dp@rsefootsspec{<spec>}{<lemma>}{<text>} parses a footnote specification. <lemma>
\l@dp@rsefootsspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@dp@rsefootsspec ber and lemma font specifier in \l@d@nums style format. The real work is done by
\l@dp@rsefootsspec \l@dp@rsefootsspec which defines macros holding the numeric values. In many cases,
\l@dp@rsefootsspec this last command is called directly. Just a reminder of the arguments:
\l@dp@rsefootsspec \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dp@rsefootsspec \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@dp@rsefootsspec
3409 \newcommand*{\l@dp@rsefootsspec}[3]{\l@dp@rsefootsspec#1|}
3410 \def\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
3411 \gdef\l@dp@rsefootsspec#1}%
3412 \gdef\l@dp@rsefootsspec#2}%
3413 \gdef\l@dp@rsefootsspec#3}%
3414 \gdef\l@dp@rsefootsspec#4}%
3415 \gdef\l@dp@rsefootsspec#5}%
3416 \gdef\l@dp@rsefootsspec#6}%
3417 }
3418 %

```

Initialise the several number value macros.

```

3419 \def\l@dp@rsefootsspec{0}%
3420 \def\l@dp@rsefootsspec{0}%
3421 \def\l@dp@rsefootsspec{0}%
3422 \def\l@dp@rsefootsspec{0}%
3423 \def\l@dp@rsefootsspec{0}%
3424 \def\l@dp@rsefootsspec{0}%
3425
3426 %

```

`\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`.

```
3427 \newcommand*\setprintlines}[6]{%
3428   \l@dpnumfalse \l@ddashfalse
3429   %
```

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

```
3430   \ifbypage@
3431     \ifnum#4=#1 \else
3432       \l@dpnumtrue
3433       \l@ddashtrue
3434     \fi
3435   \fi
3436   %
```

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.

```
3437   \ifl@dpnum \l@d@elintrue \else \l@d@elinfalse \fi
3438   \ifnum#2=#5 \else
3439     \l@d@elintrue
3440     \l@ddashtrue
3441   \fi
3442   %
```

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

```
3443   \l@d@ssubfalse
3444   \ifnum#3=0 \else
3445     \l@d@ssubtrue
3446   \fi
3447   %
```

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.

```
3448   \l@d@eslfalse
3449   \ifnum#6=0 \else
3450     \ifnum#6=#3
3451       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3452     \else
3453       \l@d@esltrue
3454       \l@ddashtrue
3455     \fi
3456   \fi%
3457   %
```

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

```

3458 \ifl@dash%
3459 \ifbool{expr{togl{fulllines@} or test{\ifcsempy{Xtwolines@
@currentseries}}}}%
3460 {}%
3461 {%
3462 \setistwofollowinglines{#1}{#2}{#4}{#5}%
3463 \ifbool{expr{%
3464 (%
3465 togl {Xtwolinesbutnotmore@}\@currentseries}%
3466 and not%
3467 (%
3468 bool {istwofollowinglines@}%
3469 )%
3470 }%
3471 or%
3472 (%
3473 (not test{\ifnumequal{#1}{#4}})%
3474 and togl{Xtwolinesonlyinsamepage@}\@currentseries}%
3475 )%
3476 }%
3477 {}%
3478 {%
3479 \l@d@dashfalse%
3480 \l@d@Xtwolinestrue%
3481 \l@d@elinfalse%
3482 \l@d@eslfalse%
3483 \ifcsempy{Xmorethantwolines@}\@currentseries}%
3484 {}%
3485 {\ifistwofollowinglines@}\else%
3486 \l@d@Xmorethantwolinestrue%
3487 \fi%
3488 }%
3489 }%
3490 }%
3491 \fi%
3492 %

```

End of \setprintlines.

```

3493 }%
3494 %

```

\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.

```

3495 \newif\ifistwofollowinglines@%
3496 \newcommand{\setistwofollowinglines}[4]{%
3497   \ifcsdef{lastlinenumberon@#1}%
3498     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
3499     {\numdef{\tmp}{0}}}%
3500   \istwofollowinglines@false%
3501   \ifnumequal{#4-#2}{1}%
3502     {\istwofollowinglines@true}%
3503     {\ifbypage@%
3504       \ifnumequal{#3-#1}{1}%
3505       {%
3506         \ifnumequal{#2}{\tmp}%
3507         {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
3508         {}%
3509       }%
3510     }%
3511   \fi%
3512 }%
3513 }%
3514 %

```

`\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

```

3515 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3516   \begingroup%
3517   %

```

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

```

3518   \ifluatex%
3519     \edef\@tmp{\the\textdir}%
3520     \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
    spurious space (bug #397)
3521   \fi%
3522   %

```

Decide which part of line number components we will print.

```

3523 \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3524 %

```

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.

```

3525 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3526 {\bgroup}%
3527 {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
3528 \ifl@d@pnum%
3529 \wrap@edcrossref{\@this@crossref@start}{#1}%
3530 \csuse{Xpagelinesep@\@currentseries}%
3531 \fi%
3532 \wrap@edcrossref{\@this@crossref@start}{%
3533 \linenumrep{#2}%
3534 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3535 }%
3536 \ifl@d@ssub%
3537 \csuse{Xsublinesep@\@currentseries}%
3538 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3539 \fi
3540 \egroup%
3541 %

```

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

```

3542 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3543 {\bgroup}%
3544 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
3545 \ifl@d@Xtwolines%
3546 \ifl@d@Xmorethantwolines%
3547 \csuse{Xmorethantwolines@\@currentseries}%
3548 \else%
3549 \csuse{Xtwolines@\@currentseries}%
3550 \fi%
3551 \else%
3552 \ifl@d@dash%
3553 \ifdefined\linerangesep%
3554 \linerangesep%
3555 \else%
3556 \csuse{Xlinerangeseparator@\@currentseries}%
3557 \fi%
3558 \fi%
3559 \ifl@d@pnum%
3560 \wrap@edcrossref{\@this@crossref@end}{#4}%
3561 \csuse{Xpagelinesep@\@currentseries}%
3562 \fi%
3563 \ifl@d@elin%

```

```

3564 \wrap@edcrossref{\@this@crossref@end}{%
3565 \linenumrep{#5}%
3566 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3567 }%
3568 \fi%
3569 \ifl@d@esl%
3570 \ifl@d@elin%
3571 \csuse{Xsublinesep@\@currentseries}%
3572 \fi%
3573 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3574 \fi%
3575 \fi%
3576 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3577 {}%
3578 {\hfill}%Prevent underfull hbox
3579 \egroup%
3580 \endgroup%
3581 }%
3582 %

```

XII.7.5 Footnote grouped by line

`\prepare@Xgroupbyline` `\prepare@Xgroupbyline` is a macro called on on the `\<XXX>vfootnote` if `\Xgroupbyline` is set to true, instead of calling directly the `\insert`.

```

3583 \newcommand{\prepare@Xgroupbyline}[3]{%
3584 \iftoggle{Xgroupbylineseparatetwolines@#1}{%
3585 \l@dparsfootspec#2%
3586 \ifcsdef{#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3587 {%
3588 \csgappto%
3589 {#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3590 {%
3591 \ifcsemt{Xsymlinenum@#1}%
3592 {\csuse{Xparafootsep@#1}}%
3593 {}%
3594 #3{#1}{#2}%
3595 \hskip\csuse{Xafternote@#1}\relax%
3596 }%
3597 }%
3598 {%
3599 \csdef%
3600 {#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3601 {%
3602 #3{#1}{#2}%
3603 \hskip\csuse{Xafternote@#1}\relax%
3604 }%

```

```

3605 }%
3606 \listcsxadd{#1@forinserting}{\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3607 }{%
3608 \ifcsdef{#1@forinserting@all}{%
3609 \csgappto%
3610 {#1@forinserting@all}}%
3611 {%
3612 \ifcsemt{Xsymlinenum@#1}%
3613 {\csuse{Xparafootsep@#1}}%
3614 }%
3615 #3{#1}{#2}%
3616 \hskip\csuse{Xafternote@#1}\relax%
3617 }%
3618 }%
3619 {%
3620 \csdef%
3621 {#1@forinserting@all}%
3622 {%
3623 #3{#1}{#2}%
3624 \hskip\csuse{Xafternote@#1}\relax%
3625 }%
3626 }%
3627 \listcsgadd{#1@forinserting}{all}%
3628 }%
3629 }%
3630 %

```

XIII Familiar footnotes

XIII.1 Adjacent footnotes

The original edmac provided users with five series of critical footnotes (`\Afootnote`, `\Bfootnote`, `\Cfootnote`, `\Dfootnote`, `\Efootnote`), and \TeX 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`

```

3631 \providecommand*{\multiplefootnotemarker}{3sp}
3632 \providecommand*{\multfootsep}{\textsuperscript{\normalfont,}}

```

```
3633
3634 %
```

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

```
3635 \providecommand*\m@mmf@prepare{%
3636   \kern-\multiplefootnotemarker
3637   \kern\multiplefootnotemarker\relax}
3638 %
```

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

```
3639 \providecommand*\m@mmf@check{%
3640   \ifdim\lastkern=\multiplefootnotemarker\relax
3641     \edef\x@sf{\the\spacefactor}%
3642     \unkern
3643     \multfootsep
3644     \spacefactor\x@sf\relax
3645   \fi}
3646
3647 %
```

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

```
3648 \@ifclassloaded{memoir}{%
3649 %
```

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

```
3650 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
3651 %
```

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

```
3652
3653 \patchcmd{\@footnotemark}
3654   {\nobreak}
3655   {\m@mmf@check
3656    \nobreak
3657   }
3658   {}{}
3659 \patchcmd{\@footnotemark}
3660   {\@makefnmark}
3661   {\@makefnmark
3662    \m@mmf@prepare
3663   }
3664   {}{}
3665 %
```

Finished the modifications for the non-memoir case.

```
3666 }
3667
3668 %
```

XIII.2 Regular footnotes for numbered texts

`\l@doldold@footnotetext` In order to enable the regular `\footnotes` in numbered text we have to play around
`\@footnotetext` with its `\@footnotetext`, using different forms for when in numbered or regular text.

```
3669 \pretocmd{\@footnotetext}{%
3670   \ifnumberedpar@
3671   \edtext{}{\l@dbfnote{#1}}}%
3672 \else
3673   {}{}
3674 \apptocmd{\@footnotetext}{\fi}{}{}%
3675 %
```

```

\l@dbfnote \l@dbfnote adds the footnote to the insert list, and \v1@dbfnote calls the original
\v1@dbfnote \@footnotetext. We also patch \footnote in order to get the correct footnote
\v1@dbfnote numbers when typesetting parallel texts. This is moved into a \get@fnmark command.
\footnote
\get@fnmark
\get@thisfootnote
3676
3677 \patchcmd%
3678   {\footnote}%
3679   {\stepcounter\@mpfn}%
3680   {%
3681     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3682       l@dprintingcolumns}}{%
3683       \global\advance\footnote@reading by \one%
3684       \get@thisfootnote%
3685       \get@fnmark{\thisc@footnote}%
3686       \ifcsdef{footnotereading\the\footnote@reading=typeset}%
3687         {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
3688           typeset}}}%
3689         {\setcounter{\@mpfn}{\footnote@reading}}}%
3690       }{%
3691         \stepcounter\@mpfn%
3692       }%
3693     }%
3694   }%
3695   {}
3696   {}
3697 \newcommand{\get@thisfootnote}{%
  \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
    \protected@xdef\thisc@footnote{\the\footnote@reading}%
  }%
```

```

3698 }{%
3699 \protected@xdef\thisc@footnote{\the\c@footnote}%
3700 }%
3701 }%
3702
3703 \newcommand{\l@dbfnote}[1]{%
3704 \get@thisfootnote%
3705 \gdef\@tag{#1\relax}%
3706 \ifledRcol%
3707 \xright@appenditem{%
3708 \ifdefined\Hy@footnote@currentHref%
3709 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3710 \fi%
3711 \noexpand\vl@dbfnote{\expandonce\@tag}}{\thisc@footnote}%
3712 }%
3713 \to\inserts@listR
3714 \global\advance\insert@countR \@ne%
3715 \else%
3716 \xright@appenditem{%
3717 \ifdefined\Hy@footnote@currentHref%
3718 \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3719 \fi%
3720 \noexpand\vl@dbfnote{\expandonce\@tag}}{\thisc@footnote}%
3721 }%
3722 \to\inserts@list
3723 \global\advance\insert@count \@ne%
3724 \fi
3725 \ignorespaces%
3726 }%
3727
3728 \newcommand{\get@fnmark}[1]{%
3729 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}%
3730 {%
3731 \stepcounter{footnote@typeset}%
3732 \setcounter{footnote}{\c@footnote@typeset}%
3733 \immediate\write\@mainaux{%
3734 \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
3735 }%
3736 \def\@thefnmark{\thefootnote}%
3737 }%
3738 {%
3739 \setcounter{footnote}{#1}%
3740 \def\@thefnmark{\thefootnote}%
3741 }%
3742 }%
3743
3744 \newcommand{\vl@dbfnote}[2]{%

```

```

3745 \get@fnmark{#2}%
3746 \@footnotetext{#1}%
3747 }%
3748 %

```

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`

```

3749 \newcommand*\prebodyfootmark{%
3750 \leavevmode
3751 \ifhmode
3752 \edef\x@sf{\the\spacefactor}%
3753 \m@mmf@check
3754 \nobreak
3755 \fi}
3756 \newcommand*\postbodyfootmark{%
3757 \m@mmf@prepare
3758 \ifhmode\spacefactor\x@sf\fi\relax}
3759 %
3760 %

```

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.

```

3761 \newcommandx*\arrangementX[2][1,usedefault]{%
3762 \def\do##1{%
3763 \csname arrangementX@#2\endcsname{##1}%
3764 }%
3765 \ifstrepty{#1}%
3766 {%
3767 \dolistloop{\@series}%
3768 }%
3769 {
3770 \docsvlist{#1}%
3771 }%
3772 }%
3773 %

```


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.

```

3774 \newcommand*{\normal@footnotemarkX}[1]{%
3775   \prebodyfootmark
3776   \wrapped@bodyfootmarkX{#1}%
3777   \postbodyfootmark}
3778
3779 %

```

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

```

3780 \newcommand*{\normalbodyfootmarkX}[1]{%
3781   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}%
3782 %

```

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

```

3783 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalvfootnoteX}[2]{%
3784   \csuse{beforeinsertingX@#1}%
3785   \insert\@nameuse{footins#1}\bgroup
3786     \reset@font%
3787     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3788     \noindent\csuse{bhooknoteX@#1}%
3789     \csuse{notefontsizeX@#1}%
3790     \footsplitskips
3791     \ifl@dpairing\ifl@dpaging\else%
3792       \setnoteswidthliketwocolumnsX@{#1}%
3793     \fi\fi%
3794     \setnotesXpositionliketwocolumns@{#1}%
3795     \spaceskip=\z@skip \xspaceskip=\z@skip
3796     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
3797
3798 %

```

`\mpnormalvfootnoteX` The minipage version.

```

3799 \newcommand*{\mpnormalvfootnoteX}[2]{%
3800   \get@thisfootnoteX{#1}%
3801   \get@fnmarkX{#1}{\thisc@footnote}%
3802   \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3803   \global\setbox\@nameuse{mpfootins#1}\vbox{%
3804     \unvbox\@nameuse{mpfootins#1}
3805     \noindent\csuse{bhooknoteX@#1}%
3806     \csuse{notefontsizeX@#1}%
3807     \hsize\columnwidth

```

```

3808 \parboxrestore
3809 \color@begingroup
3810 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
3811
3812 %

```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

3813 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmtX}[2]{%
3814 \ifluatex%
3815 \texdir\footnote@luatextextdir%
3816 \pardir\footnote@luatexpardir%
3817 \par%
3818 \fi%
3819 \protected@edef\@currentlabel{%
3820 \@nameuse{@thefnmark#1}%
3821 }%
3822 \ledsetnormalparstuffX{#1}%
3823 \hangindent=\csuse{hangindentX@#1}%
3824 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3825 \rule{z@}{\splittopskip%
3826 {\csuse{notenunfontX@#1}\wrapped@footfootmarkX{#1}}}%
3827 \csuse{wrapcontentX@#1}{#2}%
3828 \strut\par}}
3829
3830 %

```

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

```

3831 \newcommand*{\normalfootfootmarkX}[1]{%
3832 \textsuperscript{\@nameuse{@thefnmark#1}}}
3833
3834 %

```

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

```

3835 \newcommand*{\normalfootstartX}[1]{%
3836 \ifdimequal{0pt}{\prenotesX@}{}%
3837 {%
3838 \iftoggle{prenotesX@}{%
3839 \togglefalse{prenotesX@}%
3840 \skip\csname footins#1\endcsname=%
3841 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3842 }%
3843 }%
3844 }%
3845 \vskip\skip\csname footins#1\endcsname%
3846 \leftskip=z@

```

```

3847 \rightskip=\z@
3848 \ifl@dpairing\else%
3849   \hsize=\old@hsize%
3850 \fi%
3851 \setnoteswidthliketwocolumnsX@{#1}%
3852 \setnotesXpositionliketwocolumns@{#1}%
3853 \print@footnoteXrule{#1}%
3854 }%
3855
3856 %

```

\normalfootnoteruleX The rule drawn before the footnote series group.

```

3857 \let\normalfootnoteruleX=\footnoterule
3858
3859 %

```

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

```

3860 \newcommand*{\normalfootgroupX}[1]{%
3861   \csuse{bhookgroupX@#1}%
3862   \unvbox\@nameuse{footins#1}%
3863   \hsize=\old@hsize%
3864 }%
3865
3866 %

```

\mpnormalfootgroupX The minipage version.

```

3867 \newcommand*{\mpnormalfootgroupX}[1]{%
3868   \vskip\skip\@nameuse{mpfootins#1}
3869   \ifl@dpairing\ifparledgroup%
3870     \leavevmode\marks\parledgroup@{begin}%
3871     \marks\parledgroup@series{#1}%
3872     \marks\parledgroup@type{footnoteX}%
3873   \fi\fi\normalcolor
3874   \ifparledgroup%
3875     \ifl@dpairing%
3876     \else%
3877       \setnoteswidthliketwocolumnsX@{#1}%
3878       \setnotesXpositionliketwocolumns@{#1}%
3879       \print@footnoteXrule{#1}%
3880     \fi%
3881   \else%
3882     \setnoteswidthliketwocolumnsX@{#1}%
3883     \setnotesXpositionliketwocolumns@{#1}%
3884     \print@footnoteXrule{#1}%
3885   \fi%
3886   \csuse{bhookgroupX@#1}%

```

```

3887 \unvbox\@nameuse{mpfootins#1}}
3888
3889 %

```

\normalbfnoteX

```

3891 \newcommand{\normalbfnoteX}[2]{%
3892   \get@thisfootnoteX{#1}%
3893   \ifledRcol%
3894     \ifluatex
3895       \footnotelang@lua[R]%
3896     \fi
3897     \ifundefined{xpg@main@language}%if polyglossia
3898       {}%
3899       {\footnotelang@poly[R]}%
3900   \xright@appenditem{%
3901     \noexpand\led@set@index@fornote{#1}%
3902     \noexpand\prepare@edindex@fornote{\led@nums}%
3903     \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
3904 @reading\endcsname}%
3905     \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
3906     \noexpand\led@reinit@index@fornote%
3907     \unexpanded{\advance\@edindex@fornote@m@ne}%
3908   }%
3909     \to\inserts@listR
3910     \global\advance\insert@countR \@ne%
3911   \else%
3912     \ifluatex
3913       \footnotelang@lua%
3914     \fi
3915     \ifundefined{xpg@main@language}%if polyglossia
3916       {}%
3917       {\footnotelang@poly}%
3918   \xright@appenditem{%
3919     \noexpand\led@set@index@fornote{#1}%
3920     \noexpand\prepare@edindex@fornote{\led@nums}%
3921     \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
3922 @reading\endcsname}%
3923     \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
3924     \noexpand\led@reinit@index@fornote%
3925     \unexpanded{\advance\@edindex@fornote@m@ne}%
3926   }%
3927     \to\inserts@list
3928     \global\advance\insert@count \@ne%
3929   \fi
3930 \ignorespaces}
3931 %

```

\get@thisfootnoteX The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

3931 \newcommand{\get@thisfootnoteX}[1]{%
3932   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3933     \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
endcsname}%
3934   }{%
3935     \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}%
3936   }%
3937 }%
3938 %

```

\vbfnoteX This command calls the correct footnote-inserting commands.

```

3939 \newcommand{\vbfnoteX}[3]{%
3940   \get@fnmarkX{#1}{#3}%
3941   \@nameuse{regvfootnote#1}{#1}{#2}%
3942 }%
3943
3944 %

```

\get@fnmarkX This command gets the correct footnote number when typesetting parallel texts.

```

3945 \newcommand{\get@fnmarkX}[2]{%
3946   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3947     {%
3948       \stepcounter{footnote#1@typeset}%
3949       \setcounter{footnote#1}{\value{footnote#1@typeset}}%
3950       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3951       \immediate\write\@mainaux{%
3952         \csgdef{footnote#1@reading#2=typeset}{\the\csname c@footnote#1
@typeset\endcsname}%
3953       }%
3954     }%
3955     {%
3956       \setcounter{footnote#1}{#2}%
3957       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3958     }%
3959   }
3960 %
3961 %

```

\vnumfootnoteX₆₂ `\newcommand{\vnumfootnoteX}[2]{%`
`\ifnumberedpar@`
`\edtext{}{\normalbfnoteX{#1}{#2}}%`
`\else`

```

3966 \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3967 \get@thisfootnoteX{#1}%
3968 \get@fnmarkX{#1}{\expandonce\thisc@footnote}%
3969 \@nameuse{regvfootnote#1}{#1}{#2}%
3970 \fi}
3971
3972 %

```

`arrangementX@normal` `\arrangementX@normal{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```

3973 \newcommand*{\arrangementX@normal}[1]{%
3974 \csgdef{series@displayX#1}{normal}
3975 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
3976 \@namedef{footnotemark#1}{\normal@footnotemarkX{#1}}
3977 \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
3978 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
3979 \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
3980 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
3981 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
3982 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
3983 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3984 \count\csname footins#1\endcsname=1000
3985 \csxdef{default@footins#1}{1000}%Use to have note only for one side
3986 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3987 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3988 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3989 %

```

Additions for minipages.

```

3990 \ifnoledgroup@else%
3991 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3992 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
3993 \count\csname mpfootins#1\endcsname=1000
3994 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
3995 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3996 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3997 \fi
3998 }
3999
4000 %

```

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\newcommand*{\arrangementX@twocol}[1]{%
4002 \csgdef{series@displayX#1}{twocol}

```

```

4003 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4004 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4005 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4006 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4007 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4008 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4009 \twocolfootsetupX{#1}
4010 \ifnoledgroup@else%
4011   \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4012   \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4013   \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4014   \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4015   \mptwocolfootsetupX{#1}
4016 \fi%
4017 }
4018
4019 %

```

`\twocolfootsetupX` `\twocolfootsetupX{<series>}`

```

\mptwocolfootsetupX
4020 \newcommand*{\twocolfootsetupX}[1]{%
4021   \count\csname footins#1\endcsname 500
4022   \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
4023   \multiply\dimen\csname footins#1\endcsname by \tw@}
4024 \newcommand*{\mptwocolfootsetupX}[1]{%
4025   \count\csname mpfootins#1\endcsname 500
4026   \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
4027
4028 %

```

`\twocolvfootnoteX` `\twocolvfootnoteX{<series>}`

```

4029 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
4030   \csuse{beforeinsertingX@#1}%
4031   \insert\csname footins#1\endcsname\bgroup%
4032     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4033     \noindent\csuse{bhooknoteX@#1}%
4034     \csuse{notefontsizeX@#1}%
4035     \footsplitskips%
4036     \spaceskip=\z@skip \xspaceskip=\z@skip%
4037     \@nameuse{footfmt#1}{#1}{#2}\egroup}
4038
4039 %

```

`\twocolfootfmtX` `\twocolfootfmtX{<series>}`

```

4040 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4041   \protected@edef@currentlabel{%
4042     \@nameuse{@thefnmark#1}%

```

```

4043 }%
4044 \normal@pars%
4045 \hangindent=\csuse{hangindentX@#1}%
4046 \everypar{\hangindent=\csuse{hangindentX@#1}}%
4047 \hsize \csuse{hsizewocolX@#1}%
4048 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4049 \tolerance=5000\relax%
4050 \par%
4051 \@tempdima=\parindent%
4052 \csuse{colalignX@#1}%
4053 \parindent=\@tempdima%
4054 {\hspace{\parindent}}%
4055 \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4056 \csuse{wrapcontentX@#1}{#2}%
4057 \strut\par}%
4058 \allowbreak%
4059 }%
4060 %
4061 %

```

```

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX
4062 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4063 \splittopskip=\ht\strutbox
4064 \expandafter
4065 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4066
4067 \newcommand*{\mptwocolfootgroupX}[1]{%
4068 \vskip\skip\@nameuse{mpfootins#1}
4069 \ifl@dpairing\ifparledgroup%
4070 \leavevmode\marks\parledgroup@{begin}%
4071 \marks\parledgroup@series{#1}%
4072 \marks\parledgroup@type{footnoteX}%
4073 \fi\fi\normalcolor
4074 \ifparledgroup%
4075 \ifl@dpairing%
4076 \else%
4077 \setnoteswidthliketwocolumnsX@{#1}%
4078 \setnotesXpositionliketwocolumns@{#1}%
4079 \print@footnoteXrule{#1}%
4080 \fi%
4081 \else%
4082 \setnoteswidthliketwocolumnsX@{#1}%
4083 \setnotesXpositionliketwocolumns@{#1}%
4084 \print@footnoteXrule{#1}%
4085 \fi%
4086 \csuse{bhookgroupX@#1}%
4087 \splittopskip=\ht\strutbox
4088 \expandafter

```



```

4089 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4090
4091 %

```

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@threecolX  \newcommand*\arrangementX@threecolX[1]{%
4093 \csgdef{series@displayX#1}{threecol}
4094 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4095 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4096 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4097 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4098 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4099 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4100 \threecolfootsetupX{#1}
4101 \ifnoledgroup@ \else%
4102 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4103 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4104 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4105 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4106 \mpthreecolfootsetupX{#1}
4107 \fi%
4108 }
4109
4110 %

```

```

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX
4111 \newcommand*\threecolfootsetupX[1]{%
4112 \count\csname footins#1\endcsname 333
4113 \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4114 \multiply\dimen\csname footins#1\endcsname by \thr@@}
4115 \newcommand*\mpthreecolfootsetupX[1]{%
4116 \count\csname mpfootins#1\endcsname 333
4117 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
4118
4119 %

```

```

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
4120 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%
4121 \csuse{beforeinsertingX@#1}%
4122 \insert\csname footins#1\endcsname\bgroup%
4123 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%

```

```

4124 \noindent\csuse{bhooknoteX@#1}%
4125 \csuse{notefontsizeX@#1}%
4126 \footsplitskips%
4127 \@nameuse{footfmt#1}{#1}{#2}\egroup}
4128
4129 %

```

`\threecolfootfmtX` `\threecolfootfmtX{<series>}`

```

4130 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4131 \protected@edef\@currentlabel{%
4132 \@nameuse{thefnmark#1}%
4133 }%
4134 \hangindent=\csuse{hangindentX@#1}%
4135 \everypar{\hangindent=\csuse{hangindentX@#1}}%
4136 \normal@pars%
4137 \hsize \csuse{hsizethreecolX@#1}%
4138 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4139 \tolerance=5000\relax%
4140 \@tempdima=\parindent%
4141 \csuse{colalignX@#1}%
4142 \parindent=\@tempdima%
4143 f\hspace{\parindent}%
4144 \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4145 \csuse{wrapcontentX@#1}{#2}%
4146 \strut\par}\allowbreak}
4147
4148 %

```

`\threecolfootgroupX` `\threecolfootgroupX{<series>}`
`\mpthreecolfootgroupX`

```

4149 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4150 \splittopskip=\ht\strutbox
4151 \expandafter
4152 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4153
4154 \newcommand*{\mpthreecolfootgroupX}[1]{%
4155 \vskip\skip\@nameuse{mpfootins#1}
4156 \ifl@dpairing\ifparledgroup
4157 \leavevmode\marks\parledgroup@{begin}%
4158 \marks\parledgroup@series{#1}%
4159 \marks\parledgroup@type{footnoteX}%
4160 \fi\fi\normalcolor
4161 \ifparledgroup%
4162 \ifl@dpairing%
4163 \else%
4164 \setnoteswidthliketwocolumnsX@{#1}%
4165 \setnotesXpositionliketwocolumns@{#1}%
4166 \print@footnoteXrule{#1}%

```

```

4167 \fi%
4168 \else%
4169 \setnoteswidthliketwocolumnsX@{#1}%
4170 \setnotesXpositionliketwocolumns@{#1}%
4171 \print@footnoteXrule{#1}%
4172 \fi%
4173 \cuse{bhookgroupX@#1}%
4174 \splittopskip=\ht\strutbox
4175 \expandafter
4176 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4177
4178 %

```

XIII.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@threecol` `\footparagraphX{<series>}`

```

4179 \newcommand*{\arrangementX@paragraph}[1]{%
4180 \csgdef{series@displayX#1}{paragraph}%
4181 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4182 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4183 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4184 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4185 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4186 \count\csname footins#1\endcsname=1000
4187 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
4188 \dimen\csname footins#1\endcsname=\cuse{maxhnotesX@#1}
4189 \skip\csname footins#1\endcsname=\cuse{beforenotesX@#1}%
4190 \advance\skip\csname footins#1\endcsname by\cuse{afterruleX@#1}%
4191 \para@footsetupX{#1}
4192 \ifnoledgroup@else
4193 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4194 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4195 \count\csname mpfootins#1\endcsname=1000
4196 \dimen\csname mpfootins#1\endcsname=\cuse{maxhnotesX@#1}
4197 \skip\csname mpfootins#1\endcsname=\cuse{beforenotesX@#1}%
4198 \advance\skip\csname mpfootins#1\endcsname by\cuse{afterruleX@#1}%
4199 \fi
4200 }
4201
4202 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

4203 \newcommand*{\para@footsetupX}[1]{\cuse{bhookgroupX@#1}\cuse{
notefontsizeX@#1}
4204 \setnoteswidthliketwocolumnsX@{#1}%

```

```

4205 \ifcempty{widthX@#1}%
4206 {}%
4207 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4208 \dimen0=\baselineskip
4209 \multiply\dimen0 by 1024
4210 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
4211 %
4212 \expandafter
4213 \xdef\csname footfudgefactor#1\endcsname{%
4214   \expandafter\strip@pt\dimen0 }}
4215 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

4216 \newcommand*{\parafootstartX}[1]{%
4217   \ifdimequal{Opt}{\prenotesX@}{}%
4218   {%
4219     \iftoggle{prenotesX@}{%
4220       \togglefalse{prenotesX@}%
4221       \skip\csname footins#1\endcsname=%
4222       \glueexpr\csuse{prenotesX@}+\csuse{aftererruleX@#1}\relax%
4223     }%
4224   }%
4225 }%
4226 \leftskip=\z@
4227 \rightskip=\z@
4228 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4229 \vskip\skip\@nameuse{footins#1}%
4230 \setnoteswidthliketwocolumnsX@{#1}%
4231 \setnotesXpositionliketwocolumns@{#1}%
4232 \print@footnoteXrule{#1}%
4233 }
4234
4235 %

```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`
`\mppara@vfootnoteX`

```

4236 \newcommand*{\para@vfootnoteX}[2]{%
4237   \csuse{beforeinsertingX@#1}%
4238   \insert\csname footins#1\endcsname%
4239   \bgroup
4240     \csuse{notefontsizeX@#1}
4241     \footsplitskips
4242     \setbox0=\vbox{\hsize=\maxdimen%
4243       \let\bidir@RTL@everypar\@empty%
4244       \noindent\csuse{bhooknoteX@#1}%
4245       \@nameuse{footfmt#1}{#1}{#2}}%
4246     \setbox0=\hbox{\unvvhX{0}{#1}}%
4247     \dp0=\z@

```

```

4248 \ht0=\csname footfudgefactor#1\endcsname\wd0
4249 \box0
4250 \penalty0
4251 \egroup}
4252 \newcommand*{\mppara@vfootnoteX}[2]{%
4253 \get@thisfootnoteX{#1}%
4254 \get@fnmarkX{#1}{\thisc@footnote}%
4255 \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4256 \global\setbox\@nameuse{mpfootins#1}\vbox{%
4257 \unvbox\@nameuse{mpfootins#1}
4258 \csuse{notefontsizeX@#1}
4259 \footsplitskips
4260 \setbox0=\vbox{\hsize=\maxdimen%
4261 \let\bidir@RTL@everypar\@empty%
4262 \noindent\color@begingroup%
4263 \csuse{bhooknoteX@#1}%
4264 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
4265 \setbox0=\hbox{\unvxhX{0}{#1}}%
4266 \dp0=\z@
4267 \ht0=\csname footfudgefactor#1\endcsname\wd0
4268 \box0
4269 \penalty0}}
4270
4271 %

```

`\unvxhX`₇₂ \newcommand*{\unvxhX}[2]{% 2th is optional for retro-compatibility

```

4273 \setbox0=\vbox{\unvbox#1%
4274 \global\setbox1=\lastbox}%
4275 \unhbox1
4276 \unskip % remove \rightskip,
4277 \unskip % remove \parfillskip,
4278 \unpenalty % remove \penalty of 10000,
4279 \hskip\csuse{afternoteX@#2}%
4280 \relax}% but add the glue to go between the notes
4281
4282 %

```

`\parafootfmtX` \parafootfmtX{*series*}

```

4283 \newcommand*{\parafootfmtX}[2]{%
4284 \protected@edef\@currentlabel{%
4285 \@nameuse{@thefnmark#1}%
4286 }%
4287 \insertparafootsepX{#1}%
4288 \ledsetnormalparstuff@common%
4289 {\csuse{notenumfontX@#1}%
4290 \csuse{notenumfontX@#1}%
4291 \wrapped@footfootmarkX{#1}%
4292 \strut%

```

```

4293 \csuse{wrapcontentX@#1}{#2}%
4294 \penalty-10}}
4295
4296 %

```

```

\para@footgroupX \para@footgroupX{<series>}
\mppara@footgroupX
4297 \newcommand*{\para@footgroupX}[1]{%
4298 \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4299 \unvbox\csname footins#1\endcsname
4300 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4301 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4302 \makeboxofhboxes
4303 \setbox0=\hbox{\unhbox0 \removehboxes}%
4304 \csuse{hookgroupX@#1}
4305 \csuse{notefontsizeX@#1}
4306 \unhbox0\par}
4307
4308 \newcommand*{\mppara@footgroupX}[1]{%
4309 \setnoteswidthliketwocolumnsX@{#1}%
4310 \vskip\skip\@nameuse{mpfootins#1}
4311 \ifl@dpairing\ifparledgroup
4312 \leavevmode%
4313 \leavevmode\marks\parledgroup@{begin}%
4314 \marks\parledgroup@series{#1}%
4315 \marks\parledgroup@type{footnoteX}%
4316 \fi\fi\normalcolor
4317 \ifparledgroup%
4318 \ifl@dpairing%
4319 \else%
4320 \setnoteswidthliketwocolumnsX@{#1}%
4321 \setnotesXpositionliketwocolumns@{#1}%
4322 \print@footnoteXrule{#1}%
4323 \fi%
4324 \else%
4325 \setnoteswidthliketwocolumnsX@{#1}%
4326 \setnotesXpositionliketwocolumns@{#1}%
4327 \print@footnoteXrule{#1}%
4328 \fi%
4329 \unvbox\csname mpfootins#1\endcsname
4330 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4331 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4332 \makeboxofhboxes
4333 \setbox0=\hbox{\unhbox0 \removehboxes}%
4334 \csuse{hookgroupX@#1}%
4335 \csuse{notefontsizeX@#1}%
4336 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4337 \unhbox0\par}}
4338
4339 %

```

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

```
\insertparafootsepX40 \newcommand{\insertparafootsepX}[1]{%
4341 \ifledRcol{%
4342 \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
4343 {\csuse{Xparafootsep#1}}%
4344 {}%
4345 \global\csname prevpage#1@numR\endcsname=\page@numR%
4346 \else%
4347 \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
4348 {\csuse{Xparafootsep#1}}%
4349 {}%
4350 \global\csname prevpage#1@num\endcsname=\page@num%
4351 \fi%
4352 }
4353 %
```

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.

```
4354 \newcommand{\wrapped@footfootmarkX}[1]{%
4355 \ifdefined\hypertarget%
4356 \hyperlink%
4357 {@bodyfootmark#1@\this@footnoteX@reading}%
4358 {\@nameuse{footfootmark#1}}%
4359 \Hy@raisedlink{%
4360 \hypertarget%
4361 {@footnotemark#1@\this@footnoteX@reading}%
4362 {}%
4363 }%
4364 \else%
4365 \@nameuse{footfootmark#1}%
4366 \fi%
4367 }%
4368 %
```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package's commands, if needed.

```
4369 \newcommand{\wrapped@bodyfootmarkX}[1]{%
4370 \ifdefined\hypertarget%
4371 \hyperlink%
4372 {@footnotemark#1\expandafter\the\csname footnote#1@reading\endcsname}%
4373 {\@nameuse{bodyfootmark#1}}%
4374 \Hy@raisedlink{%
```

```

4375 \hypertarget%
4376 {\bodyfootmark#1@\expandafter\the\csname footnote#1@reading\
endcsname}%
4377 {}%
4378 }%
4379 \else%
4380 \@nameuse{bodyfootmark#1}%
4381 \fi%
4382 }%
4383 %

```

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 Lua[®]TeX.

That why we have defined \ledsetnormalparstuff@common in order to make this setting for both familiar and critical notes. This command is called by command to make specific setting to critical or familiar footnote.

```

\ledsetnormalparstuff@common%4 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff%5 \ifluatex%
\ledsetnormalparstuffX%6 \ifdefstring{\footnote@luatextextdir}{TLT}{}%
4387 {\textdir\footnote@luatextextdir}%
4388 \pardir\footnote@luatexpardir%
4389 \fi%
4390 \csuse{\csuse{footnote@dir}}%
4391 \normal@pars%
4392 \parfillskip \z@ \@plus 1fil}%
4393
4394 \newcommand*{\Xledsetnormalparstuff}[1]{%
4395 \ledsetnormalparstuff@common%
4396 \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
4397 }%
4398
4399 \newcommand*{\ledsetnormalparstuffX}[1]{%
4400 \ledsetnormalparstuff@common%
4401 \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}%
4402 }%
4403 %

```


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.

```

4404
4405 \newdimen\old@hsize%
4406 \AtBeginDocument{\old@hsize=\hsize}%
4407
4408 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
4409   \global\let\hsize@fornote=\hsize%
4410   \global\old@hsize=\hsize%
4411   \let\old@columnwidth=\columnwidth%
4412   \iftoggle{Xnoteswidthliketwocolumns@#1}%
4413     {%
4414       \csuse{setwidthliketwocolumns@\columns@position}%
4415       \global\let\hsize@fornote=\hsize%
4416     }%
4417     {}%
4418   \let\hsize=\hsize@fornote%
4419   \let\columnwidth=\old@columnwidth%
4420 }%
4421
4422 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4423   \global\let\hsize@fornote=\hsize%
4424   \global\old@hsize=\hsize%
4425   \let\old@columnwidth=\columnwidth%
4426   \iftoggle{noteswidthliketwocolumnsX@#1}%
4427     {%
4428       \csuse{setwidthliketwocolumns@\columns@position}%
4429       \global\let\hsize@fornote=\hsize%
4430     }%
4431     {}%
4432   \let\hsize=\hsize@fornote%
4433   \let\columnwidth=\old@columnwidth%
4434 }%
4435
4436 %

```

`\espositionliketwocolumns@` 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`.

```

4437 \newcommand{\setXnotespositionliketwocolumns@}[1]{%

```

```

4438 \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4439 \csuse{setnotespositionliketwocolumns@\columns@position}%
4440 }{}%
4441 }%
4442
4443 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
4444 \iftoggle{noteswidthliketwocolumnsX@#1}{%
4445 \csuse{setnotespositionliketwocolumns@\columns@position}%
4446 }{}%
4447 }%
4448
4449 %

```

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.

```

4450 \def\@fnpos{familiar-critical}
4451 \def\@mpfnpos{critical-familiar}
4452 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
4453 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
4454 %

```

XVII Footnotes' rule

Because the footnotes' rules can be shifted to the right when footnotes are set like two columns, we do not print them directly, but we put them in a `\vbox`.

```

\print@Xfootnoterule55 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoterule56 \vskip-\csuse{Xafterterrule@#1}%Because count in \dimen\csuse{#1footins}
4457 \nointerlineskip%
4458 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
4459 \nointerlineskip%
4460 \vskip\csuse{Xafterterrule@#1}%
4461 }%
4462
4463 \newcommand{\print@footnoterule}[1]{%
4464 \vskip-\csuse{afterterruleX@#1}%Because count in \dimen\csuse{footins#1}
4465 \nointerlineskip%
4466 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
4467 \nointerlineskip%
4468 \vskip\csuse{afterterruleX@#1}%
4469 }%
4470
4471 %

```

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@Xprenotes` before inserting any critical notes, in order to prevent page number overlapping.

1. If it is the first note of the current page, it changes the footnote skip for the series to the value specified to `\Xbeforenotes`. It also keeps the series of the note as the first one of the current page.
2. If it is not the first note of the current page:
 - If the current series is printed after the series kept as the first of the current page, then nothing happens.
 - If the current series is printed before the series kept as the first of the current page, then it changes the footnote skip of the current series to the value normally used by the series which was marked as the first of the page. It also keeps the current series as the new first one of the current page.

For example, suppose the series order is A,B. We call first a `\Bfootnote` and a `\Afootnote`. The only skips used are, finally, the skip specific to the first series of the page, and the skip for the B series. If we have not called `\Afootnote`, the only skip used is the skip specific to the first series of the page.

That is perfect.

The series skip and the first series of the current page are reset before the footnotes are printed. Then, the `footstart` macros manage the problem of the first series of the page.

After the rule, the space which is defined by `\Xafterrule` does not depend on whether the series is the first one of the page or not. So we use its normal value for each series.

And now, implementation !

XVIII.0.2 User level command

`\Xprenotes@` If user redefines `\Xprenotes@`, via `\Xprenotes` to a value greater than 0 pt, this skip will be added before first series notes instead of the notes skip.

```

4472 \newtoggle{Xprenotes@}%
4473 \toggletrue{Xprenotes@}%
4474 \newcommand{\Xprenotes@}{0pt}%
4475 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
4476 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
4477 %

```

The same, but for familiar footnotes.

```

\Xprenotes 78 \newtoggle{prenotesX@}
\Xprenotes@ 79 \toggletrue{prenotesX@}
4480 \newcommand{\prenotesX@}{Opt}
4481 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
4482 %

```

XVIII.0.3 Internal commands

```

firstXseries@ 83 \gdef\firstXseries@{}
prepare@Xprenotes 84 \newcommand{\prepare@Xprenotes}[1]{%
4485 \ifdimequal{Opt}{\Xprenotes@}%
4486 {}%
4487 {%
4488 \IfStrEq{\firstXseries@}{}{%
4489 \global\skip\csuse{#1footins}=\Xprenotes@%
4490 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4491 \gdef\firstXseries@{#1}%
4492 }%
4493 {%
4494 \ifseriesbefore{#1}{\firstXseries@}%
4495 {%
4496 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@firstXseries@}%
4497 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4498 \gdef\firstXseries@{#1}%
4499 }%
4500 {%
4501 }%
4502 }%
4503 }
4504 %

```

The same thing is required for familiar notes and \prenotesX.

```

firstseriesX@ 05 \gdef\firstseriesX@{}
prepare@prenotesX 06 \newcommand{\prepare@prenotesX}[1]{%
4507 \ifdimequal{Opt}{\prenotesX@}%
4508 {}%
4509 {%
4510 \IfStrEq{\firstseriesX@}{}{%
4511 \global\skip\csuse{footins#1}=\prenotesX@%
4512 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4513 \gdef\firstseriesX@{#1}%
4514 }%
4515 {%
4516 \ifseriesbefore{#1}{\firstseriesX@}%

```

```

4517     {%
4518     \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
4519     \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4520     \gdef\firstXseries@{#1}%
4521     }%
4522     {}%
4523     }%
4524     }%
4525 }
4526 %

```

XIX Endnotes

First, check the noend option.

```

4527 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
4528 %

```

XIX.1 Internal commands

`\l@dend@open` `\l@dend@open` and `\l@dend@close` are the macros that are used to open and close the endnote file. Note that all our writing to this file is `\immediate`: all page and line numbers for the endnotes are generated by the same mechanism we use for the footnotes, so that there is no need to defer any writing to catch information from the output routine. The argument of these two command is the series letter.

```

4529 \newcommand{\l@dend@open}[1]{%
4530     \global\booltrue{l@dend@#1}%
4531     \expandafter\immediate%
4532     \expandafter\openout%
4533     \csname l@d@#1end\endcsname%
4534     =\l@auxdir\jobname.#1end\relax%
4535 }%
4536 \newcommand{\l@dend@close}[1]{%
4537     \global\boolfalse{l@dend@#1}%
4538     \expandafter\immediate%
4539     \expandafter\closeout\csname l@d@#1end\endcsname%
4540 }%
4541
4542 %

```

`\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.

```

4543 \newcommand{\l@dend@stuff}{%
4544     \def\do##1{%

```

```

4545 \ifbool{l@dend@##1}{}%
4546   {\l@dend@open{##1}}%
4547   \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
4548   }%
4549   \dolistloop{@series}%
4550 }%
4551
4552 %

```

`\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.

```

4553 \global\newbool{parapparatus@}{\long}\def\endprint#1#2#3#4#5#6#7{%
4554   \csuse{Xendbhooknote@#4}%
4555   \csuse{Xendnotefontsize@#4}%
4556   \hangindent=\csuse{Xendhangindent@#4}%
4557   \ifXendinsertsep@%
4558     \hskip\csuse{Xendafternote@#4}\relax%
4559     \csuse{Xendsep@#4}%
4560   \else%
4561     \iftoggle{Xendparagraph@#4}%
4562       {\global\Xendinsertsep@true}%
4563     }%
4564   \fi%
4565   \xdef\@currentseries{#4}%
4566   \def\do##1{%
4567     \setkeys[mac]{truefootnoteoption}{##1}%
4568   }%
4569   \notblank{#5}{\docsvlist{#5}}{}%
4570   \IfStrEq{#6}{R}{\ledRcol@true}{}%
4571   \def\@this@crossref@start{#7:start}%
4572   \def\@this@crossref@end{#7:end}%

```

```

4573 \printlineendnote{#1}{#4}%
4574 \IfStrEq{#6}{R}{\ledRcol@false}{}%
4575 \undef\@this@crossref@start%
4576 \undef\@this@crossref@end%
4577 \nottoggle{Xendlemmadisablefontselection@#4}%
4578   {\select@lemmafont#1|}%
4579   {}%
4580 \bgroup%
4581   \csuse{Xendlemmafont@#4}%
4582   \csuse{Xendwraplemma@#4}{#2}%
4583 \egroup%
4584 \ifboolexpr{%
4585   togl {nosep@}%
4586   or test{\ifcempty{Xendlemmaseparator@#4}}%
4587   }%
4588   {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
4589   {\nobreak%
4590     \hskip\csuse{Xendbeforelemmaseparator@#4}%
4591     \csuse{Xendlemmaseparator@#4}%
4592     \hskip\csuse{Xendafterlemmaseparator@#4}%
4593     \relax%
4594   }%
4595   \csuse{Xendwrapcontent@#4}{#3}%
4596   \nottoggle{Xendparagraph@#4}{\par}{}%
4597   \def\do##1{%
4598     \setkeys[mac]{falsefootnoteoption}{##1}%
4599   }%
4600   \notblank{#5}{\docsvlist{#5}}}%
4601 }}%
4602
4603 \let\l@d@section=\@gobble
4604
4605 %

```

\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.

```

4606 \newcommand{\printlineendnote}[2]{%
4607   \l@dp@rsefootspec#1|%
4608   \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
4609     \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
4610       \l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
4611       \l@dparsedendsub}%
4612     \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
4613       \l@dparsedstartsub}%
4614   }%
4615   \ifboolexpr{%

```

```

4615     tog1 {nonum@}%
4616     or tog1 {Xendnonumber@#2}%
4617     }%
4618     {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4619     {%
4620     \iftoggle{Xendnumberonlyfirstinline@#2}%
4621     {\ifcdef{prevendline#2}%
4622     {\ifcsequal{prevendline#2}{lineinfo@}%
4623     {%
4624     \csuse{Xendbhookinplaceofnumber@#2}%
4625     \ifcempty{Xendsymmlinenumber@#2}%
4626     {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4627     {\printsymlineendnotearea{#2}}}%
4628     \csuse{Xendahookinplaceofnumber@#2}%
4629     }%
4630     {\printlineendnotearea{#1}{#2}}}%
4631     {\printlineendnotearea{#1}{#2}}}%
4632     }%
4633     {\printlineendnotearea{#1}{#2}}%We keep every time line
4634     \csxdef{prevendline#2}{\lineinfo@}%
4635     }%
4636 }%
4637 %

```

```

\printsymlineendnotearea{#1}{%
4639 \hspace{\csuse{Xendbeforesymmlinenumber@#1}}%
4640 \csuse{Xendnotenumfont@#1}%
4641 \ifdimequal{\csuse{Xendboxsymmlinenumber@#1}}{\z@}%
4642 {\csuse{Xendsymmlinenumber@#1}}%
4643 {\hbox to \csuse{Xendboxsymmlinenumber@#1}%
4644 {\csuse{Xendsymmlinenumber@#1}\hfill}%
4645 }%
4646 \hspace{\csuse{Xendaftersymmlinenumber@#1}}%
4647 }%
4648 %

```

\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.)

```

4649 \newcommand{\printlineendnotearea}[2]{%
4650 \csuse{Xendbhooklinenumber@#2}%
4651 \hspace{\csuse{Xendbeforenumber@#2}}%
4652 \bgroup%
4653 \csuse{Xendnotenumfont@#2}%
4654 \ifdimequal{\csuse{Xendboxlinenumber@#2}}{\Opt}%
4655 {\printendlines#1||\ifledRcol@{\Rlineflag\fi}%
4656 {\leavevmode%

```



```

4657 \hbox to \csuse{Xendboxlinenum@#2}%
4658 {%
4659 \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4660 \printendlines#1||\ifledRcol@\@Rlineflag\fi%
4661 \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4662 }%
4663 \egroup%
4664 \hspace{\csuse{Xendafternumber@#2}}%
4665 \csuse{Xendahooklinenumber@#2}%
4666 }%
4667 %

```

XIX.2 User level commands

XIX.2.1 Inserting contents to endnotes

The `\Xendnotes` commands are defined above, when defining apparatus commands by series. Here, we define only `\toendnotes` command not specific to a series, in order to insert arbitrary code. The regular version writes an unexpanded argument, while the regular version writes a once-expanded argument.

```

\toendnotes*68 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes*69 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4670 \def\do##1{%
4671 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4672 {\unexpanded{#2}\@percentchar}%
4673 }%
4674 \ifstrempy{#1}%
4675 {\dolistloop{\@series}}%
4676 {\docsvlist{#1}}%
4677 }\led@err@toendnotes@outsidenumbering}%
4678 }%
4679 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
4680 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4681 \def\do##1{%
4682 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4683 {#2\@percentchar}%
4684 }%
4685 \ifstrempy{#1}%
4686 {\dolistloop{\@series}}%
4687 {\docsvlist{#1}}%
4688 }\led@err@toendnotes@outsidenumbering}%
4689 }%
4690 %

```

XIX.2.2 Printing endnotes

`\doendnotes` `\doendnotes` is the command you use to print one series of endnotes; it takes one argument: the series letter of the note series you want to print. `\Xendinsertsep@`

set to true at the first note of the series, and to false at the last one.

```

4691 \newif\ifXendinsertsep%
4692 \newcommand*{\doendnotes}[1]{%
4693   \l@dend@close{#1}%
4694   \begingroup
4695     \csxdef{prevpagenum@#1}{}%
4696     \csxdef{prevpagerange@#1}{}%
4697     \makeatletter
4698     \expandafter\let\csname #1end\endcsname=\endprint
4699     \input\l@auxdir\jobname.#1end%
4700     \global\Xendinsertsep@false%
4701   \endgroup}
4702 %

```

\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.

```

4703 \newcommand*{\doendnotesbysection}[1]{%
4704   \l@dend@close{#1}%
4705   \csxdef{prevpagenum@#1}{}%
4706   \csxdef{prevpagerange@#1}{}%
4707   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
4708   \begingroup%
4709     \makeatletter%
4710     \def\l@d@section##1{%
4711       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
4712       {\cslet{#1end}{\endprint}}%
4713       {\cslet{#1end}{\@gobbleseven}}%
4714     }%
4715     \input\l@auxdir\jobname.#1end%
4716     \global\Xendinsertsep@false%
4717   \endgroup%
4718 }%
4719 %

```

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.

```

4720 }%
4721 %

```

\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.

```

4722 \newcommand*{\setprintendlines}[6]{%
4723   \l@d@pnumfalse \l@d@dashfalse
4724   \ifnum#4=#1 \else
4725     \l@d@pnumtrue
4726     \l@d@dashtrue
4727   \fi
4728   %

```

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.

```

4729   \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
4730   \ifnum#2=#5 \else
4731     \l@d@elintrue
4732     \l@d@dashtrue
4733   \fi
4734   %

```

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

```

4735   \l@d@ssubfalse
4736   \ifnum#3=0 \else
4737     \l@d@ssubtrue
4738   \fi
4739   %

```

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.

```

4740   \l@d@eslfalse
4741   \ifnum#6=0 \else
4742     \ifnum#6=#3
4743       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4744     \else
4745       \l@d@esltrue
4746       \l@d@dashtrue
4747     \fi
4748   \fi%
4749   %

```

```

4750   \ifl@d@dash%
4751   \ifboolexpr{togl{fulllines@} or test{\ifcsemt{Xendtwolines@}\
@currentseries}}}%
4752   {}%
4753   {%
4754   \setistwofollowinglines{#1}{#2}{#4}{#5}%

```

```

4755 \ifboolexpr{%
4756   (%
4757     togl {Xendtwolinesbutnotmore@\@currentseries}%
4758     and not%
4759       (%
4760         bool {istwofollowinglines@}%
4761       )%
4762   )%
4763   or%
4764   (%
4765     (not test{\ifnumequal{#1}{#4}})%
4766     and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
4767   )%
4768   }%
4769   {%
4770     {%
4771       \l@d@dashfalse%
4772       \l@d@Xtwolinestrue%
4773       \l@d@elinfalse%
4774       \l@d@eslfalse%
4775       \ifcempty{Xendmorethantwolines@\@currentseries}%
4776         {%
4777           {\ifistwofollowinglines@\else%
4778             \l@d@Xmorethantwolinestrue%
4779           }%
4780         }%
4781       }%
4782     }%
4783   \fi%
4784   %

End of \setprintendlines.

4785 }%
4786 %

```

`\printendlines` Now we are ready to print it all.

```

4787 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4788   \begingroup
4789   \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
4790   %

```

The only subtlety left here is when to print a period between numbers. But the only instance in which this is tricky is for the ending sub-line number: it could be coming after the starting sub-line number (in which case we want only the dash) or after an ending line number (in which case we need to insert a period).

So, first, start the starting line box, if needed.

```

4791 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
4792   {\bgroup}%

```

```

4793 {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}%
4794 %

```

Then, print the starting page number-

```

4795 \ifboolexpr{%
4796   (%
4797     test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
4798     and not%
4799     (togl{Xendpagenumberonlyfirsttifsingl@\@currentseries} and bool{
l@d@pnum}})%
4800   )%
4801   or%
4802   (%
4803     test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
4804     )%
4805   }%
4806   {%
4807     \ifcempty{Xendsympagenum@\@currentseries}%
4808     {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
4809     {\csuse{Xendsympagenum@\@currentseries}}%
4810   }%
4811   {%
4812     \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
4813   }%
4814 %

```

Then, determine what must be printed before the start line.

```

4815 \ifl@d@dash%
4816 \ifl@d@pnum%
4817   \csuse{Xendlineprefixsingle@\@currentseries}%
4818 \else%
4819   \ifcempty{Xendlineprefixmore@\@currentseries}%
4820   {\csuse{Xendlineprefixsingle@\@currentseries}}%
4821   {\csuse{Xendlineprefixmore@\@currentseries}}%
4822 \fi%
4823 \else%
4824   \csuse{Xendlineprefixsingle@\@currentseries}%
4825 \fi%
4826 %

```

The print the starting line, followed, if needed, by the side flag and the starting sub line number.

```

4827 \wrap@edcrossref{\@this@crossref@start}{%
4828   \ifledRcol%
4829     \linenumrepR{#2}%
4830   \else%
4831     \linenumrep{#2}%
4832   \fi%

```

```

4833 }%
4834 \iftoggle{Xendlineflag@{\@currentseries}{\ifledRcol@{\@Rlineflag\fi}{}}%
4835 \ifl@d@ssub%
4836   \csuse{Xendsublinesep@{\@currentseries}}%
4837   \wrap@edcrossref{\@this@crossref@start}{}%
4838   \ifledRcol@%
4839     \sublinenumrepR{#3}%
4840   \else%
4841     \sublinenumrep{#3}%
4842   \fi%
4843 }%
4844 \fi%
4845 %

```

Close the box.

```

4846 \egroup%
4847 %

```

Open the box for the ending line number.

```

4848 \ifdimequal{\csuse{Xendboxendlinenumber@{\@currentseries}}{0pt}}%
4849 {\bgroup}%
4850 {\hbox to \csuse{Xendboxendlinenumber@{\@currentseries}}\bgroup}%
4851 %

```

Print the dash + the ending line number, or the line number range symbol.

```

4852 \ifl@d@Xtwolines%
4853   \ifl@d@Xmorethantwolines%
4854     \csuse{Xendmorethantwolines@{\@currentseries}}%
4855   \else%
4856     \csuse{Xendtwolines@{\@currentseries}}%
4857   \fi%
4858 \else%
4859   \ifl@d@dash%
4860     \ifdefined\linangesep@%
4861       \linangesep@%
4862     \else%
4863       \csuse{Xendlinangeseparator@{\@currentseries}}%
4864     \fi%
4865   \fi%
4866 %

```

Print the ending page number.

```

4867 \ifl@d@pnum%
4868   \ifcsstring{prevpagerange@{\@currentseries}}{#1-#4}%
4869   {%
4870     \ifcempty{Xendsympagenum@{\@currentseries}}%
4871       {\hspace{\csuse{Xendinplaceofpagenumber@{\@currentseries}}}}%
4872       {\csuse{Xendsympagenum@{\@currentseries}}}%
4873     }%

```

```

4874     {%
4875     \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
4876     }%
4877     \fi%
4878 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag and the subtitle number.

```

4879     \ifl@d@elin%
4880     \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
4881     \wrap@edcrossref{\@this@crossref@end}{%
4882     \ifledRcol%
4883     \linenumrepR{#5}%
4884     \else%
4885     \linenumrep{#5}%
4886     \fi%
4887     }%
4888     \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
4889     \fi%
4890     \ifl@d@esl%
4891     \ifl@d@elin%
4892     \csuse{Xendsublinesep@\@currentseries}%
4893     \fi%
4894     \wrap@edcrossref{\@this@crossref@end}{%
4895     \ifledRcol%
4896     \sublinenumrepR{#6}%
4897     \else%
4898     \sublinenumrep{#6}%
4899     \fi%
4900     }%
4901     \fi%
4902     \fi%
4903 %

```

Close the ending line box.

```

4904     \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
4905     {}%
4906     {\hfill}%Prevent underfull hbox
4907     \egroup%
4908 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

4909     \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
4910     {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%
4911     {\csxdef{prevpagerange@\@currentseries}{#1-#4}}%
4912     {\csxdef{prevpagenum@\@currentseries}{#4}}%
4913     }%
4914     {}%
4915 %

```

Now, the end of `\printendlines` macro.

```
4916 \endgroup%
4917 }%
4918
4919 %
```

`\printnpnum` A macro to print a page number in an endnote. Should not be override anymore

```
4920 \newcommand*\printnpnum[1]{\csuse{Xendbeforepagenumber@\@currentseries
4921 }#1\csuse{Xendafterpagenumber@\@currentseries}}
4922 %
```

XX Generate series of notes

In this section, X means the name of the series (A, B etc.)

`\series` `\series` creates one more new series. It is a public command, which just loops on the private command `\newseries@`.

```
4923 \newcommand{\newseries}[1]{%
4924     \def\do##1{\newseries@{##1}}%
4925     \docsvlist{#1}
4926 }
4927 %
```

`\@series` The `\series@` macro is an etoolbox list, which contains the name of all series.

```
4928 \newcommand{\@series}{}
4929 %
```

The command `\newseries@series` creates a new series of the footnote.

```
\newseries@30 \newcommand{\newseries@}[1]{
4931 %
```

XX.1 Test if series is still existing

```
4932 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
4933 {%
4934 %
```

XX.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.


```

4935 \ifdefined\newseries@par%
4936 \newseries@par{#1}%
4937 \fi%
4938 %

```

XX.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```

4939 \unless\ifnocritical@
4940 %

```

XX.3.1 Options

```

4941 \newtoggle{Xlineflag@#1}
4942 \newtoggle{Xparindent@#1}
4943 \newtoggle{Xlemmadisablefontselection@#1}
4944 \csgdef{Xwrapcontent@#1}{}%
4945 \csgdef{Xbeforeinserting@#1}{}%
4946 \csgdef{Xhangindent@#1}{Opt}%
4947 \csgdef{Xragged@#1}{}%
4948 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
4949 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
4950 \csgdef{Xcolalign@#1}{\raggedright}%
4951 \csgdef{Xnotenumfont@#1}{\normalfont}%
4952 \csgdef{Xnotefontsize@#1}{\footnotesize}%
4953 \csgdef{Xbhooknote@#1}{}%
4954 \csgdef{Xbhookgroup@#1}{}%
4955
4956 \csgdef{Xboxlinenum@#1}{Opt}%
4957 \csgdef{Xboxlinenumalign@#1}{L}%
4958
4959 \csgdef{Xboxstartlinenum@#1}{Opt}%
4960 \csgdef{Xboxendlinenum@#1}{Opt}%
4961
4962 \csgdef{Xboxsymlinenum@#1}{Opt}%
4963 \newtoggle{Xnumberonlyfirstinline@#1}%
4964 \newtoggle{Xgroupbyline@#1}%
4965 \newtoggle{Xgroupbylineseparetwolines@#1}%
4966 \newtoggle{Xnumberonlyfirstintwolines@#1}%
4967 \csgdef{Xtwolines@#1}{}%
4968 \csgdef{Xmorethantwolines@#1}{}%
4969 \csgdef{Xsublinesep@#1}{\fullstop}%
4970 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
4971 \newtoggle{Xtwolinesbutnotmore@#1}%
4972 \newtoggle{Xtwolinesonlyinsamepage@#1}%

```

```

4973 \newtoggle{Xonlypstart@#1}%
4974 \newtoggle{Xpstarteverytime@#1}%
4975 \newtoggle{Xpstart@#1}%
4976 \newtoggle{Xstanza@#1}%
4977 \csgdef{Xstanzaseparator@#1}{}%
4978 \csgdef{Xsymlinenum@#1}{}%
4979 \newtoggle{Xnonnumber@#1}%
4980 \csgdef{Xbeforenumber@#1}{0pt}%
4981 \csgdef{Xtxtbeforenumber@#1}{}%
4982 \csgdef{Xafternumber@#1}{0.5em}%
4983 \newtoggle{Xnonbreakableafternumber@#1}%
4984 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
4985 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
4986 \csgdef{Xinplaceofnumber@#1}{1em}%
4987 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
4988 \csgdef{Xbeforelemmaseparator@#1}{0em}%
4989 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
4990 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
4991 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
4992 \csgdef{Xafterrule@#1}{0pt}%
4993
4994 \csgdef{Xtxtbeforenotes@#1}{%
4995 \newtoggle{Xtxtbeforenotes@#1@typeset}%Not directly used by user,
but internal
4996
4997 \csgdef{Xmaxhnotes@#1}{0.8\vsizer%
4998 \newtoggle{Xnoteswidthliketwocolumns@#1}%
4999 \csgdef{Xparafootsep@#1}{}%
5000 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%
5001 \csgdef{Xlinrangeseparator@#1}{\endashchar}%
5002
5003 \csgdef{Xlemmafont@#1}{}%
5004 \csgdef{Xwraplemma@#1}{%
5005 \csgdef{Xwidth@#1}{\hsizer%
5006 %

```

XX.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

5007 \expandafter\newinsert\csname #1footins\endcsname%
5008 \unless\ifnoledgroup@%
5009 \expandafter\newinsert\csname mp#1footins\endcsname%
5010 \fi%
5011 %

```

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.

```

5012 \global\newcommand\parapparatus@{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand\csname #1footnote\endcsname[2] [] {%
5013 \if@edtext@secondarg%
5014 \ifledRcol%
5015 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}}%
5016 \else%
5017 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}}%
5018 \fi%
5019 \begingroup%
5020 \newcommand{\content}{##2}%
5021 \ifnumberedpar%
5022 \ifledRcol%
5023 \ifluatex%
5024 \footnotelang@lua[R]%
5025 \fi%
5026 \@ifundefined{xpg@main@language}%if polyglossia
5027 {}%
5028 {\footnotelang@poly[R]}%
5029 \footnoteoptions@{R}{##1}{true}%
5030 \xright@appenditem{%
5031 \ifbool{indtl@innote}%
5032 {\unexpanded{\let\index\nindex}}%
5033 {}%
5034 \ifbool{indtl@notenumber}%
5035 {\unexpanded{\let\index\nindex}}%There is no note
...number so
5036 {}%
5037 \noexpand\Xnote@true%
5038 \noexpand\prepare@Xprenotes{#1}%
5039 \noexpand\prepare@edindex@fornote{\l@d@nums}%
5040 \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.
5041 \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
5042 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}}%
5043 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5044 \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
5045 \noexpand\csuse{v#1footnote}{#1}%
5046 {\l@d@nums}\expandonce\@tag{\expandonce\content}}
%
5047 \noexpand\Xnote@false%
5048 \unexpanded{\advance\@edindex@fornote@m@ne}%
5049 \unexpanded{\undef\@this@crossref@start}%
5050 \unexpanded{\undef\@this@crossref@end}%

```

```

5051         \ifbool{indtl@innote}%
5052             {\unexpanded{\let\index\orig@@index}}%
5053             {}%
5054         \ifbool{indtl@notenumber}%
5055             {\unexpanded{\let\index\orig@@index}}%
5056             {}%
5057         }\to\inserts@listR
5058         \footnoteoptions@{R}{##1}{false}%
5059         \global\advance\insert@countR \@ne%
5060     \else%
5061         \ifluatex%
5062             \footnotelang@lua%
5063         \fi%
5064         \ifundefined{xpg@main@language}%if polyglossia
5065             {}%
5066             {\footnotelang@poly}%
5067         \footnoteoptions@{L}{##1}{true}%
5068         \xright@appenditem{%
5069             \ifbool{indtl@innote}%
5070                 {\unexpanded{\let\index\nindex}}%
5071                 {}%
5072             \ifbool{indtl@notenumber}%
5073                 {\unexpanded{\let\index\nindex}}%There is no note
5074             ...number so
5075             {}%
5076             \noexpand\Xnote@true%
5077             \noexpand\prepare@Xprenotes{##1}%
5078             \noexpand\prepare@edindex@fornote{\l@d@nums}%
5079             \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.
5080             \ifl@dpairing%
5081                 \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
5082             \fi%
5083             \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5084             \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5085             \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
5086             \noexpand\csuse{v#1footnote}%
5087             {##1}%
5088             {\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
5089             \unexpanded{\undef\@this@crossref@start}%
5090             \unexpanded{\undef\@this@crossref@end}%
5091             \noexpand\Xnote@false%
5092             \unexpanded{\advance\@edindex@fornote@m@ne}%
5093             \ifbool{indtl@innote}%

```

Create counter used to determine on which page the previous note was called.

We need to be able to modify `reledmac`'s footnote macros and restore their

XX.3.4 Set standard display

End of for critical footnotes.

XX.4 For familiar footnotes

```
5121 \unless\ifnofamiliar@
5122 %
```

XX.4.1 Options

```

5123 \newtoggle{parindentX@#1}
5124 \csgdef{wrapcontentX@#1}{}%
5125 \csgdef{hangindentX@#1}{Opt}%
5126 \csgdef{beforeinsertingX@#1}{}%
5127 \csgdef{raggedX@#1}{}%
5128 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
5129 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
5130 \csgdef{colalignX@#1}{\raggedright}%
5131 \csgdef{notenumfontX@#1}{\normalfont}%
5132 \csgdef{notefontsizeX@#1}{\footnotesize}%
5133 \csgdef{bhooknoteX@#1}{}%
5134 \csgdef{bhookgroupX@#1}{}%
5135 \csgdef{afterruleX@#1}{Opt}
5136 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
5137 \csgdef{maxhnotesX@#1}{0.8\vsize}%
5138 \newtoggle{noteswidthliketwocolumnsX@#1}%
5139 \csgdef{parafootsepX@#1}{}%
5140 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
5141 \csgdef{widthX@#1}{\hsize}%
5142 % End of for familiar footnotes.
5143 % \subsubsection{Create inserts, needed to add notes in foot}
5144 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
5145 % \begin{macrocode}
5146 \expandafter\newinsert\csname footins#1\endcsname%
5147 \unless\ifnoledgroup%
5148 \expandafter\newinsert\csname mpfootins#1\endcsname%
5149 \fi%
5150 %

```

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.

```

5151 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
5152 \begingroup%
5153 \prepare@prenotesX{#1}%
5154 \newcommand{\content}{##1}%
5155 %
5156 %

```

If we use the `\csquotes` package, we reset quotation level.

```

5157 \ifdefined\csq@qllevel%
5158 \csq@qllevel=0\relax%
5159 \fi%
5160 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.1.2 p. 50).

```

5161         \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
5162         \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or
bool{l@dprintingcolumns}}{%
5163             \ifcsdef{footnote#1reading\the\csname footnote#1@reading\
endcsname=typeset}%
5164                 {\setcounter{footnote#1}{\csuse{footnote#1reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
5165                 {\setcounter{footnote#1}{\the\csname footnote#1@reading
\endcsname}}}%
5166             }{%
5167                 \stepcounter{footnote#1}%
5168             }%
5169 %

```

We also have to check consistency with `\onlysideX` setting.

```

5170         \ifledRcol%
5171             \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}{%
5172             \else%
5173                 \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}{%
5174             \fi%
5175 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

5176         \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5177         \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
5178             {\csuse{@footnotemark#1}}%
5179             {}%
5180         \ifluatex%
5181             \xdef\footnote@luatextextdir{\the\textdir}%
5182             \xdef\footnote@luatexpardir{\the\pardir}%
5183         \fi%
5184         \if@ledgroup%
5185             \led@set@index@fornote{#1}%
5186         \fi%
5187         \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
5188         \ifbool{indtl@innote}%
5189             {\let\index\orig@@index}%
5190             {}%
5191         \ifbool{indtl@notenumber}%
5192             {\let\index\orig@@index}%
5193             {}%
5194         \endgroup%
5195     }
5196 %

```

Then define the counters. The \LaTeX counter `footnoteX` is the only one manipulated by the user. This is the one which is printed. The \TeX counter `\footnoteX@reading` is increased at each footnote. It is used for hyperlinks, for using `hyperlink` package, and for getting the correct footnote number when using parallel typesetting (V.1.2 p. 50).

```

5197     \newcounter{footnote#1}
5198     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
5199     \expandafter\newcount\csname footnote#1@reading\endcsname%
5200 %

```

Create counter used to determine on which page the previous note was called.

```

5201     \expandafter\newcount\csname prevpage#1@num\endcsname%
5202     \expandafter\newcount\csname prevpage#1@numR\endcsname%
5203 %

```

Add `\let\footnoteX@gobble` to `\no@expands`.

```

5204     \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
5205 %

```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to classical `\footnotemark` and `\footnotetext`.

```

5206     \expandafter\newcommand\csname footnote#1mark\endcsname{%
5207     \begingroup%
5208     \prepare@prenotesX{#1}%
5209     \stepcounter{footnote#1}%
5210     \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5211     \csuse{@footnotemark#1}%
5212     \m@mmf@prepare%
5213     \endgroup%
5214 }%
5215     \expandafter\newcommand\csname footnote#1text\endcsname[1]{%
5216     \begingroup%
5217     \csuse{vfootnote#1}{#1}{\expandonce{##1}}%
5218     \endgroup%
5219 }%
5220 %

```

Do not forget to initialize the series.

```

5221     \arrangementX@normal{#1}%
5222     \fi
5223 %

```

XX.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.


```

5224 \unless\ifnoend@
5225 %

```

XX.5.1 The auxiliary file

Endnotes of all varieties are saved up in a file, one by series, typically named $\langle jobname \rangle$.Xend. $\backslash l@d@Xend$ is the output stream number for this file, and $\backslash ifl@dend@X$ is a flag that is true when the file is open.

```

5226 \expandafter\newwrite\csname l@d@#1end\endcsname%
5227 \expandafter\newif\csname ifl@dend@#1\endcsname%
5228 %

```

XX.5.2 The main macro

The $\backslash Xendnote$ macro functions to write one endnote to the .Xend file. We change $\backslash newlinechar$ 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.

```

5229 \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,
5230 usedefault]{%
5231 \bgroup%
5232 \newlinechar='40%
5233 \global\@noneed@Footnotetrue%
5234 \newcommand{\content}{##2}%
5235 \stepcounter{labidx}%
5236 \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
5237 \unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\
expandafter{\sw@inthisedtext}}\@percentchar\space%Explicit space, to add a
linebreak in the output file
5238 \expandafter\string\csname #1end\endcsname%
5239 {\ifnumberedpar@l@d@nums\fi}%
5240 {\ifnumberedpar@\expandonce\@tag\fi}%
5241 {\expandonce\content}%
5242 {\#1}%
5243 {\unexpanded{##1}}%
5244 {\ifledRcol R\else L\fi}%
5245 {\theedtext}%
5246 \@percentchar%
5247 }%
5248 \egroup%
5249 \ignorespaces%
5250 }%
5251 %

```

XX.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

5252 \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
5253 \toendnotes[#1]{##1}%
5254 }%
5255
5256 \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
5257 endcsname*[1]{%
5258 \toendnotes*[#1]{##1}%
5259 }%
5260
5261 %

```

XX.5.4 Internal commands

`\Xendnote` commands called `\Xend` commands on to the endnote file; these are analogous to the various `footfmt` commands above, and they take the same arguments. When we process this file, we want to pick out the notes of one series and ignore all the rest. To do that, we equate the end command for the series we want to `\endprint`, and leave the rest equated to `\@gobbleseven`, which just skips over its seven arguments.

```

5262 \global\cslet{#1end}{\@gobbleseven}
5263
5264 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

5265 \global\expandafter\newcount\csname #1end@bysection\endcsname%
5266 %

```

XX.5.5 The options

```

5267 \csgdef{Xendwraplemma@#1}{%
5268 \csgdef{Xendwrapcontent@#1}{}%
5269 \csgdef{Xendtwolines@#1}{}%
5270 \csgdef{Xendmorethantwolines@#1}{}%
5271 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
5272 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
5273 \newtoggle{Xendlemmadisablefontselection@#1}{}%
5274 \csgdef{Xendnotenumfont@#1}{\normalfont}%
5275 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
5276 \csgdef{Xendbhooknote@#1}{}%
5277
5278 \csgdef{Xendsublinesep@#1}{\fullstop}%
5279
5280 \csgdef{Xendbeforenumber@#1}{\Opt}

```

```

5281 \csgdef{Xendafternumber@#1}{0.5em}
5282
5283 \csgdef{Xendboxlinenum@#1}{Opt}%
5284 \csgdef{Xendboxlinenumalign@#1}{L}%
5285
5286 \csgdef{Xendboxstartlinenum@#1}{Opt}%
5287 \csgdef{Xendboxendlinenum@#1}{Opt}%
5288
5289 \csgdef{Xendlemmaseparator@#1}{}%
5290 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
5291 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
5292 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
5293
5294 \newtoggle{Xendparagraph@#1}%
5295 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
5296 \csgdef{Xendsep@#1}{}%
5297
5298 \csgdef{Xendinplaceofnumber@#1}{Opt}%
5299 \newtoggle{Xendnonumber@#1}%
5300
5301 \csgdef{Xendhangindent@#1}{Opt}%
5302 \newtoggle{Xendnumberonlyfirstinline@#1}%
5303 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
5304
5305 \csgdef{Xendbeforesymmlinenumber@#1}{\csuse{Xendbeforenumber@#1}}%
5306 \csgdef{Xendaftersymmlinenumber@#1}{\csuse{Xendafternumber@#1}}%
5307 \csgdef{Xendsymmlinenumber@#1}{}%
5308 \csgdef{Xendboxsymmlinenumber@#1}{Opt}%
5309
5310 \csgdef{Xendbhooklinenumber@#1}{}%
5311 \csgdef{Xendehooklinenumber@#1}{}%
5312 \csgdef{Xendbhookinplaceofnumber@#1}{}%
5313 \csgdef{Xendehookinplaceofnumber@#1}{}%
5314
5315 \csgdef{Xendlinangeseparator@#1}{\endashchar}%
5316
5317 \csgdef{Xendbeforepagenumber@#1}{p.}%
5318 \csgdef{Xendafterpagenumber@#1}{) }%
5319 \csgdef{Xendlineprefixsingle@#1}{}%
5320 \csgdef{Xendlineprefixmore@#1}{}%
5321
5322 \newtoggle{Xendlineflag@#1}
5323
5324 \csgdef{Xendlemmafont@#1}{}%
5325
5326 \newtoggle{Xendpagenumberonlyfirst@#1}%
5327 \newtoggle{Xendpagenumberonlyfirsttifsingletwo@#1}%
5328 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
5329 \csgdef{Xendsympagenumber@#1}{}%
5330 \csgdef{Xendinplaceofpagenumber@#1}{Opt}%

```

```

5331 %
5332 %
End of endnotes declaration
5333 \fi%
5334 %
Dump series in \@series
5335 \listxadd{\@series}{#1}
5336 }
5337 }% End of \newseries
5338 %

```

XX.6 Init standards series (A,B,C,D,E)

```

5339 \expandafter\newseries\expandafter{\default@series}
5340 %

```

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.

```

5341 \newcommand{\seriesatbegin}[1]{%
5342   \StrDel{\@series}{#1}[\@series]%
5343   \edef\@new{}%
5344   \listead{\@new}{#1}%
5345   \listead{\@new}{\@series}%
5346   \xdef\@series{\@new}%
5347 }
5348 %

```

\seriesatend And `\seriesatend` moves the series to the end of the list.

```

5349 \newcommand{\seriesatend}[1]{%
5350   \StrDel{\@series}{#1}[\@series]%
5351   \edef\@new{}%
5352   \listead{\@new}{\@series}%
5353   \listead{\@new}{#1}%
5354   \xdef\@series{\@new}%
5355 }
5356 %

```

XXI.2 Test series order

\ifseriesbefore `\ifseriesbefore{⟨seriesA⟩}{⟨seriesB⟩}{⟨true⟩}{⟨false⟩}` expands `⟨true⟩` if `⟨seriesA⟩` is printed before `⟨seriesB⟩`, expands `⟨false⟩` otherwise.

```

5357 \newcommand{\ifseriesbefore}[4]{%
5358   \StrPosition{\@series}{#1}[\@first]%
5359   \StrPosition{\@series}{#2}[\@second]%
5360   \ifnumgreater{\@second}{\@first}{#3}{#4}%
5361 }
5362 %

```

XXI.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries@
5363 \newcommand{\@getfirstseries}{%
5364   \ifdefempty{\@series}%
5365   {\xdef\@firstseries{}}%
5366   {\StrChar{\@series}{1}[\@firstseries]}%
5367 }%
5368 %

```

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`.

```

5369 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
5370   \def\do##1{%
5371     \global\settoggle{#2@##1}{#3}%
5372     \ifstrequal{#4}{critical}{
5373       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
5374     }{}
5375     \ifstrequal{#4}{familiar}{
5376       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
5377     }{}
5378   }%
5379   \ifstreempty{#1}{%
5380     \dolistloop{\@series}%
5381     \ifstreempty{#5}{}%
5382     \docsvlist{#5}%
5383   }
5384   }%
5385   {%
5386     \docsvlist{#1}%
5387   }%
5388 }
5389 %

```

`\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`.

```

5390 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
5391   \def\do##1{
5392     \csgdef{#2@##1}{#3}
5393     \ifstrequal{#4}{critical}{%
5394       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
5395     }{}
5396     \ifstrequal{#4}{familiar}{%
5397       \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
5398     }{}%
5399   }%
5400   \ifstrepty{#1}{%
5401     \dolistloop{\@series}%
5402     \ifstrepty{#5}{%
5403       \docsvlist{#5}
5404     }
5405   }%
5406   {%
5407     \docsvlist{#1}%
5408   }%
5409 }%
5410 %

```

XXI.3.3 Tools to generate options commands

\newhookcommand@series \newhookcommand@series\command names is a generic command to add new commands for hooks, like \Xhsizetwocol. The first argument is the name of the hook, the second a comma-separated list of pseudo-series where the hook can be used, like appref in the case of \Xtwolines. The second argument is also used to create commands named \<hookname><pseudoseris>, like \Xtwolinesappref.

```

5411 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
5412   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5413     \setcommand@series{##1}{#1}{##2}[] [#2]%
5414   }%
5415   \ifstrepty{#2}{%
5416     \def\do##1{%
5417       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
5418       [1]{%
5419         \csuse{#1}[##1]{####1}%
5420       }%
5421     }%
5422     \docsvlist{#2}%
5423   }%
5424 }%

```

\newhooktoggle@series \newhooktoggle@series\command names is a generic command to add new commands for a new toggle hook, like \Xnumberonlyfirstinline. The second argu-

ment is also used to create commands named `\<hookname><pseudoserious>`, like `\Xtwolinesbutnotmoreappref`.

```

5425 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
5426   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
5427     \settoggle@series{##1}{#1}{##2}[][#2]%
5428   }%
5429   \ifstrempy{#2}{%
5430     \def\do##1{%
5431       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
5432         \csuse{#1}[##1]%
5433       }%
5434     }%
5435     \docsvlist{#2}%
5436   }%
5437 }
5438 %

```

`\newhooktoggle@series@reload` `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement, depending of type of notes

```

5439 \newcommand{\newhooktoggle@series@reload}[2]{%
5440   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
5441     \settoggle@series{##1}{#1}{##2}[#2]%
5442   }%
5443 }%
5444 %

```

`\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.

```

5445 \newcommand{\newhookcommand@series@reload}[2]{%
5446   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5447     \setcommand@series{##1}{#1}{##2}[#2]%
5448   }%
5449 }
5450 %

```

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.

```

5451 \unless\ifnocritical@
5452   \newhookcommand@series{Xwrapcontent}%
5453   \newhookcommand@series{Xbeforeinserting}%

```



```

5454 \newhookcommand@series{Xlemmafont}%
5455 \newhookcommand@series{Xwraplemma}%
5456 \newhooktoggle@series{Xparindent}
5457 \newhookcommand@series{Xhangindent}
5458 \newhookcommand@series{Xragged}
5459 \newhookcommand@series{Xhsizetwocol}
5460 \newhookcommand@series{Xhsizethreecol}
5461 \newhookcommand@series{Xcolalign}%
5462 \newhookcommand@series{Xnotenumfont}
5463 \newhookcommand@series{Xbhooknote}
5464 \newhookcommand@series@reload{Xbhookgroup}{critical}
5465 \newhookcommand@series{Xboxsymlinenum}%
5466 \newhookcommand@series{Xsymlinenum}
5467 \newhookcommand@series{Xbeforenumber}
5468 \newhookcommand@series{Xtxtbeforenumber}
5469 \newhookcommand@series{Xafternumber}
5470 \newhookcommand@series{Xbeforesymlinenum}
5471 \newhookcommand@series{Xaftersymlinenum}
5472 \newhookcommand@series{Xinplaceofnumber}
5473 \newhookcommand@series{Xlemmaseparator}
5474 \newhookcommand@series{Xbeforelemmaseparator}
5475 \newhookcommand@series{Xafterlemmaseparator}
5476 \newhookcommand@series{Xinplaceoflemmaseparator}
5477 \newhookcommand@series{Xtxtbeforenotes}
5478 \newhookcommand@series@reload{Xafterrule}{critical}
5479 \newhooktoggle@series{Xnumberonlyfirstinline}
5480 \newhooktoggle@series{Xnumberonlyfirstintwolines}
5481 \newhooktoggle@series{Xgroupbyline}%
5482 \newhooktoggle@series{Xgroupbylineseparetwolines}%
5483 \newhooktoggle@series{Xnonumber}
5484 \newhooktoggle@series{Xpstart}
5485 \newhooktoggle@series{Xpstarteverytime}%
5486
5487 \newhooktoggle@series{Xstanza}%
5488 \newhookcommand@series{Xstanzaseparator}%
5489
5490 \newhooktoggle@series{Xonlypstart}
5491 \newhooktoggle@series{Xnonbreakableafternumber}
5492 \newhooktoggle@series{Xlemmadisablefontselection}
5493 \newhookcommand@series@reload{Xmaxhnotes}{critical}
5494 \newhookcommand@series@reload{Xbeforenotes}{critical}
5495 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
5496 \newhookcommand@series@reload{Xnotefontsize}{critical}
5497
5498 \newhookcommand@series{Xboxlinenum}%
5499 \newhookcommand@series{Xboxlinenumalign}%
5500
5501 \newhookcommand@series{Xboxstartlinenum}%
5502 \newhookcommand@series{Xboxendlinenum}%
5503

```

```

5504 \newhookcommand@series{Xafternote}%
5505 \newhookcommand@series{Xparafootsep}
5506
5507 \newhookcommand@series@reload{Xwidth}{critical}%
5508
5509 \ifundef{\Xhsize}%
5510 {%
5511   \newcommandx{\Xhsize}[2][1,usedefault]{%
5512     \led@warning@Xhsize@deprecated%
5513     \Xwidth[#1]{#2}%
5514   }%
5515 }%
5516 {}%
5517 \fi
5518 \newhooktoggle@series{Xlineflag}[appref,SEref]
5519 \newhookcommand@series{Xtwolines}[appref,SEref]
5520 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
5521 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
5522 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
5523 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
5524 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
5525 \newhookcommand@series{Xlinerrangeseparator}[appref,SEref]
5526 %

```

XXI.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

5527 \unless\ifnofamiliar@
5528   \newhookcommand@series{wrapcontentX}%
5529   \newhookcommand@series{beforeinsertingX}%
5530   \newhooktoggle@series{parindentX}
5531   \newhookcommand@series{hangindentX}
5532   \newhookcommand@series{raggedX}
5533   \newhookcommand@series{hsizetwocolX}
5534   \newhookcommand@series{hsizethreecolX}
5535   \newhookcommand@series{colalignX}%
5536   \newhookcommand@series{notenumfontX}
5537   \newhookcommand@series{bhooknoteX}
5538   \newhookcommand@series@reload{bhookgroupX}{familiar}
5539   \newhookcommand@series@reload{beforenotesX}{familiar}
5540   \newhookcommand@series@reload{maxhnotesX}{familiar}
5541   \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
5542   \newhookcommand@series@reload{afterruleX}{familiar}
5543   \newhookcommand@series@reload{notefontsizeX}{familiar}
5544   \newhookcommand@series{afternoteX}
5545   \newhookcommand@series{parafootsepX}
5546   \newhookcommand@series@reload{widthX}{familiar}%
5547   \ifundef{\hsizeX}%

```

```

5548 {%
5549 \newcommandx{\hsizeX}[2][1,usedefault]{%
5550 \led@warning@hsizeX@deprecated%
5551 \widthX[#1]{#2}%
5552 }%
5553 }%
5554 {}%
5555 \fi
5556 %

```

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.

```

5557 \unless\ifnoend@
5558 \newhookcommand@series{Xendwraplemma}
5559 \newhookcommand@series{Xendwrapcontent}
5560 \newhookcommand@series{Xendnotenumfont}
5561 \newhookcommand@series{Xendlemmafont}%
5562 \newhookcommand@series{Xendbhooknote}
5563
5564 \newhookcommand@series{Xendboxlinenum}%
5565 \newhookcommand@series{Xendboxlinenumalign}%
5566
5567 \newhookcommand@series{Xendboxstartlinenum}%
5568 \newhookcommand@series{Xendboxendlinenum}%
5569
5570 \newhookcommand@series{Xendnotefontsize}
5571 \newhooktoggle@series{Xendlemmadisablefontselection}
5572 \newhookcommand@series{Xendlemmaseparator}
5573 \newhookcommand@series{Xendbeforelemmaseparator}
5574 \newhookcommand@series{Xendafterlemmaseparator}
5575 \newhookcommand@series{Xendinplaceoflemmaseparator}
5576
5577 \newhookcommand@series{Xendbeforenumber}%
5578 \newhookcommand@series{Xendafternumber}%
5579
5580 \newhooktoggle@series{Xendparagraph}
5581 \newhookcommand@series{Xendafternote}
5582 \newhookcommand@series{Xendsep}
5583
5584 \newhookcommand@series{Xendinplaceofnumber}%
5585 \newhooktoggle@series{Xendnonumber}%
5586
5587 \newhooktoggle@series{Xendnumberonlyfirstinline}%
5588 \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
5589
5590 \newhookcommand@series{Xendsymmlinenum}%
5591 \newhookcommand@series{Xendbeforesymmlinenum}%

```

```

5592 \newhookcommand@series{Xendaftersymmlinenumber}%
5593 \newhookcommand@series{Xendboxsymmlinenumber}%
5594
5595 \newhookcommand@series{Xendbhooklinenumber}%
5596 \newhookcommand@series{Xendahooklinenumber}%
5597 \newhookcommand@series{Xendbhookinplaceofnumber}%
5598 \newhookcommand@series{Xendahookinplaceofnumber}%
5599
5600 \newhookcommand@series{Xendhangindent}%
5601
5602 \newhooktoggle@series{Xendpagenumberonlyfirst}%
5603 \newhooktoggle@series{Xendpagenumberonlyfirstifsingle}%
5604 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
5605 \newhookcommand@series{Xendsympagenum}%
5606 \newhookcommand@series{Xendinplaceofpagenumber}%
5607
5608 \fi
5609 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
5610 \newhookcommand@series{Xendtwwlines}[apprefwithpage,SErefwithpage]
5611 \newhookcommand@series{Xendmorethantwwlines}[apprefwithpage,SErefwithpage]
5612 \newhooktoggle@series{Xendtwwlinesbutnotmore}[apprefwithpage,SErefwithpage]
5613 \newhooktoggle@series{Xendtwwlinesonlyinsamepage}[apprefwithpage,
5614 SErefwithpage]
5615 \newhookcommand@series{Xendlinrangeseparator}[apprefwithpage,SErefwithpage]
5616 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
5617 SErefonlypage]
5618 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
5619 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
5620 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
5621 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
5622
5623 %

```

XXI.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

5622 \newcommand{\newhooktoggle@specific}[1]{%
5623   \newtoggle{#1}%
5624   \define@key{mac}{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
5625   When enabling footnote option
5626   \define@key{mac}{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false}}
5627 }
5628 %

```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

5628 \newcommand{\newhookarg@specific}[1]{%
5629   \define@key[mac]{truefootnoteoption}{#1}{\global\def\linrangesep@{#1}}%
    When enabling footnote option
5630   \define@key[mac]{falsefootnoteoption}{#1}{\global\undef\linrangesep@}%
    When
5631 }
5632 %

```

And now, we define some hooks specific to a note.

```

5633 \newhooktoggle@specific{fulllines}%
5634 \newhooktoggle@specific{nonum}
5635 \newhooktoggle@specific{nosep}
5636 \newhookarg@specific{linrangesep}
5637 %

```

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.

```

5638 %

```

\nomk@ \nomk@ toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```

5639 \newtoggle{nomk@}%
5640 %

```

XXI.5 Alias

\Xnolemmaseparator \Xnolemmaseparator[*<series>*] is just an alias for \Xlemmaseparator[*<series>*]{ }.

```

5641 \newcommandx*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]{}}
5642 %

```

XXII Output routine

Now we begin the output routine and associated things.

XXII.1 Extra footnotes output

With luck we might only have to change `\@makecol` and `\@reinserts` of the \TeX 's kernel. Since `reledmac`, we use `etoolbox`'s patching commands instead of overriding. It should provides better compatibility with other package which modify these commands

\doxtrafeet \doxtrafeet is the code extending \@makecol to cater for the extra reledmac feet. We have two categories of extra footnotes. By default, we order the footnote inserts so that the regular footnotes of L^AT_EX are first, then familiar familiar footnotes and finally the critical footnotes.

```

5643 \newcommand*\loddoxtrafeet}{%
5644   \IfStrEq{familiar-critical}{\@fnpos}
5645     {\do@feetX\do@Xfeet}%
5646     {%
5647       \IfStrEq{critical-familiar}{\@fnpos}%
5648         {\do@Xfeet\do@feetX}%
5649         {%
5650           \setbox\@outputbox \vbox{%
5651             \unvbox\@outputbox%
5652             \do@feet@custom@order}{\@fnpos}%
5653           }%
5654         }%
5655       }%
5656     }%
5657   %
5658 }
```

\do@feet@custom@order \do@feet@custom@order is called when \@fnpos is neither ‘familiar-critical’, nor ‘critical-familiar’, that is, when the order is more complex. In this case, people must define the order for all footnote series. If they don’t, L^AT_EX could perform an infinite run.

```

5659 \newcommand{\do@feet@custom@order}[2]{%
5660   \def\do##1{%
5661     \edef\@notesseries{\@firstoftwo##1}%
5662     \edef\@notetype{\@secondoftwo##1}%
5663     \ifdefstring{\@notetype}{critical}%
5664       {\csuse{#1append@Xnotes}{\@notesseries}}%
5665       {\ifdefstring{\@notetype}{familiar}%
5666         {\csuse{#1append@notesX}{\@notesseries}}%
5667         {}%
5668       }%
5669     }%
5670   \expandafter\docsvlist\expandafter{#2}%
5671 }%
5672 %
```

\do@Xfeet \do@Xfeet is the code extending \@makecol to cater to the extra critical feet.

```

5673 \newcommand*\do@Xfeet}{%
5674   \setbox\@outputbox \vbox{%
5675     \unvbox\@outputbox
5676     \@opXfeet}}
5677 %
```

`\@opXfeet` The extra critical feet to be added to the output. . A macro which appends critical notes to the output's routine, also adding vertical space before notes

`\append@Xnotes`

`\print@Xnotes`

```

5678 \newcommand{\append@Xnotes}[1]{%
5679   \ifvoid\csuse{#1footins}\else%
5680     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
5681     \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
5682     \print@Xnotes{#1}%
5683   \fi%
5684 }%
5685 %

```

The normal way to add one series, `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

5686 \newcommand\print@Xnotes[1]{%
5687   \xdef\@currentseries{#1}%
5688   \csuse{#1footstart}{#1}%
5689   \csuse{#1footgroup}{#1}%
5690 }%
5691 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

5692 \newcommand*\@opXfeet{%
5693   \unless\ifnocritical@%
5694     \gdef\firstXseries@{%
5695       \def\do##1{%
5696         \append@Xnotes{##1}%
5697       }%
5698       \dolistloop{\@series}%
5699     \fi%
5700 }%
5701 %

```

`\l@ddodoreinextrafeet` `\l@ddodoreinextrafeet` is the code for catering for the extra footnotes within `\@reinserts`. We use the same category and ordering as in `\l@ddoxtrafeet`.

```

5702 \newcommand*\l@ddodoreinextrafeet{%
5703   \IfStrEq{familiar-critical}{\@fnpos}
5704     {\@doreinfeetX\X@doreinfeet}%
5705   {%
5706     \IfStrEq{critical-familiar}{\@fnpos}%
5707       {\X@doreinfeet\@doreinfeetX}%
5708       {\@doreinfeetX\X@doreinfeet}%
5709     }%
5710 }
5711
5712 %

```

`\X@doreinfeet` `\X@doreinfeet` is the code for catering for the extra critical footnotes within `\@reinserts`.

```

5713 \newcommand*{\X@doreinfeet}{%
5714   \unless\ifnocritical@%
5715     \def\do##1{%
5716       \ifvoid\csuse{##1footins}\else%
5717         \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
5718       \fi}%
5719     \dolistloop{\@series}
5720   \fi%
5721 }
5722
5723 %

```

`\print@notesX` We have to add all the new kinds of familiar footnotes to the output routine. A macro
`\append@notesX` which appends the familiar footnotes of one series onto the output routine, also adding
`\do@feetX` vertical skip before notes.

```

\@doreinfeetX
5724 \newcommand{\append@notesX}[1]{%
5725   \ifvoid\csuse{footins#1}\else%
5726     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
5727     \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
5728     \print@notesX{#1}%
5729   \fi%
5730 }%
5731 %

```

The normal way to print one series of notes. `\print@Xnotes` is replaced by `reledpar` when using `\Pages`.

```

5732 \newcommand\print@notesX[1]{%
5733   \xdef\@currentseries{#1}%
5734   \csuse{footstart#1}{#1}%
5735   \csuse{footgroup#1}{#1}%
5736 }%
5737 %

```

We print all the series of notes by looping on them. We check before printing them that they are not voided.

```

5738 \newcommand*{\do@feetX}{%
5739   \unless\ifnofamiliar@%
5740     \gdef\firstseriesX@{}%
5741     \setbox\@outputbox \vbox{%
5742       \unvbox\@outputbox%
5743       \def\do##1{%
5744         \append@notesX{##1}%
5745       }%
5746       \dolistloop{\@series}}%
5747   \fi%
5748 }%
5749
5750 \newcommand{\@doreinfeetX}{%

```



```

5751 \unless\ifnofamiliar@%
5752 \def\do##1{%
5753 \ifvoid\csuse{footins##1}\else
5754 \insert%
5755 \csuse{footins##1}
5756 {\unvbox\csuse{footins##1}}}%
5757 \fi%
5758 }%
5759 \dolistloop{\@series}%
5760 \fi%
5761 }%
5762
5763 %

```

XXII.2 Patching standard output's commands

The memoir class does not use the ‘standard’ versions of `\@makecol` and `\@reinserts`, due to its sidebar insert. We had better add that code if memoir is used. (It can be awkward dealing with `\if` code within `\if` code, so don't use `\ifl@dmemoir` here.)

```

5764 \@ifclassloaded{memoir}{%
5765 %

```

memoir is loaded so we use memoir's built in hooks.

```

5766 \g@addto@macro{\m@mddoextrafeet}{\l@dddoextrafeet}%
5767 \g@addto@macro{\m@mddodoreinextrafeet}{\l@ddodoreinextrafeet}%
5768 }{%
5769 %

```

memoir has not been loaded, so patch `\@makecol` and `\@reinserts`. If the fancyhdr package < version 3.8 has been loaded, we patch the `\latex@makecol` command, because this package redefines the standard `\@makecol` in the preamble, to call `\latex@makecol` which have been `\let` to `\@makecol`. If this package is not loaded, we directly patch `\@makecol`. If the fancyhdr package \geq version 3.8, we also directly patch `\@makecol`, because fancyhdr does its own patch `\AtBeginDocument`.

```

5770 \ifboolexpr{%
5771 test{\@ifpackageloaded{fancyhdr}}%
5772 and test {\ifdef{\latex@makecol}}%
5773 }{%
5774 \patchcmd%
5775 {\latex@makecol}%
5776 {\xdef\@freelist{\@freelist\@midlist}}%
5777 {\xdef\@freelist{\@freelist\@midlist}\l@dddoextrafeet}%
5778 }%
5779 {\led@error@fail@patch@\@makecol}%
5780 }{%
5781 \patchcmd%
5782 {\@makecol}%

```

```

5783     {\xdef\@freelist{\@freelist\@midlist}}}%
5784     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5785     }%
5786     {\led@error@fail@patch@@makecol}%
5787   }%
5788
5789   \patchcmd%
5790     {\@reinserts}%
5791     {\ifvbox}%
5792     {\l@ddodoreinextrafeet\ifvbox}%
5793     {}%
5794     {\led@error@fail@patch@@reinserts}%
5795   }
5796
5797   %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

5798   \newif\if@led@nofoot
5799
5800   %

```

```

5801   \@ifclassloaded{memoir}{%
5802   %

```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet%03 \g@addto@macro{\@mem@extranofeet}{%%
5804   \def\do#1{%
5805     \unless\ifnocritical@%
5806       \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
5807     \fi%
5808     \unless\ifnofamiliar@%
5809       \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
5810     \fi%
5811   }
5812   \dolistloop{\@series}%
5813   }%
5814 }{%
5815 %

```

As memoir is not loaded we have patch `\@doclearpage`.

```

\@led@testifnofoot%16 \newcommand*{\@led@testifnofoot}{%
\@doclearpage%17   \@led@nofoottrue%
5818   \ifvoid\footins\else%
5819   \@led@nofootfalse%
5820   \fi%

```

```

5821 \def\do##1{%
5822   \unless\ifnocritical@%
5823     \ifvoid\csuse{##1footins}\else%
5824       \@led@nofootfalse%
5825     \fi%
5826   \fi%
5827   \unless\ifnofamiliar@%
5828     \ifvoid\csuse{footins##1}\else%
5829       \@led@nofootfalse%
5830     \fi%
5831   \fi%
5832 }%
5833 \dolistloop{\@series}%
5834 }%
5835
5836 \pretocmd%
5837   {\@docclearpage}%
5838   {\@led@testifnofoot}%
5839   {}%
5840   {\led@error@fail@patch@@docclearpage}%
5841
5842 \patchcmd%
5843   {\@docclearpage}%
5844   {\ifvoid\footins}%
5845   {\if@led@nofoot}%
5846   {}%
5847   {\led@error@fail@patch@@docclearpage}%
5848
5849 }
5850
5851 %

```

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.

```

5852 \list@create{\labelref@list}

```

```
5853 %
```

\zz@@@ A convenience macro to zero three labeling counters in one go.

```
5854 \newcommand*{\zz@@@}{000|000|000}% Set three counters to zero in one go
5855
5856 %
```

\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.³³

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 Billett³⁴ found that the original code could be off by several pages. This version, hopefully cures that, and also allows for non-arabic page numbering.

```
5857 \newcommand*{\edlabel}[1]{%
5858   \ifl@dpairing\ifautopar%
5859     \strut%
5860     \fi\fi%
5861     \@bsphack%
5862     \ifboolexpr{bool{ledRcol} or bool{ledRcol}}{%
5863       \ifXnote%
5864         \protected@write\@auxout{%
5865           {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstartR|{#1}}}%
5866         \ifdef{\hypertarget}%
5867           {\Hy@raisedlink{\hypertarget{#1}{}}}%
5868           {}%
5869         \else%
5870           \write\linenum@outR{\string\@lab}%
5871           \ifx\labelref@listR\empty%
5872             \xdef\label@refs{\zz@@@}%
5873           \else%
5874             \gl@p\labelref@listR\to\label@refs%
5875           \fi%
5876           \ifvmode%
5877             \advancelabel@refs%
5878           \fi%
5879   %
```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

³³The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

³⁴(jdb43@cam.ac.uk) via the `ctt` thread ‘`ledmac cross referencing`’, 25 August 2003.

```

5880 \protected@write\@auxout{}\%
5881 {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}}%
5882 \ifdef\hypertarget{%
5883 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5884 {}}%
5885 \fi%
5886 }{%
5887 \ifXnote@%
5888 \ifl@dpairing%pstart or pstartL?
5889 \protected@write\@auxout{}\%
5890 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\the\c@pstartL|{#1}}}%
5891 \ifdef\hypertarget{%
5892 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5893 {}}%
5894 \else%
5895 \protected@write\@auxout{}\%
5896 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\the\c@pstart|{#1}}}%
5897 \ifdef\hypertarget{%
5898 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5899 {}}%
5900 \fi%
5901 \else%
5902 \write\linenum@out{\string\@lab}%
5903 \ifx\labelref@list\empty%
5904 \xdef\label@refs{\zz@@@}%
5905 \else%
5906 \gl@p\labelref@list\to\label@refs%
5907 \fi%
5908 \ifvmode%
5909 \advancelabel@refs%
5910 \fi%
5911 \ifl@dpairing%Pstart or PstartL?
5912 \protected@write\@auxout{}\%
5913 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}}%
5914 \ifdef\hypertarget{%
5915 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5916 {}}%
5917 \else%
5918 \protected@write\@auxout{}\%
5919 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}}%
5920 \ifdef\hypertarget{%
5921 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5922 {}}%
5923 \fi%
5924 \fi%

```

```

5925 }%
5926 \@esphack}%
5927
5928 %

```

`\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.

```

5929 \newcounter{line}%
5930 \newcounter{subline}%
5931 \newcounter{absline}%
5932 \newcommand{\advancelabel@refs}{%
5933   \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
5934   \stepcounter{line}%
5935   \setcounter{absline}{\expandafter\labelrefsparseabsline\label@refs}%
5936   \stepcounter{absline}%
5937   \ifsublines%
5938     \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
5939   %
5940   \stepcounter{subline}{1}%
5941   \def\label@refs{\theline|\thesubline|\theabsline}%
5942   \else%
5943     \def\label@refs{\theline|0|\theabsline}%
5944   \fi%
5945 }
5946 \def\labelrefsparseline#1|#2|#3{#1}%
5947 \def\labelrefsparsesubline#1|#2|#3{#2}%
5948 \def\labelrefsparseabsline#1|#2|#3{#3}%
5949 %

```

`\l@dmake@labels` The `\l@dmake@labels` macro gets executed when the labels file is read. For each label it defines a macro, whose name is made up partly from the label you supplied, that contains the page, line and sub-line numbers. But first it checks to see whether the label has already been used (and complains if it has).

The initial use of `\newcommand` is to catch if `\l@dmake@labels` has been previously defined (by a class or package).

#1 page number, #2 line number, #3 sub-line number, #4 absolute line number, #5 pstart number, #6 label.

```

5949 \newcommand*{\l@dmake@labels}{%
5950 \def\l@dmake@labels#1|#2|#3|#4|#5|#6{%
5951   \expandafter\ifx\csname the@label\csuse{XR@prefix}#6\endcsname%
5952     \relax%
5953   \else%
5954     \led@warn@DuplicateLabel{\csuse{XR@prefix}#6}%
5955   \fi

```

```

5956 \global\providetoggle{label@#6@ledRcol}%False is the default value of
this toggle, which tell us if a label is linked to a right or a left side
5957 \expandafter\gdef\csname the@label\csuse{XR@prefix}#6\endcsname
{#1|#2|#3|#4|#5|\relax}%
5958 \ignorespaces}
5959
5960 %

```

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.

```

5961 \AtBeginDocument{%
5962 \def\l@dmake@labels#1|#2|#3|#4|#5|#6{%
5963 }
5964
5965 %

```

\@lab The \@lab command, which appears in the \linenum@out file, appends the current value of page, line, sub-line, and absolute line to the \labelref@list. These values are defined by the earlier \@page, \@nl, and the \sub@on and \sub@off commands appearing in the \linenum@out file.

TeX uses the page counter for page numbers. However, it appears that this is not the right place to grab the page number. That task is now done in the \edlabel macro. This version of \@lab appends just the current line and sub-line numbers to \labelref@list.

```

5966
5967 \newcommand*{\@lab}{%
5968 \ifledRcol
5969 \xright@appenditem{\linenumr@p{\line@numR}}|%
5970 \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|\the\
absline@numR}%
5971 \to\labelref@listR
5972 \else
5973 \xright@appenditem{\linenumr@p{\line@num}}|%
5974 \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|\the\absline@num}
%
5975 \to\labelref@list
5976 \fi}
5977 %

```

\applabel \applabel, if called in \edtext will insert automatically both a start and an end label for the current edtext lines.

```

5978 \newcommand*{\applabel}[1]{%
5979 \if@edtext@secondarg@%
5980 %

```

Label should not be already defined.

```

5981 \ifcsundef{the@label#1}{%
5982 \csdef{the@label#1}{applabel}%
5983 }%
5984 {%
5985 \led@warn@DuplicateLabel{#1 (applabel)}%
5986 }%
5987 %

```

Parse the `\edtext` line numbers.

```

5988 \expandafter\l@dp@rsefootsspec\l@d@nums|%
5989 %

```

Use the \TeX standard hack for label.

```

5990 \@bsphack%
5991 %

```

And now, write the data in the auxiliary file.

```

5992 \ifledRcol%
5993 \protected@write\@auxout{%
5994   {\string\l@dmake@labelsR\space\l@dparsedstartpage|\
5995   \l@dparsedstartline|\l@dparsedstartsub||\the\c@pstartR|{#1:start}}%
5996   \ifdef{\hypertarget}%
5997     {\Hy@raisedlink{\hypertarget{#1:start}}}%
5998   }%
5999   \protected@write\@auxout{%
6000     {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline
6001     ||\l@dparsedendsub|\the\c@pstartR|{#1:end}}%
6002   \else%
6003     \ifl@dpairing%pstart or pstartL?
6004     \protected@write\@auxout{%
6005       {\string\l@dmake@labelsR\space\l@dparsedstartpage|\
6006       \l@dparsedstartline|\l@dparsedstartsub||\the\c@pstartL|{#1:start}}%
6007       \ifdef{\hypertarget}%
6008         {\Hy@raisedlink{\hypertarget{#1:start}}}%
6009       }%
6010       \protected@write\@auxout{%
6011         {\string\l@dmake@labelsR\space\l@dparsedendpage|\
6012         \l@dparsedendline|\l@dparsedendsub||\the\c@pstartL|{#1:end}}%
6013       \else%
6014         \protected@write\@auxout{%
6015           {\string\l@dmake@labelsR\space\l@dparsedstartpage|\
6016           \l@dparsedstartline|\l@dparsedstartsub||\the\c@pstart|{#1:start}}%
6017           \ifdef{\hypertarget}%

```



```

6018 \fi%
6019 %

```

Use the \LaTeX standard hack for label.

```

6020 \@esphack%
6021 %

```

Warning if `\applabel` is called outside of `\edtext`.

```

6022 \else%
6023 \led@warn@AppLabelOutSecondArgEdtext{#1}%
6024 \fi%
6025 %

```

End of `\applabel`

```

6026 }%
6027 %

```

`\edlabels` `\edlabelS` and `\edlabelE` are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
6028 \newcommand{\edlabelS}[1]{%
6029 \edlabel{#1:start}%
6030 }
6031 \newcommand{\edlabelE}[1]{%
6032 \edlabel{#1:end}%
6033 }
6034 \newcommand{\edlabelSE}[1]{%
6035 \edlabelS{#1}%
6036 \edlabelE{#1}%
6037 }
6038 %

```

`\wrap@edcrossref` `\wrap@edcrossref` is called around all `reledmac` crossref commands, except those which start with `x`. It adds the hyperlink.

```

6039 \newrobustcmd{\wrap@edcrossref}[2]{%
6040 \ifdef{\hyperlink}%
6041 {\hyperlink{#1}{#2}}%
6042 {#2}%
6043 }
6044 %

```

`\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.

\LaTeX already defines a `\pageref`, so changing the name to `\edpageref`.

```

6045 \newcommand*\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@
6046   dgetref@num{1}{#1}}}
6047 \newcommand*\xpageref}[1]{\l@dgetref@num{1}{#1}}
6048 %

```

\edlineref If the specified label exists, \lineref gives its line number.

```

\lineref
6049 \newcommand*\edlineref}[1]{%
6050   \l@dref@undefined{#1}%
6051   \wrap@edcrossref{#1}{%
6052     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
6053     \iftoggle{label@#1@ledRcol}%
6054       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
6055       {\linenumrep{\l@dgetref@num{2}{#1}}}%
6056     \xflagref{#1}%
6057   }%
6058 }%
6059 \newcommand*\xlineref}[1]{\l@dgetref@num{2}{#1}}%
6060 %
6061 %

```

\sublineref If the specified label exists, \sublineref gives its sub-line number.

```

\sublineref
6062 \newcommand*\sublineref}[1]{%
6063   \l@dref@undefined{#1}%
6064   \wrap@edcrossref{#1}{%
6065     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
6066     \iftoggle{label@#1@ledRcol}%
6067       {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
6068       {\sublinenumrep{\l@dgetref@num{3}{#1}}}%
6069     }%
6070   }%
6071 \newcommand*\xsublineref}[1]{\l@dgetref@num{3}{#1}}
6072 %
6073 %

```

\xabslineref If the specified label exists, \xabslineref gives its absolute line number. That is used usually only by some reledmac internal macros.

```

6074 \newcommand*\xabslineref}[1]{\l@dgetref@num{4}{#1}}%
6075 %

```

\pstartref If the specified label exists, \pstartref gives its pstart number.

```

\pstartref
6076 \newcommand*\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@
6077   dgetref@num{5}{#1}}}%
6078 \newcommand*\xpstartref}[1]{\l@dgetref@num{5}{#1}}%

```

```
6078 %
6079 %
```

\xflagref \xflagref finds the side flag of any ref defined with \edlabel.

```
6080 \newcommand*\xflagref{1}{\l@dgetref@num{6}{#1}}%
6081 %
```

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.

```
6082 \newcommand*\l@dref@undefined{1}{%
6083   \expandafter\ifx\csname the@label#1\endcsname\relax
6084     \led@warn@RefUndefined{#1}%
6085   \fi}
6086 %
6087 %
```

\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.

```
6088 \newcommand*\l@dgetref@num{2}{%
6089   \expandafter
6090   \ifx\csname the@label#2\endcsname \relax
6091     000%
6092   \else
6093     \expandafter\expandafter\expandafter
6094     \l@dlabel@parse\csname the@label#2\endcsname|#1%
6095   \fi}
6096 %
6097 %
```

\l@dlabel@parse Notice that we slipped another | delimiter into the penultimate line of \l@dgetref@num, to keep the ‘switch-number’ separate from the reference numbers. This | is used as another parameter delimiter by \l@dlabel@parse, which extracts the appropriate number from its first arguments. The |-delimited arguments consist of the expanded label-macro (three reference numbers), followed by the switch-number (1, 2, 3, 4 or 5) which defines which of the earlier six numbers to pick out. (It was earlier given as the first argument of \l@dgetref@num.)

```

6098 \newcommand*\l@dlabel@parse#1{}
6099 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7{%
6100   \ifcase #7%
6101     \or #1%
6102     \or #2%
6103     \or #3%
6104     \or #4%
6105     \or #5%
6106     \or #6%
6107     \fi}
6108 %

```

\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.

```

6109 \newcommand*\xxref}[2]{%
6110   {%
6111     \expandafter\ifx\csname the@label#1\endcsname \relax%
6112       \expandafter\let\csname the@@label#1\endcsname\zz@@@%
6113     \else%
6114       \expandafter\def\csname the@@label#1\endcsname{\l@dgetref@num
6115         {1}{#1}|\l@dgetref@num{2}{#1}|\l@dgetref@num{3}{#1}}%
6116       \fi%
6117       \expandafter\ifx\csname the@label#2\endcsname \relax%
6118         \expandafter\let\csname the@@label#2\endcsname\zz@@@%
6119       \else%
6120         \expandafter\def\csname the@@label#2\endcsname{\l@dgetref@num
6121           {1}{#2}|\l@dgetref@num{2}{#2}|\l@dgetref@num{3}{#2}}%
6122         \fi%
6123         \letcs{\@tempa}{the@@label#1}%
6124         \letcs{\@tempb}{the@@label#2}%
6125         \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@start{#1}}%
6126         \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@end{#2}}%
6127         \linenum{\@tempa|
6128           \@tempb}}}%

```

\appref `\appref`, `\Seref`, `\apprefwithpage`, `\Serefwithpage` and `\SEonlypage` print cross-ref to some start / end lines defined by specific commands. It prints the lines as they should be printed in the apparatus (critical notes for not suffixed versions, endnotes for suffixed versions).

\apprefwithpage Here we define hooks similar to some those related to critical footnotes or endnotes.

\Seref

\Serefwithpage

\SEonlypage

So, first declare the default value of the hooks for the pseudo-series. Also declare the internal toggle which are switch by reledmac.

```

6129 \def\Xtwolines@appref{}%
6130 \def\Xtwolines@SEref{}%
6131
6132 \def\Xmorethantwolines@appref{}%
6133 \def\Xmorethantwolines@SEref{}%
6134
6135 \def\Xlinerangeseparator@appref{\endashchar}%
6136 \def\Xlinerangeseparator@SEref{\endashchar}%
6137
6138 \def\Xsublinesep@appref{\fullstop}%
6139 \def\Xsublinesep@SEref{\fullstop}%
6140
6141 \def\Xpagelinesep@appref{\fullstop}%
6142 \def\Xpagelinesep@SEref{\fullstop}%
6143
6144
6145 \newtoggle{Xtwolinesbutnotmore@appref}%
6146 \newtoggle{Xtwolinesbutnotmore@SEref}%
6147
6148 \newtoggle{Xtwolinesonlyinsamepage@appref}%
6149
6150 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
6151
6152 \newtoggle{Xlineflag@appref}%
6153 \toggletrue{Xlineflag@appref}%Here exception
6154 \newtoggle{Xlineflag@SEref}%
6155 \toggletrue{Xlineflag@SEref}%Here exception
6156
6157 \def\Xendtwolines@apprefwithpage{}%
6158 \def\Xendtwolines@SErefwithpage{}%
6159
6160 \def\Xendmorethantwolines@apprefwithpage{}%
6161 \def\Xendmorethantwolines@SErefwithpage{}%
6162
6163 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
6164 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
6165 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
6166
6167 \def\Xendbeforepagenumber@apprefwithpage{p.}%
6168 \def\Xendbeforepagenumber@SErefwithpage{p.}%
6169 \def\Xendbeforepagenumber@SEonlypage{p.}%
6170
6171 \def\Xendafterpagenumber@apprefwithpage{ }%
6172 \def\Xendafterpagenumber@SErefwithpage{ }%
6173
6174
6175 \def\Xendlineprefixsingle@apprefwithpage{}%

```

```

6176 \def\Xendlineprefixsingle@Serefwithpage{%
6177
6178 \def\Xendlineprefixmore@apprefwithpage{%
6179 \def\Xendlineprefixmore@Serefwithpage{%
6180
6181 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
6182 \newtoggle{Xendtwolinesbutnotmore@Serefwithpage}%
6183
6184 \def\Xendsublinesep@apprefwithpage{\fullstop}%
6185 \def\Xendsublinesep@Serefwithpage{\fullstop}%
6186
6187 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
6188 \newtoggle{Xendtwolinesonlyinsamepage@Serefwithpage}%
6189
6190 \newtoggle{Xendlineflag@apprefwithpage}
6191 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
6192 \newtoggle{Xendlineflag@Serefwithpage}
6193 \toggletrue{Xendlineflag@Serefwithpage}%Here, exception
6194
6195 %

```

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.

```

6196
6197 \gdef\Xboxstartlinenum@appref{Opt}
6198 \gdef\Xboxstartlinenum@Seref{Opt}
6199
6200 \gdef\Xboxendlinenum@appref{Opt}
6201 \gdef\Xboxendlinenum@Seref{Opt}
6202
6203 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}
6204 \gdef\Xendboxstartlinenum@Serefwithpage{Opt}
6205
6206 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
6207 \gdef\Xendboxendlinenum@Serefwithpage{Opt}
6208
6209 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
6210 \newtoggle{Xendpagenumberonlyfirst@Serefwithpage}
6211
6212 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
6213 \newtoggle{Xendpagenumberonlyfirstifsingle@Serefwithpage}
6214
6215 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
6216 \newtoggle{Xendpagenumberonlyfirstintwo@Serefwithpage}
6217
6218 \gdef\Xendsympagenum@apprefwithpage{}
6219 \gdef\Xendsympagenum@Serefwithpage{}
6220
6221 \gdef\Xendinplaceofpagenumber@apprefwithpage{}

```

```

6222 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
6223
6224 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

6225 \newcommand\@apprefprefixsingle{}%
6226 \newcommand\@SErefprefixsingle{}%
6227
6228 \newcommand\@apprefprefixmore{}%
6229 \newcommand\@SErefprefixmore{}%
6230
6231 \newcommand{\setapprefprefixsingle}[1]{%
6232   \gdef\@apprefprefixsingle{#1}%
6233 }
6234 \newcommand{\setSErefprefixsingle}[1]{%
6235   \gdef\@SErefprefixsingle{#1}%
6236 }
6237
6238 \newcommand{\setapprefprefixmore}[1]{%
6239   \gdef\@apprefprefixmore{#1}%
6240 }
6241 \newcommand{\setSErefprefixmore}[1]{%
6242   \gdef\@SErefprefixmore{#1}%
6243 }
6244
6245 %

```

And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.

```

6246 \newcommand{\setSErefonlypageprefixsingle}[1]{%
6247   \gdef\SErefonlypage@prefixsingle{#1}%
6248 }%
6249 \newcommand{\setSErefonlypageprefixmore}[1]{%
6250   \gdef\SErefonlypage@prefixmore{#1}%
6251 }%
6252 %

```

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.

```

6253
6254 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
6255 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
6256
6257 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
6258 {#1}{#2}{appref}}
6259 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
6260 {#1}{#2}{SEref}}

```

```

6259 \newcommandx{\Serefonlypage}[2][1,usedefault]{\reformattedonlypage@
        {#1}{#2}{Seref}}
6260
6261
6262 \newcommand{\reformatted@}[3]{%
6263     \def\do##1{%
6264         \setkeys[mac]{truefootnoteoption}{##1}%
6265     }%
6266     \notblank{#1}{\docsvlist{#1}}{%}%
6267     \xdef\@currentseries{#3}%
6268     \ifcempty{@#3prefixmore}%
6269         {\@apprefprefixsingle}%
6270     {%
6271         \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}{%
6272             {\csuse{@#3prefixsingle}}%
6273             {\csuse{@#3prefixmore}}%
6274         }%
6275     \ifboolexpr{%
6276         test{\ifcsundef{the@label#2:start}}}%
6277         or test{\ifcsundef{the@label#2:end}}}%
6278     }%
6279     {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6280     {%
6281         \def\@this@crossref@start{#2:start}%
6282         \def\@this@crossref@end{#2:end}%
6283         \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|
xflagref{#2:start}}|}%
6284         \undef\@this@crossref@end%
6285         \undef\@this@crossref@start%
6286     }%
6287     \def\do##1{%
6288         \setkeys[mac]{falsefootnoteoption}{##1}%
6289     }%
6290     \notblank{#1}{\docsvlist{#1}}{%}%
6291 }%
6292
6293 \newcommand{\reformattedwithpage@}[3]{%
6294     \def\do##1{%
6295         \setkeys[mac]{truefootnoteoption}{##1}%
6296     }%
6297     \notblank{#1}{\docsvlist{#1}}{%}%
6298     \xdef\@currentseries{#3withpage}%
6299     \ifboolexpr{%
6300         test{\ifcsundef{the@label#2:start}}}%
6301         or test{\ifcsundef{the@label#2:end}}}%
6302     }%
6303     {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6304     {%
6305         \def\@this@crossref@start{#2:start}%

```



```

6306 \def\@this@crossref@end{#2:end}%
6307 \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|%
6308 \undef\@this@crossref@end%
6309 \undef\@this@crossref@start%
6310 }%
6311 \def\do##1{%
6312 \setkeys[mac]{falsefootnoteoption}{##1}%
6313 }%
6314 \notblank{#1}{\docsvlist{#1}}{ }%
6315 }%
6316
6317 \newcommand{\reformattedonlypage@}[3]{%
6318 \def\do##1{%
6319 \setkeys[mac]{truefootnoteoption}{##1}%
6320 }%
6321 \notblank{#1}{\docsvlist{#1}}{ }%
6322 \xdef\@currentseries{#3onlypage}%
6323 \ifboolexpr{%
6324 test{\ifcsundef{the@label#2:start}}%
6325 or test{\ifcsundef{the@label#2:end}}%
6326 }%
6327 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6328 {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
6329 {%
6330 \ifcsvoid{#3onlypage@prefixsingle}%
6331 {}%
6332 {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
}}%
6333 \printnpnum{%
6334 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6335 }%
6336 }%
6337 {%
6338 \ifcsvoid{#3onlypage@prefixmore}%
6339 {}%
6340 {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
6341 \ifdefined\linangesep@%
6342 \printnpnum{%
6343 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6344 \linangesep@%
6345 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
6346 }%
6347 \else%
6348 \printnpnum{%
6349 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6350 \csuse{Xendlinangeseparator@}\@currentseries%
6351 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
6352 }%

```

```

6353     \fi%
6354   }%
6355 }%
6356 \def\do##1{%
6357   \setkeys[mac]{falsefootnoteoption}{##1}%
6358 }%
6359 \notblank{#1}{\docsvlist{#1}}{ }%
6360 }%
6361 %

```

\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`.

```

6362 \newcommand*{\edmakelabel}[2]{\expandafter\edef\csname the@label#1\
endcsname{#2}}
6363
6364 %

```

(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. 131 and V.9 p. 98), 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`.

```

6365 \pretocmd{\XR@test}%
6366   {\XR@test@mac+++#1#2#3#4+++}%
6367 {}%
6368 {}%
6369 %

```

\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`.

```

6370 \long\def\xr@test@mac+++#1+++{\xr@test@mac@test#1}
6371 %

```

`\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`.

```

6372 \long\def\xR@test@mac@test#1#2...{%The triple dots (NOT \ldots) are because
    of the line 22 of xr.sty v5.02 1994/05/28
6373 \ifx#1\l@dmake@labels%
6374 \l@dmake@labels#2%
6375 \else
6376 \ifx#1\l@dmake@labelsR%
6377 \l@dmake@labelsR #2%
6378 \fi%
6379 \fi%
6380 }%
6381 %

```

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.

```

6382 \pretocmd{\@xympar}%
6383 {\ifnumberedpar@
6384 \led@warn@NoMarginpars
6385 \@esphack
6386 \else}%
6387 {}%
6388 {}%
6389
6390 \apptocmd{\@xympar}%
6391 {\fi}%
6392 {}
6393 {}
6394
6395 %

```

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

```

6396 \newcount\sidenote@margin
6397 \newcommand*{\sidenotemargin}[1]{\{%
6398   \l@dgetsidenote@margin{#1}%
6399   \ifnum\@l@dtmptcntb>\m@ne
6400     \ifledRcol
6401       \global\sidenote@marginR=\@l@dtmptcntb
6402     \else
6403       \global\sidenote@margin=\@l@dtmptcntb
6404     \fi
6405   \fi}}
6406 \newcommand*{\l@dgetsidenote@margin}[1]{\%
6407   \def\@tempa{#1}\def\@tempb{left}%
6408   \ifx\@tempa\@tempb
6409     \@l@dtmptcntb \z@
6410   \else
6411     \def\@tempb{right}%
6412     \ifx\@tempa\@tempb
6413       \@l@dtmptcntb \@ne
6414     \else
6415       \def\@tempb{outer}%
6416       \ifx\@tempa\@tempb
6417         \@l@dtmptcntb \tw@
6418       \else
6419         \def\@tempb{inner}%
6420         \ifx\@tempa\@tempb
6421           \@l@dtmptcntb \thr@@
6422         \else
6423           \led@warn@BadSidenotemargin
6424           \@l@dtmptcntb \m@ne
6425         \fi
6426       \fi
6427     \fi
6428   \fi}
6429 \sidenotemargin{right}
6430
6431 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
6432 \newbox\l@dlp@rbox
6433 \newbox\l@drp@rbox
6434
6435 %

```

`\ledlsnotewidth` These specify the width of the left/right boxes (initialised to `\marginparwidth`), their
`\ledrsnotewidth` distance from the text (initialised to `\linenumsep`), and the fonts used.
`\ledlsnotesep`
`\ledrsnotesep`
`\ledlsnotefontsetup`
`\ledrsnotefontsetup`

```

6436 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
6437 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
6438 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
6439 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
6440 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
6441 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
6442
6443 %

```

`\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.

```

6444 \newcommand*\ledleftnote[1]{\edtext{}\l@dlsnote{#1}}
6445 \newcommand*\ledrightnote[1]{\edtext{}\l@drsnote{#1}}
6446 \newcommand*\ledsidenote[1]{\edtext{}\l@dcsnote{#1}}%
6447 \newcommand*\ledinnernote[1]{\edtext{}\l@disnote{#1}}%
6448 \newcommand*\ledouternote[1]{\edtext{}\l@dosnote{#1}}%
6449 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.

```

6450 \l@dcsnote \newif\ifrighnoteup
6451 \l@desnote \rightnoteuptrue
6452 \l@disnote
6453 \newcommand*\l@dlsnote[1]{%
6454   \begingroup%
6455   \newcommand{\content}{#1}%
6456   \ifnumberedpar@
6457     \ifledRcol%
6458       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
6459         \to\inserts@listR
6460       \global\advance\insert@countR \@ne%
6461     \else%
6462       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%
6463         \to\inserts@list
6464       \global\advance\insert@count \@ne%
6465     \fi
6466   \fi%
6467   \ignorespaces%
6468   \endgroup%
6469 }%
6470
6471 \newcommand*\l@drsnote[1]{%
6472   \begingroup%
6473   \newcommand{\content}{#1}%
6474   \ifnumberedpar@
6475     \ifledRcol%

```

```

6476 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
6477 \to\inserts@listR
6478 \global\advance\insert@countR \@ne%
6479 \else%
6480 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
6481 \to\inserts@list
6482 \global\advance\insert@count \@ne%
6483 \fi
6484 \fi\ignorespaces%
6485 \endgroup%
6486 }%
6487
6488 \newcommand*{\l@dcnote}[1]{%
6489 \begingroup%
6490 \newcommand{\content}{#1}%
6491 \ifnumberedpar@
6492 \ifledRcol%
6493 \xright@appenditem{\noexpand\vl@dcnote{\expandonce\content}}}%
6494 \to\inserts@listR
6495 \global\advance\insert@countR \@ne%
6496 \else%
6497 \xright@appenditem{\noexpand\vl@dcnote{\expandonce\content}}}%
6498 \to\inserts@list
6499 \global\advance\insert@count \@ne%
6500 \fi
6501 \fi\ignorespaces%
6502 \endgroup%
6503 }%
6504
6505 \newcommand*{\l@disnote}[1]{%
6506 \begingroup%
6507 \newcommand{\content}{#1}%
6508 \ifnumberedpar@%
6509 \ifledRcol%
6510 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
6511 \to\inserts@listR%
6512 \global\advance\insert@countR \@ne%
6513 \else%
6514 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
6515 \to\inserts@list%
6516 \global\advance\insert@count \@ne%
6517 \fi%
6518 \fi\ignorespaces%
6519 \endgroup%
6520 }%
6521
6522 \newcommand*{\l@dosnote}[1]{%
6523 \begingroup%
6524 \newcommand{\content}{#1}%
6525 \ifnumberedpar@%

```

```

6526 \ifledRcol%
6527 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
6528 \to\inserts@listR%
6529 \global\advance\insert@countR \@ne%
6530 \else%
6531 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
6532 \to\inserts@list%
6533 \global\advance\insert@count \@ne%
6534 \fi%
6535 \fi\ignorespaces%
6536 \endgroup%
6537 }%
6538
6539 %

```

`\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.

`\vl@disnote` and `\vl@dosnote` just call `\vl@dlsnote` or `\vl@drsnote`, depending of the page.

```

6540 \newcommand*{\vl@dlsnote}[1]{%
6541 \ifledRcol%
6542 \@l@tempcntb=\sidenote@marginR%
6543 \ifnum\@l@tempcntb>\@ne%
6544 \advance\@l@tempcntb by\page@numR%
6545 \fi%
6546 \else%
6547 \@l@tempcntb=\sidenote@margin%
6548 \ifnum\@l@tempcntb>\@ne%
6549 \advance\@l@tempcntb by\page@num%
6550 \fi%
6551 \fi%
6552 \ifodd\@l@tempcntb%
6553 \listgadd{\l@dcsnotetext@l}{#1}%

```

```

6554 \else%
6555 \listgadd{\l@dcsnotetext}{#1}%
6556 \fi
6557 }
6558 \newcommand*{\vl@drsnote}[1]{%
6559 \ifledRcol%
6560 \@l@tempcntb=\sidenote@marginR%
6561 \ifnum\@l@tempcntb>\@ne%
6562 \advance\@l@tempcntb by\page@numR%
6563 \fi%
6564 \else%
6565 \@l@tempcntb=\sidenote@margin%
6566 \ifnum\@l@tempcntb>\@ne%
6567 \advance\@l@tempcntb by\page@num%
6568 \fi%
6569 \fi%
6570 \ifodd\@l@tempcntb%
6571 \listgadd{\l@dcsnotetext}{#1}%
6572 \else%
6573 \listgadd{\l@dcsnotetext@r}{#1}%
6574 \fi%
6575 }
6576 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
6577
6578 \newcommand{\vl@disnote}[1]{%
6579 \ifledRcol%
6580 \@tempcnta=\page@numR%
6581 \else%
6582 \@tempcnta=\page@num%
6583 \fi%
6584 \ifodd\@tempcnta% ODD => right page => inner side = left side
6585 \vl@dlsnote{#1}%
6586 \else%
6587 \vl@drsnote{#1}%
6588 \fi%
6589 }%
6590
6591 \newcommand{\vl@dosnote}[1]{%
6592 \ifledRcol%
6593 \@tempcnta=\page@numR%
6594 \else%
6595 \@tempcnta=\page@num%
6596 \fi%
6597 \ifodd\@tempcnta% ODD => right page => outer side = right side
6598 \vl@drsnote{#1}%
6599 \else%
6600 \vl@dlsnote{#1}%
6601 \fi%
6602 }%
6603

```


6604 %

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box. And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```
6605 \newcommand*{\setl@dlp@rbox}[1]{%
6606   \begingroup%
6607   \parindent\z@ \hsize=\ledlsnotewidth%
6608   \ledlsnotefontsetup% We kept it outside of the vbox, because can affect
the ragging
6609   \global\setbox\l@dlp@rbox%
6610   \ifleftnoteup%
6611     =\vbox to\z@{\{\ledlsnotefontsetup\vss #1}\}% We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
6612   \else%
6613     =\vbox to 0.70\baselineskip{\{\ledlsnotefontsetup\strut#1\vss}\}%
6614   \fi%
6615   \endgroup%
6616 }
6617
6618 \newcommand*{\setl@drp@rbox}[1]{%
6619   \begingroup%
6620   \parindent\z@ \hsize=\ledrsnotewidth%
6621   \ledrsnotefontsetup% We kept it outside of the vbox, because can affect
the ragging
6622   \global\setbox\l@drp@rbox%
6623   \ifrightrightnoteup%
6624     =\vbox to\z@{\{\ledrsnotefontsetup\vss#1}\}% We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
6625   \else%
6626     =\vbox to 0.7\baselineskip{\{\ledrsnotefontsetup\strut#1\vss}\}%
6627   \fi%
6628   \endgroup%
6629 }%
6630 \newif\ifleftnoteup
6631 \leftnoteuptrue
6632 %
```

`\@sidenotesep` This macro is used to separate sidenotes of the same line.

```
6633 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
6634 \newcommand{\@sidenotesep}{, }
6635 %
```

`\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.

```

6636 \newcommand*{\affixside@note}{%
6637   \prepare@edindex@for@note{\the\page@num|\the\line@num|\the\subline@num|\
the\page@num|\the\line@num|\the\subline@num}%
6638   \def\sidenotecontent@{%
6639     \numgdef{\itemcount@}{0}%
6640     \def\do##1{%
6641       \ifnumequal{\itemcount@}{0}%
6642         {%
6643           \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
6644         {\appto\sidenotecontent@{\@sidenotesep ##1}%
6645         }%
6646         \numgdef{\itemcount@}{\itemcount@+\@one}%
6647     }%
6648     \dolistloop{\l@dcsof@text}%
6649     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
6650 }%
```

And we do the same for left and right notes (not movable).

```

6651 \gdef\@templ@d{%
6652 \gdef\@templ@n{\l@dcsof@text\l@dcsof@text@l\l@dcsof@text@r}%
6653 \ifx\@templ@d\@templ@n \else%
6654   \if@twocolumn%
6655     \if@firstcolumn%
6656       \setl@dlp@rbox{##1}{\sidenotecontent@}%
6657     \else%
6658       \setl@drp@rbox{\sidenotecontent@}%
6659     \fi%
6660   \else%
6661     \@l@dtmpcntb=\sidenote@margin%
6662     \ifnum\@l@dtmpcntb>\@one%
6663       \advance\@l@dtmpcntb by\page@num%
6664     \fi%
6665     \ifodd\@l@dtmpcntb%
6666       \setl@drp@rbox{\sidenotecontent@}%
6667       \gdef\sidenotecontent@{%
6668         \numgdef{\itemcount@}{0}%
6669         \dolistloop{\l@dcsof@text@l}%
6670         \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
6671         \setl@dlp@rbox{\sidenotecontent@}%
6672       \else%
6673         \setl@dlp@rbox{\sidenotecontent@}%
6674         \gdef\sidenotecontent@{%
6675           \numgdef{\itemcount@}{0}%
6676           \dolistloop{\l@dcsof@text@r}%
6677           \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
6678           \setl@drp@rbox{\sidenotecontent@}%

```

```

6679     \fi%
6680     \fi%
6681     \fi%
6682     \advance\@edindex@fornote@m@ne%
6683   }
6684   %

```

XXV Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage`.
`\l@dfetendmini` They can be extended to handle other things if necessary.

```

6685 \ifnoledgroup@else%
6686 \newcommand*\l@dfetbeginmini{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
6687 \newcommand*\l@dfetendmini{%
6688   \IfStrEq{critical-familiar}{\@mpfnpos}%
6689   {\l@dedendmini\l@dfamendmini}%
6690   {%
6691     \IfStrEq{familiar-critical}{\@mpfnpos}%
6692     {\l@dfamendmini\l@dedendmini}%
6693     {\do@feet@custom@order{mp@}{\@mpfnpos}}%
6694   }%
6695   }%
6696 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.
`\l@dedendmini`
`\mp@append@Xnotes`

```

6697 \newcommand*\l@dedbeginmini{%
6698   \unless\ifnocritical@%
6699   \def\do##1{%
6700     \csletcs{v##1footnote}{mpv##1footnote}%
6701   }%
6702   \dolistloop{\@series}%
6703   \fi%
6704   }
6705 \newcommand*\l@dedendmini{%
6706   \unless\ifnocritical@%
6707   \ifl@dpairing%
6708     \ifledRcol%
6709       \flush@notesR%
6710     \else%
6711       \flush@notes%
6712     \fi%

```

```

6713 \fi
6714 \def\do##1{%
6715 \mp@append@Xnotes{##1}%
6716 }%
6717 \dolistloop{\@series}%
6718 \fi%
6719 }%
6720 \newcommand{\mp@append@Xnotes}[1]{%
6721 \ifvoid\csuse{mp#1footins}\else%
6722 \ifl@dpairing%
6723 \ifparledgroup%
6724 \ifledRcol%
6725 \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
6726 \else%
6727 \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\
skip\@nameuse{mp#1footins}}%
6728 \fi%
6729 \fi%
6730 \fi%
6731 \ifcsstring{series@display#1}{paragraph}{%
6732 \setbox\@nameuse{mp#1footins}=\vbox{%
6733 \csuse{Xnotefontsize@#1}%
6734 \ifcsdef{Xhsize\csuse{series@display#1}@#1}{%
6735 \hsize\csuse{Xhsize\csuse{series@display#1}@#1}%
6736 }{%
6737 \noindent\csuse{Xtxtbeforenotes@#1}%
6738 \unvbox\@nameuse{mp#1footins}%
6739 \@parboxrestore%
6740 }%
6741 }%
6742 \csuse{mp#1footgroup}{#1}%
6743 \fi%
6744 }%
6745 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
6746 \newcommand*\l@dfambeginmini{%
6747 \unless\ifnofamiliar%
6748 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
6749 \dolistloop{\@series}%
6750 \fi%
6751 }%
6752
6753 \newcommand*\l@dfamendmini{%
6754 \unless\ifnofamiliar%
6755 \def\do##1{%
6756 \mp@append@notesX{##1}%
6757 }%

```

```

6758 \dolistloop{\@series}%
6759 \fi%
6760 }%
6761 \newcommand{\mp@append@notesX}[1]{%
6762 \ifvoid\csuse{mpfootins#1}\else%
6763 \csuse{mpfootgroup#1}{#1}%
6764 \fi%
6765 }%
6766 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

6767 \patchcmd%
6768 {\@iiiminipage}%
6769 {\let\@footnotetext\@mpfootnotetext}%
6770 {\let\@footnotetext\@mpfootnotetext\l@dfetbeginmini}%
6771 {}%
6772 {\led@error@fail@patch@\@iiiminipage}%
6773 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

6774 \patchcmd%
6775 {\endminipage}%
6776 {\footnoterule}%
6777 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
6778 {}%
6779 {\led@error@fail@patch@endminipage}
6780
6781 \patchcmd%
6782 {\endminipage}%
6783 {\@minipagefalse}%
6784 {\l@dfetendmini\@minipagefalse}%
6785 {}%
6786 {\led@error@fail@patch@endminipage}
6787
6788 %

```

`\l@dunboxmpfoot` `\l@dunboxmpfoot` insert normal footnotes for `ledgroup`.
`edgroup@beforenormalnotes`

```

6789 \newcommand*{\l@dunboxmpfoot}{%
6790 \vskip\skip\@mpfootins
6791 \normalcolor
6792 \footnoterule
6793 \l@advance@parledgroup@beforenormalnotes
6794 \unvbox\@mpfootins%
6795 }
6796 %

```

When using parallel `ledgroup`, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

6797 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
6798   \ifparledgroup
6799     \ifl@dpairing
6800       \ifledRcol
6801         \dimgdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
6802       \else
6803         \dimgdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
6804       \fi
6805     \fi
6806   \fi
6807 }
6808 %

```

`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.

```

6809
6810 \newenvironment{ledgroup}{%
6811   \resetprevpage@num%
6812   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
6813   \let\@footnotetext\@mpfootnotetext
6814   \l@dfeetbeginmini%
6815 }{%
6816   \par
6817   \unskip
6818   \ifvoid\@mpfootins\else
6819     \l@dunboxmpfoot
6820   \fi
6821   \l@dfeetendmini%
6822   \@ledgroupfalse%
6823 }
6824
6825
6826 %

```

`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.

```

6827 \newenvironment{ledgroupsize}[2][1]{%
6828 %

```

Set the various text measures.

```

6829   \hspace #2\relax
6830 %

```

Initialize fills for centering.

```

6831 \let\ledllfill\hfil
6832 \let\ledrlfill\hfil
6833 \def\@tempa{#1}\def\@tempb{1}%
6834 %

```

Left adjusted numbered lines

```

6835 \ifx\@tempa\@tempb
6836 \let\ledllfill\relax
6837 \else
6838 \def\@tempb{r}%
6839 \ifx\@tempa\@tempb
6840 %

```

Right adjusted numbered lines

```

6841 \let\ledrlfill\relax
6842 \fi
6843 \fi
6844 %

```

Set up the footnoting.

```

6845 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
6846 \let\@footnotetext\@mpfootnotetext
6847 \l@dfeetbeginmini%
6848 }{f%
6849 \par
6850 \unskip
6851 \ifvoid\@mpfootins\else
6852 \l@dunboxmpfoot
6853 \fi
6854 \l@dfeetendmini%
6855 }
6856
6857 %

```

Close the \ifnoledgroup@else.

```

6858 \fi%
6859 %

```

\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.

```

6860 \newif\ifledgroupnotesL@
6861 \newif\ifledgroupnotesR@
6862 %

```

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.

```

6863 \AtBeginDocument{%
6864   \unless\ifl@imakeidx%
6865     \ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
6866   %
6867   \fi%
6868   \unless\ifl@indextools%
6869     \ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}
6870   %
6871   \fi%
6872   \unless\ifl@footmisc%
6873     \ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
6874   %
6875   \fi%
6876 }
6877 %

```

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
6875 \newcommand{\pagelinesep}{-}
6876 \newcommand{\edindexlab}{\&\&}
6877 \newcounter{labidx}
6878 \setcounter{labidx}{0}
6879
6880 %

```

\doedindexlabel This macro sets an \edlabel.

```

6881 \newcommand{\doedindexlabel}{%
6882   \stepcounter{labidx}%
6883   \edlabel{\edindexlab\thelabidx}%
6884 }
6885
6886 %

```

\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.

```

6887 \newcounter{pageline}%
6888 \renewcommand{\thepageline}{%
6889   \thepage%
6890   \pagelinesep%
6891   \xlineref{\edindexlab\thelabidx}%

```



```
6892 }
6893 %
```

`\thestartpageline` These macros make up the page/line start/end number when the `\edindex` command is called in critical notes.

`\theendpageline`

```
6894 \newcommand{\thestartpageline}{%
6895   \l@dparsedstartpage%
6896   \pagelinesep%
6897   \l@dparsedstartline%
6898 }
6899 \newcommand{\theendpageline}{%
6900   \l@dparsedendpage%
6901   \pagelinesep%
6902   \l@dparsedendline%
6903 }
6904 %
```

XXVI.3 Code specific to `\edindex` in critical footnotes

`\@edindex@fornote@` This counter is incremented at the beginning of each note (either a footnote or a sidenote), and decremented at the end of each note. If its value is greater than 0, that means we are inside a note.

```
6905 \newcount\@edindex@fornote@
6906 %
```

`\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 `|`.

```
6907 \newcommand{\prepare@edindex@fornote}[1]{%
6908   \l@dp@rsefootspec#1| %
6909   \advance\@edindex@fornote@\@ne%
6910 }
6911 %
```

`\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.

```
6912 \newcommand{\get@edindex@ledinnote@command}{%
6913   \ifxindy@%
6914     \gdef\@ledinnote@command{%
6915       ledinnote\thelabidx%
6916     }%
6917   \ifxindyhyperref@%
```

```

6918 \immediate\write\eledmac@xindy@out{%
6919 (define-attributes ("ledinnote\thelabidx"))^^J
6920 \space\space(markup-locref^^J
6921 \eledmacmarkuplocdepth^^J
6922 :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
}{ "^^J
6923 :close "}"^^J
6924 :attr "ledinnote\thelabidx"^^J
6925 )
6926 }%
6927 \else%
6928 \immediate\write\eledmac@xindy@out{%
6929 (define-attributes ("ledinnote\thelabidx"))^^J
6930 \space\space(markup-locref^^J
6931 \eledmacmarkuplocdepth^^J
6932 :open "\string\ledinnote{\@index@command}{ "^^J
6933 :close "}"^^J
6934 :attr "ledinnote\thelabidx"^^J
6935 )
6936 }%
6937 \fi%
6938 %

```

If we do not use xindy option, \@ledinnote@command will produce something like ledinnote{formattingcommand}.

```

6939 \else%
6940 \gdef\@ledinnote@command{%
6941 ledinnote[\edindexlab\thelabidx]{\@index@command}%
6942 }%
6943 \fi%
6944 }
6945 %

```

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 |.

```

6946 \def\get@index@command#1|#2+{%
6947 \gdef\@index@txt{#1}%
6948 \gdef\@index@command{#2}%
6949 \xdef\@index@parenthesis{%
6950 \IfBeginWith{\@index@command}{(}{%
6951 \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
6952 \global\let\@index@command\@index@command@%
6953 \xdef\@index@parenthesis{(%
6954 }}%
6955 \IfBeginWith{\@index@command}{)}{%
6956 \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
6957 \global\let\@index@command\@index@command@%

```

```

6958 \xdef\@index@parenthesis{)}%
6959 }{}%
6960 }
6961 %

```

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`

```

6962 \newcommand{\ledinnote}[3][1,usedefault]{%
6963   \ifboolexpr{%
6964     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
6965     or%
6966     bool {xindyhyperref@}%
6967   }%
6968   {%
6969     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
6970   }%
6971   {%
6972     \csuse{#2}{\ledinnotemark{#3}}%
6973   }%
6974 }%
6975 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage
6976   {#2}}}}%
6977 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
6978 %

```

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`

```

6978 \newcommand{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
6979   index name, #2 = the text
6979   \ifl@imakeidx%
6980     \ifnum\@edindex@fornote@>\z@%
6981       \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6982       \get@edindex@ledinnote@command%

```

```

6983 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}\the startpageline}%
6984 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}\the endpageline}%
6985 \else%
6986 \get@edindex@hyperref{#2}%
6987 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
6988 \fi%
6989 \else%
6990 \ifnum\@edindex@fornote@>\z@%
6991 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6992 \get@edindex@ledinnote@command%
6993 \expandafter\protected@write\@indexfile{}%
6994 {\string\indexentry{\@index@txt|(\@ledinnote@command)}{\the startpageline}
}\%
6995 \expandafter\protected@write\@indexfile{}%
6996 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\the endpageline}
}\%
6997 \else%
6998 \protected@write\@indexfile{}%
6999 {\string\indexentry{#2}{\thepageline}
}\%
7000 \fi%
7001 \endgroup
7002 \esphack%
7003 }
7004 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

7009 \pretocmd{\makeindex}{%
7010 \def\edindex{%
7011 \ifbool{bool{numbering} or bool{numberingR} or bool{
l@dprintingpages} or bool{l@dprintingcolumns}}{%
7012 \bsphack%
7013 \doedindexlabel%
7014 \begingroup%
7015 \sanitize%
7016 \wredindex%
7017 }%
7018 {%
7019 \led@warn@edinde@outsidenumbering%
7020 \index%
7021 }%
7022 }%
7023 }%
7024 {}%
7025 {\led@error@fail@patch@makeindex}%
7026 \newcommand{\edindex}[1]{\bsphack\esphack}

```

```
7027 %
```

XXVI.7 Hyperlink

\hyperlinkformat \hyperlinkformat command is to be used to have both a internal hyperlink and a format, when indexing.

```
7028 \newcommand{\hyperlinkformat}[3]{%
7029   \ifstrempy{#1}%
7030     {\hyperlink{#2}{#3}}%
7031     {\csuse{#1}{\hyperlink{#2}{#3}}}%
7032   }%
7033 %
```

\hyperlinkR \hyperlinkR command is to be used to create a internal hyperlink and \ledRflag, when indexing.

```
7034 \newcommand{\hyperlinkR}[2]{%
7035   \hyperlink{#1}{#2\@Rlineflag}%
7036 }%
7037
7038 %
```

\hyperlinkformatR \hyperlinkformatR command is to be used to create a internal hyperlink, a format and a \@Rlineflag, when indexing.

```
7039 \newcommand{\hyperlinkformatR}[3]{%
7040   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
7041 }%
7042
7043 %
```

\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).

```
7044 \newcommand{\get@edindex@hyperref}[1]{%
7045 %
```

We have to disable temporary spaces to work through a xstring bug (or feature?)

```
7046 \edef\temp@{%
7047   \catcode`\ =9 %space need for catcode
7048   \detokenize{#1}%For active character in unicode
7049   \catcode`\ =10 % space need for catcode
7050 }%
7051 %
```

Now, we define \@edindex@hyperref if the hyperindex of hyperref is enabled.

```

7052 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
7053   \IfSubStr{\temp@}{|}%
7054   {\get@index@command#1+%
7055    \ifledRcol%
7056     \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7057     hyperlinkformatR{\@index@command}%
7058     {\edindexlab\thelabidx}}%
7059   }%
7060   \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7061   hyperlinkformat{\@index@command}%
7062   {\edindexlab\thelabidx}}%
7063   \fi%
7064 }%
7065 {\get@index@command#1|+%
7066  \ifledRcol%
7067   \gdef\@edindex@hyperref{hyperlinkR{\edindexlab\thelabidx}}%
7068  }%
7069  \gdef\@edindex@hyperref{hyperlink{\edindexlab\thelabidx}}%
7070  \fi%
7071 }%
7072 }%
7073 %

```

```

7074 % If we use both xindy and hyperref, first get the \protect\cs{
7075 index@command} command.
7076 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
7077 % \begin{macrocode}
7078 {\ifxindyhyperref%
7079  \IfSubStr{\temp@}{|}%
7080  {\get@index@command#1+%
7081   {\get@index@command#1|+%
7082   \gdef\@edindex@hyperref{eledmac\thelabidx}}%
7083 }%
7084 }%

```

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`.

```

7083 \IfStrEq{\@index@parenthesis}{(}%
7084 {%
7085  \csxdef{xindy@parenthesis@\@index@txt}{\thelabidx}%
7086  \gdef\@edindex@hyperref{| (eledmac\thelabidx}%
7087 }%
7088 }%
7089 %

```

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`.

```

7090 \IfStrEq{\@index@parenthesis}{)}%

```

```

7091 {%
7092 \xdef\@edindex@hyperref{|\}eledmac\csuse{xindyparenthesis@
@index@txt}}}%
7093 \global\csundef{xindyparenthesis@|\@index@txt}}%
7094 }%
7095 %

```

Write in the .xdy file the attributes of the location.

```

7096 {%
7097 \immediate\write\eledmac@xindy@out{%
7098 (define-attributes ("eledmac\thelabidx"))^^J
7099 \space\space(markup-locref^^J
7100 \eledmacmarkuplocdepth^^J
7101 :open "\string\hyperlink%
7102 \ifledRcol R\fi%
7103 {\edindexlab\thelabidx}%
7104 {\ifdefempty{\@index@command}%
7105 {}%
7106 {\@backslashchar\@index@command}%
7107 {"^^J
7108 :close "}}^^J
7109 :attr "eledmac\thelabidx"^^J
7110 )
7111 }%
7112 }%
7113 %

```

And now, in any other case.

```

7114 \else%
7115 \gdef\@index@txt{#1}%
7116 \gdef\@edindex@hyperref{}%
7117 \fi%
7118 }%
7119 }
7120 %

```

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.

```

7121 \newcommand{\led@set@index@fornote}[1]{%
7122 \ifbool{indtl@innote}%
7123 {\let\index\nindex}%
7124 {}%
7125 \ifbool{indtl@notenumber}%
7126 {}%

```

```

7127 \renewcommand{\index}[2][\indtl@jobname]{%
7128   \orig@@index[##1]{%
7129     ##2|innotenumber{\this@footnoteX@reading}%
7130   }%
7131 }%
7132 }%
7133 {}%
7134 }%
7135 %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

7136 \newcommand{\led@reinit@index@fornote}{%
7137   \ifbool{indtl@innote}%
7138     {\let\index\orig@@index}%
7139   }%
7140   \ifbool{indtl@notenumber}%
7141     {\let\index\orig@@index}%
7142   }%
7143 }%
7144 %

```

XXVII Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

7145 \AtBeginDocument{%
7146   \@ifpackageloaded{glossaries}{%
7147     %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

7148   \renewcommand{\do}[1]{%
7149     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
7150       \doedindexlabel%
7151       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
7152     }%
7153     \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,3,usedefault]{%
7154       \doedindexlabel%
7155       \csname#1\endcsname*[counter=pageline,##1]{##2}[##3]%
7156     }%
7157   }%
7158   \docsvlist{%
7159     gls,%
7160     Gls,%
7161     GLS,%
7162     glspl,%

```



```

7163     Glspl,%
7164     GLSpl,%
7165     glstext,%
7166     Glstext,%
7167     GLStext,%
7168     Glsfirst,%
7169     GLSfirst,%
7170     glsplural%
7171     Glsplural,%
7172     GLSplural,%
7173     glsfirstplural,%
7174     Glsfirstplural,%
7175     GLSfirstplural,%
7176     glsname,%
7177     Glsname,%
7178     GLSname,%
7179     glssymbol,%
7180     Glssymbol,%
7181     GLSsymbol,%
7182     glsdesc,%
7183     Glsdesc,%
7184     GLSdesc,%
7185     glsuseri,%
7186     Glsuseri,%
7187     GLSuseri,%
7188     glsuserii,%
7189     Glsuserii,%
7190     GLSuserii,%
7191     glsuseriii,%
7192     Glsuseriii,%
7193     GLSuseriii,%
7194     glsuseriv,%
7195     Glsuseriv,%
7196     GLSuseriv,%
7197     glsuserv,%
7198     Glsuserv,%
7199     GLSuserv,%
7200     glsuservi,%
7201     Glsuservi,%
7202     GLSuservi%
7203     }%
7204 %

```

Then those which arguments are `[<options>]{<label>}{<link text>}`.

```

7205     \renewcommand{\do}[1]{%
7206     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,
usedefault]{%
7207     \doedindexlabel%
7208     \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
7209     }%

```

```

7210 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,usedefault]{%
7211 \doedindexlabel%
7212 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
7213 }%
7214 }%
7215 \docsvlist{glsdisp,glslink}%
7216 %

```

Then those which arguments are [*options*]{*label*}.

```

7217 \renewcommand{\do}[1]{%
7218 \expandafter\DeclareRobustCommand\csname ed#1\endcsname[2][1,
usedefault]{%
7219 \doedindexlabel%
7220 \csname#1\endcsname[counter=pageline,##1]{##2}%
7221 }%
7222 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[2][1,usedefault]{%
7223 \doedindexlabel%
7224 \csname#1\endcsname*[counter=pageline,##1]{##2}%
7225 }%
7226 }%
7227 \docsvlist{glsadd}%
7228 }{}%
7229 }%
7230 %

```

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.

```

7231 \def\@hangingsymbol{}
7232 \newcommand*\sethangingsymbol[1]{%
7233 \gdef\@hangingsymbol{#1}%
7234 }%
7235 \newif\ifinstanza
7236 %

```

`\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.

```

7237 \newif\ifinserthangingsymbol
7238 \newcommand{\inserthangingsymbol}{%
7239 \ifinserthangingsymbol%
7240 \ifinstanza%
7241 \@hangingsymbol%
7242 \fi%
7243 \fi%
7244 }
7245 %

```

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.

```

7246 \newcommand*{\ampersand}{\char`\&}
7247
7248 %

```

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.

```

7249 \chardef\body=\catcode`\@
7250 \catcode`\@=11
7251 \chardef\next=\catcode`\&
7252 \catcode`\&=\active
7253
7254 %

```

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.

```

7255 \newcount\stanza@count
7256 \newlength{\stanzaindentbase}
7257 \setlength{\stanzaindentbase}{20pt}
7258
7259 %

```

`\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.

```

7260 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
7261 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,,|}%
7262   \stanza@count\z@
7263   \def\next{\expandafter\strip@szacnt\@tempa
7264     \ifx\@tempb\empty\let\next\relax\else
7265     \expandafter\mathchardef\csname #1\@number\stanza@count
7266     \@endcsname\@tempb\relax
7267     \advance\stanza@count\@ne\fi\next}%
7268   \next}
7269
7270 %

```

`\setstanzaindents` In the original edmac, `\setstanzavalues{sza}{⟨...⟩}` had to be called to set the in-
`\setstanzapenalties` `\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).

```

7271 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
7272 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
7273 %
7274 %

```

`\managestanza@modulo` Since version 0.13, the `stanzaindentsrepetition` counter can be used when the indentation is repeated every *n* verses. The `\managestanza@modulo` is a command which modifies the counter `stanza@modulo`. The command adds 1 to `stanza@modulo`, but if `stanza@modulo` is equal to the `stanzaindentsrepetition` counter, the command restarts it.

```

7275 \newcounter{stanzaindentsrepetition}
7276 \newcount\stanza@modulo
7277
7278 \newcommand*\managestanza@modulo[0]{%
7279   \advance\stanza@modulo\@ne%
7280   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
7281   \stanza@modulo\@ne%
7282   \fi%
7283 }
7284 %

```

`\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.

```

7285 \newcommand{\stanzaindent}[1]{%
7286   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
7287   \ignorespaces%
7288 }%
7289 \WithSuffix\newcommand\stanzaindent*[1]{%
7290   \stanzaindent{#1}%
7291   \global\advance\stanza@modulo-\@ne%
7292   \ifnum\stanza@modulo=0%
7293     \global\stanza@modulo=\value{stanzaindentrepetition}%
7294   \fi%
7295   \ignorespaces%
7296 }%
7297 %

```

XXVIII.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza \newcounter{stanza}
7299 \renewcommand{\thestanza}{%
7300   \textbf{\arabic{stanza}}%
7301 }
7302 %

```

`\ifnumberstanza` Then, macro to activate automatically numbering of stanza.

```

7303 \newif\ifnumberstanza%
7304 %

```

`\@insertstanzanumber` Now, macro called at the first line of verse of a stanza.

```

7305 \newcommand{\@insertstanzanumber}[0]{%
7306   \ifnumberstanza%
7307     \ifl@dpairing%
7308       \ifledRcol%
7309         \stanzanumwrapper{\thestanzaR}%
7310       \else%
7311         \stanzanumwrapper{\thestanzaL}%
7312       \fi%
7313     \else%
7314       \stanzanumwrapper{\thestanza}%
7315     \fi%
7316     \setline{1}%
7317   \fi%
7318 }%
7319 %

```

`\@advancestanzanumber` Also a command to advance the counter of stanza.

```

7320 \newcommand{\@advancestanzanumber}[0]{%
7321   \ifnumberstanza%
7322   \ifl@dpairing%
7323     \ifledRcol%
7324       \addtocounter{stanzaR}{1}%
7325     \else%
7326       \addtocounter{stanzaL}{1}%
7327     \fi%
7328   \else%
7329     \addtocounter{stanza}{1}%
7330   \fi%
7331 \fi%
7332 }%
7333 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

7334 \newcommand{\stanzanumwrapper}[1]{%
7335   \flagstanza{#1}%
7336 }%
7337 %

```

XXVIII.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

7338 \newcommand{\printstanza}[0]{%
7339   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
7340     l@dprintingcolumns}}{%
7341     \ifledRcol%
7342       \thestanzaR%
7343     \else%
7344       \thestanzaL%
7345     \fi%
7346   }{%
7347     \thestanza%
7348   }%
7349 %

```

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.

```

7350 \newcommandx{\stanza@line}[2][1,2,usedefault]{%
7351   \ifnum\value{stanzaindentrepetition}=0
7352     \ifcsdef{sza@\number\stanza@count @}{%
7353       {%
7354         \parindent=\csname sza@\number\stanza@count @\endcsname\
stanzaindentbase%
7355       }{%
7356         \led@err@StanzaIndentNotDefined%
7357       }%
7358     \else
7359       \ifcsdef{sza@\number\stanza@modulo @}{%
7360         \parindent=\csname sza@\number\stanza@modulo @\endcsname\
stanzaindentbase%
7361         \managestanza@modulo%
7362       }{%
7363         {%
7364           \led@err@StanzaIndentNotDefined%
7365         }%
7366       \fi
7367       \pstart[#1][#2]\stanza@hang\ignorespaces%
7368     }%
7369 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
7370   \hangindent\expandafter
7371   \noexpand\csname sza@00\endcsname\stanzaindentbase
7372   \hangafter\@ne}
7373 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
7374   \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
7375   \penalty\fi\count@}
7376 %

```

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 `\&`.

```

7377 \xdef\@startstanza[#1][#2]{%
7378   \noexpand\instanzatrue\expandafter
7379   \begingroup%
7380   \catcode`\noexpand\&\active%
7381   \global\stanza@count\@ne\stanza@modulo\@ne
7382   \noexpand\ifnum\expandafter\noexpand
7383   \csname sza@00\endcsname=\z@\let\noexpand\stanza@hang\relax
7384   \let\noexpand\endlock\relax\noexpand\else\interlinepenalty

```

```

7385 \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
7386 \expandafter\noexpand\csname szp@0@\endcsname=\z@
7387 \let\noexpand\sza@penalty\relax\noexpand\fi%
7388 \def\noexpand&{%
7389   \noexpand\newverse[] []}%
7390 \def\noexpand\&{\noexpand\@stopstanza}%
7391 \noexpand\@advancestanzanumber%
7392 \noexpand\stanza@line[#1][#2]%
7393 \noexpand\@insertstanzanumber%
7394 \let\par\relax\ignorespaces%No paragraph in verses
7395 }
7396
7397 \newcommandx{\stanza}[2][1,2,usedefault]{%
7398   \ifboolexpr{%
7399     not test{\ifdefvoid{\at@every@stanza}}%
7400     and test{\ifstrempy{#1}}%
7401     and test{\ifstrempy{#2}}}%
7402     {\@startstanza[] [\at@every@stanza] \at@start@every@stanza}%
7403     {\@startstanza[#1][#2] \at@start@every@stanza}%
7404   }%
7405
7406 \newcommandx{\@stopstanza}[2][1,2,usedefault]{%
7407   \unskip%
7408   \endlock%
7409   \ifboolexpr{%
7410     not test{\ifdefvoid{\at@every@stop@stanza}}%
7411     and test{\ifstrempy{#1}}%
7412     and test{\ifstrempy{#2}}}%
7413     {\before@every@stop@stanza\pend[] [\at@every@stop@stanza]}%
7414     {\before@every@stop@stanza\pend[#1][#2]}%
7415   \endgroup%
7416   \instanzafalse%
7417 }
7418
7419 \newcommand{\AtEveryStopStanza}[1]{%
7420   \ifstrempy{#1}%
7421     {\gdef\at@every@stop@stanza{}}%
7422     {\gdef\at@every@stop@stanza{\noindent#1}}%
7423 }%
7424 \WithSuffix\newcommand\AtEveryStopStanza*[1]{%
7425   \ifstrempy{#1}%
7426     {\gdef\at@every@stop@stanza{}}%
7427     {\gdef\at@every@stop@stanza{#1}}%
7428 }%
7429 \def\at@every@stop@stanza{%
7430
7431 \newcommand{\AtEveryStanza}[1]{%
7432   \ifstrempy{#1}%
7433     {\gdef\at@every@stanza{}}%
7434     {\gdef\at@every@stanza{\noindent#1}}%

```



```

7435 }%
7436 \WithSuffix\newcommand\AtEveryStanza*[1]{%
7437   \ifstrempy{#1}%
7438   {\gdef\at@every@stanza{}}%
7439   {\gdef\at@every@stanza{#1}}%
7440 }%
7441
7442
7443
7444 \newcommand{\AtStartEveryStanza}[1]{%
7445   \ifstrempy{#1}%
7446   {\gdef\at@start@every@stanza{}}%
7447   {\gdef\at@start@every@stanza{#1}}%
7448 }%
7449 \def\at@start@every@stanza{}%
7450
7451 \newcommand{\BeforeEveryStopStanza}[1]{%
7452   \ifstrempy{#1}%
7453   {\gdef\before@every@stop@stanza{}}%
7454   {\gdef\before@every@stop@stanza{#1}}%
7455 }%
7456 \def\before@every@stop@stanza{}%
7457
7458 \newcommandx*\newverse}[4][1,2,3,4,usedefault]{%
7459   \unskip%
7460   \endlock\pend[#1][#3]\sza@penalty\global%
7461   \advance\stanza@count\@ne\stanza@line[#2][#4]%
7462 }
7463
7464 %

```

\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`.

```

7465 \newcommand*\flagstanza}[2][\stanzaindentbase]{%
7466   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
7467
7468 %

```

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.

```

7469 \catcode`\&=\next

```

```

7470 \catcode`\@=\body
7471 \setstanzavalues{szp}{0}
7472
7473 %

```

XXIX Apparatus of Manuscripts

XXIX.1 User level macro

\msdata The user level **\msdata** command only writes the manuscripts data in numbered auxiliary file. There are two associated etoolbox counters.

```

\msdata@cR
\msdata@cR
7474 \def\msdata@c{}%
7475 \def\msdata@cR{}%
7476 \newcommand{\msdata}[1]{%
7477   \leavevmode%
7478   \unless\ifstopmsdata@inserted%
7479     \stopmsdata%
7480     \led@warning@msdatawithoutstop%
7481   \fi%
7482   \global\stopmsdata@inserted@false%
7483   \unless\ifledRcol%
7484     \numgdef{\msdata@c}{\msdata@c+1}%
7485     \ifdef{\hypertarget}{%
7486       \edlabel{\msdata@c:start:msdata}%
7487     }%
7488     \protected@write\linenum@out{}{%
7489       \string\@msd{#1}%
7490     }%
7491   \else%
7492     \numgdef{\msdata@cR}{\msdata@cR+1}%
7493     \ifdef{\hypertarget}{%
7494       \edlabel{\msdata@cR:start:msdata}%
7495     }%
7496     \protected@write\linenum@outR{}{%
7497       \string\@msd{#1}%
7498     }%
7499   \fi%
7500 }%
7501 %

```

\stopmsdata The user level **\stopmsdata** command only writes information about the end of manuscripts data in numbered auxiliary file.

```

7502 \newcommand{\stopmsdata}[0]{%
7503   \leavevmode%
7504   \unless\ifledRcol%
7505     \protected@write\linenum@out{}{%
7506       \string\@stopmsd%

```

```

7507 }%
7508 \ifdef{\hypertarget}{%
7509   \edlabel{\msdata@c:end:msdata}%
7510 }{}%
7511 \else%
7512   \protected@write\linenum@outR{}{%
7513     \string\@stopmsd%
7514   }%
7515   \ifdef{\hypertarget}{%
7516     \edlabel{\msdata@cR:end:msdata}%
7517   }{}%
7518   \fi%
7519   \global\stopmsdata@inserted@true%
7520 }%
7521 %

```

\ifstopmsdata@inserted@ The `\ifstopmsdata@inserted@` boolean is set to TRUE at every `\stopmsdata` and reset to FALSE at all `\msdata`. It also set to TRUE at every `\beginnumbering`. It is used to automatically insert `\stopmsdata` if forgotten before `\msdata`

```

7522 \newif\ifstopmsdata@inserted@%
7523 %

```

XXIX.2 Setting macro

Setting macros for the manuscripts apparatus tools is very easy: they just save their argument in an internal macro.

\setmsdataseries In which series of notes will be printed the apparatus of manuscripts?

```

7524 \newcommand{\setmsdataseries}[1]{%
7525   \gdef\@msdata@series{#1}%
7526 }%
7527 \def\@msdata@series{A}%
7528 %

```

\setmsdataposition The label for the manuscripts data.

```

7529 \def\ms@data@position{msdata-regular}%
7530 \newcommand{\setmsdataposition}[1]{%
7531   \gdef\ms@data@position{#1}%
7532 }%
7533 %

```

\setmsdatalabel The label for the manuscripts data.

```

7534 \def\ms@data@label{Ms.}%
7535 \newcommand{\setmsdatalabel}[1]{%
7536   \gdef\ms@data@label{#1}%
7537 }%
7538 %

```

XXIX.3 Counters and lists

\@msd@c \@msd@c is a counter incremented at each \@msd read in auxiliary file.

```
7539 \numdef{\@msd@c}{0}
7540 \numdef{\@msd@cR}{0}
7541 %
```

\add@msd@ \add@msd@ is a counter incremented at each \add@msddata, that is at each time we prepare the insertion of manuscripts data footnote.

```
7542 \numdef{\add@msd@c}{0}%
7543 \numdef{\add@msd@cR}{0}%
7544 %
```

\@msdata@list The \@msdata@list will contain, for each line, the lists of command to be executed to insert the manuscripts apparatus. It will be filled on \add@msdata and looped on \insert@msdata, then emptied.

```
7545 \def\@msdata@list{}%
7546 %
```

XXIX.4 Auxiliary file macros

\@msd The \@msd macro is written in the auxiliary file. It just defines three macros by \@msdata macro, which allow us to know the manuscripts data, the line number and the absolute line number where it was called

It also stores the action code 1010 in the list of actions by line.

```
7547 \newcommand{\@msd}[1]{%
7548   \unless\ifledRcol%
7549     \global\numdef{\@msd@c}{\@msd@c+\@ne}%
7550     \csgdef{\@msdata@\@msd@c @data}{#1}%
7551     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
7552     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
7553     \xright@appenditem{\the\absline@num}\to\actionlines@list%
7554     \xright@appenditem{-1010}\to\actions@list%
7555   \else%
7556     \global\numdef{\@msd@cR}{\@msd@cR+\@ne}%
7557     \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
7558     \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
7559     \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
7560     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
7561     \xright@appenditem{-1010}\to\actions@listR%
7562   \fi%
7563 }%
7564 %
```

\@stopmsd Inserted in the auxiliary file by \@stopmsd, the \@stopmsd macro will store in two commands the line number and the absolute line number on which it is called.

```

7565 \newcommand{\@stopmsd}[0]{%
7566   \unless\ifledRcol%
7567     \ifcsundef{@msdata@\@msd@c @stoplinenumber}{%
7568       \csxdef{@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
7569       \csxdef{@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
7570     }{}%
7571   \else%
7572     \ifcsundef{@msdata@\@msd@cR @stoplinenumberR}{%
7573       \csxdef{@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
7574       \csxdef{@msdata@\@msd@cR @stoplinenumberR}{\the\line@numR}%
7575     }%
7576   }{}%
7577   \fi%
7578 }%
7579 %

```

XXIX.5 Action macro

\add@msdata \add@msdata is executed on each line when action code 1010 is seen. It will not insert immediately the manuscript data footnote, as action code are executed before the line be typeset, and, consequently, could be on the previous page. So it just stores the manuscript data footnote to \@msdata@list.

```

7580 \newcommand{\add@msdata}{%
7581   \bgroup%
7582   \normalfont%
7583   \unless\ifledRcol%
7584     \numgdef{\add@msd@c}{\add@msd@c+\@ne}%
7585     \ifcsdef{@msdata@\add@msd@c @data}{%
7586       \letcs{@data}{@msdata@\add@msd@c @data}%
7587       \edef\l@d@nums{%
7588         000| % Start page = we don't print it
7589         \csuse{@msdata@\add@msd@c @linenumber}| % Start line number
7590         000| % Start subline number, for now, not used
7591         000| % End page number, we don't print it
7592         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
7593         {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
7594         {\csuse{@lastline@forpage@\the\page@num}}}%Otherwiser, last
number of the page
7595       |%
7596       000| % End sub line number, for now, not used
7597       \edfont@info%Font
7598     }%
7599     \@msd@options@fullpagefalse%
7600     \if@firstlineofpage%Try if the data are for the full page. If yes
, will add options to the list.
7601       \unless\if@msdata@insertedfrompreviouspage%

```

```

7602 \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse
{\@msdata@add@msd@c @stopabslinenumber}+\@ne}%
7603 {%
7604 \numdef{\@tmp}{\add@msd@c+\@ne}%
7605 \ifcsdef{\@msdata@\@tmp @abslinenumber}%
7606 {\ifnumequal{\csuse{\@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}}%
7607 {}%
7608 {\@msd@options@fullpagetrue}%
7609 }%
7610 {\@msd@options@fullpagetrue}%
7611 }%
7612 {}%
7613 \fi%
7614 \fi%
7615 \listxadd{\@msdata@list}{%
7616 \@msd@options@iffullpage%
7617 \ifluatex%
7618 \csxdef{footnote@luatextextdir}{the\textdir}%
7619 \csxdef{footnote@luatexpardir}{the\pardir}%
7620 \fi%
7621 \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
7622 \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
7623 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
7624 \reset@msd@options@iffullpage%
7625 }%
7626 }%
7627 {}%
7628 \else%
7629 \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
7630 \ifcsdef{\@msdata@\add@msd@cR @dataR}{%
7631 \letcs{\@data}{\@msdata@\add@msd@cR @dataR}%
7632 \edef\l@d@nums{%
7633 000| % Start page = we don't print it
7634 \csuse{\@msdata@\add@msd@cR @linenumberR}| % Start line number
7635 000| % Start subline number, for now, not used
7636 000| % End page number, we don't print it
7637 \ifnumless{\csuse{\@msdata@\add@msd@cR @stopabslinenumberR}}{\@
csuse{@lastline@forpageR@the\page@numR}}%
7638 {\csuse{\@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
7639 {\csuse{@lastline@forpageR@the\page@numR}}%Otherwiser, last
number of the page
7640 |%
7641 000| % End sub line number, for now, not used
7642 \edfont@info%Font
7643 }%
7644 \@msd@options@fullpagefalse%
7645 \if@firstlineofpageR%

```

```

7646         \unless\if@msdata@insertedfrompreviouspage%
7647         \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
csuse{@msdata@add@msd@c @stopabslinenumberR}+\@ne}%
7648         {%
7649         \numdef{\@tmp}{\add@msd@cR+\@ne}%
7650         \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7651         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
7652         {}%
7653         {\@msd@options@fullpagetrue}%
7654         }%
7655         {\@msd@options@fullpagetrue}%
7656         }%
7657         {}%
7658         \fi%
7659     \fi%
7660     \listxadd{\@msdata@list}{%
7661     \@msd@options@iffullpage%
7662     \ifluatex%
7663     \csxdef{footnote@luatextextdir}{\the\textdir}%
7664     \csxdef{footnote@luatexpardir}{\the\pardir}%
7665     \fi%
7666     \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
7667     \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
7668     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
7669     \reset@msd@options@iffullpage%
7670     }%
7671     }%
7672     {}%
7673     \fi%
7674     \egroup%
7675 }%
7676 %

```

\insertedfrompreviouspage The `\if@msdata@insertedfrompreviouspage` boolean is set to TRUE if `reledmac` automatically inserts data from previous page in the first line of a page.

```

7677 \newif\if@msdata@insertedfrompreviouspage%
7678 %

```

\add@msdata@firstlineofpage `\add@msdata@firstlineofpage` is called at the first line of every page. It inserts manuscript data which start on one of the previous pages and continue on this page.

```

7679 \newcommand{\add@msdata@firstlineofpage}{%
7680     \bgroup%
7681     \normalfont%
7682     \unless\ifledRcol%
7683     \ifcsdef{@msdata@\add@msd@c @data}{%

```

```

7684 \ifnumless{\the\absline@num-\@ne}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}}%
7685 {%
7686 \global\@msdata@insertedfrompreviouspagetrue%
7687 \letcs{\@data}{\@msdata@\add@msd@c \@data}%
7688 \edef\l@d@nums{%
7689 000|% Start page = we don't print it
7690 \numexpr\the\line@num+\@ne\relax|% Start line number = first line
of the page. As \add@msdata@firstlineofpage is called before line number
has been incremented, we increment it for printing
7691 000|% Start subline number, for now, not used
7692 000|% End page number, we don't print it
7693 \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
7694 {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
7695 {\csuse{@lastline@forpage@\the\page@num}}}%Otherwise, last
number of the page
7696 |%
7697 000|% End sub line number, for now, not used
7698 \edfont@info%Font
7699 }%
7700 \@msd@options@fullpagefalse%
7701 \ifnumless{\csuse{@lastabsline@forpage@\the\page@num}}{\csuse{
@msdata@\add@msd@c @stopabslinenumber}+\@ne}%We will test if the ms data is
for the full page
7702 {%
7703 \numdef{\@tmp}{\add@msd@c+\@ne}%
7704 \ifcsdef{@msdata@\@tmp @abslinenumber}%
7705 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
7706 {}%
7707 {\@msd@options@fullpagetrue}%
7708 }%
7709 {\@msd@options@fullpagetrue}%
7710 }%
7711 {}%
7712 \listxadd{\@msdata@list}{%
7713 \@msd@options@iffullpage%
7714 \ifluatex%
7715 \csxdef{footnote@luatextextdir}{\the\textdir}%
7716 \csxdef{footnote@luatexpardir}{\the\pardir}%
7717 \fi%
7718 \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
7719 \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
7720 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\msdata@label}{\expandonce\@data}}}%
7721 \reset@msd@options@iffullpage%
7722 }%
7723 }%

```



```

7724     {\global\@msdata@insertedfrompreviouspagefalse}%
7725   }{}%
7726   \else%
7727     \ifcsdef{@msdata@\add@msd@cR @dataR}{%
7728     \ifnumless{\the\absline@numR-\@ne}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}}%
7729     {%
7730     \global\@msdata@insertedfrompreviouspagetrue%
7731     \letcs{\@data}{@msdata@\add@msd@cR @dataR}%
7732     \edef\l@d@nums{%
7733     000}% Start page = we don't print it
7734     \numexpr\the\line@numR+\@ne\relax}% Start line number = first
line of the page. As \add@msdata@firstlineofpage is called before line
number has been incremented, we increment it for printing
7735     000}% Start subline number, for now, not used
7736     000}% End page number, we don't print it
7737     \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@\the\page@numR}}{%
7738     {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
7739     {\csuse{@lastline@forpageR@\the\page@numR}}}%Otherwise, last
number of the page
7740     |%
7741     000}% End sub line number, for now, not used
7742     \edfont@info%Font
7743   }%
7744   \@msd@options@fullpagefalse%
7745   \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@ne}%
7746   {%
7747   \numdef{\@tmp}{\add@msd@cR+\@ne}%
7748   \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7749   {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}}%
7750   {}%
7751   {\@msd@options@fullpagetrue}%
7752   }%
7753   {\@msd@options@fullpagetrue}%
7754   }%
7755   {}%
7756   \listxadd{\@msdata@list}{%
7757   \@msd@options@iffullpage%
7758   \ifluatex%
7759   \csxdef{footnote@luatextextdir}{\the\textdir}%
7760   \csxdef{footnote@luatexpardir}{\the\pardir}%
7761   \fi%
7762   \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
7763   \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
7764   \noexpand\csuse{v@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%

```

```

7765 \reset@msd@options@iffullpage%
7766 }%
7767 }%
7768 {\global\@msdata@insertedfrompreviouspagefalse}%
7769 }{}%
7770 \fi%
7771 \egroup%
7772 }%
7773 %

```

XXIX.6 Inserting footnote

Just before inserting standard insert (familiar and critical footnotes, sidenotes), we call `\insert@msdata` to insert manuscripts data's footnotes.

```

\insert@msdata 74 \newcommand{\insert@msdata}{%
7775 \def\do##1{##1}%
7776 \dolistloop{\@msdata@list}%
7777 \global\let\@msdata@list\relax%
7778 }%
7779 %

```

XXIX.7 Other

`\@msd@options@iffullpage` `\@msd@options@iffullpage` sets some options if the manuscripts data are for all the page. `\reset@msd@options@iffullpage` resets them after the footnote. `\if@msd@options@fullpage` is switch to true in `add@msdata@firstlineofpage` if these option must be inserted.

```

7780 \newif\if@msd@options@fullpage%
7781 \newcommand{\@msd@options@iffullpage}[0]{%
7782 \if@msd@options@fullpage%
7783 \noexpand\toggletrue{nonum@}%
7784 \ifdefvoid{\ms@data@label}%
7785 {\noexpand\toggletrue{nosep@}}%
7786 }%
7787 \fi%
7788 }%
7789 \newcommand{\reset@msd@options@iffullpage}[0]{%
7790 \noexpand\togglefalse{nonum@}%
7791 \noexpand\togglefalse{nosep@}%
7792 }%
7793 %

```

XXX Arrays and tables

XXX.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread ‘`eq and amstex`’, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsgen` package.

```
7794 \newtoks\@emptytoks
7795
7796 %
```

The rest is from `amsmath`.

`\l@denbody` A token register to contain the body.

```
7797 \newtoks\l@denbody
7798
7799 %
```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```
7800 \newcommand{\addtol@denbody}[1]{%
7801   \global\l@denbody\expandafter{\the\l@denbody#1}}
7802
7803 %
```

`\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`.

```
7804 \newcommand{\l@dcollect@body}[1]{%
7805   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
7806   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currentenv}}%
7807   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
7808   \begingroup
7809     \expandafter\let\csname\@currentenv\endcsname\l@dcollect@@body
7810     \edef\processl@denbody{\expandafter\noexpand\csname\@currentenv\endcsname}%
7811     \processl@denbody%
7812   }%
7813
7814 %
```

`\l@dpush@begins` When adding a piece of the current environment's contents to `\l@denvbody`, we scan it to check for additional `\begin` tokens, and add a 'b' to the stack for any that we find.

```

7815 \def\l@dpush@begins#1\begin#2{%
7816   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
7817
7818 %

```

`\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.

```

7819 \def\l@dcollect@@body#1\end#2{%
7820   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
7821     \expandafter\@gobble\l@dbegin@stack}%
7822   \ifx\@empty\l@dbegin@stack
7823     \endgroup
7824     \@checkend{#2}%
7825     \addtol@denvbody{#1}%
7826   \else
7827     \addtol@denvbody{#1\end{#2}}%
7828   \fi
7829   \processl@denvbody % A little tricky! Note the grouping
7830 }
7831
7832 %

```

There was a question on CTT about how to use `\collect@body` for a macro taking an argument. The following is part of that thread.

From: Heiko Oberdiek <oberdiek@uni-freiburg.de>
 Newsgroups: comp.text.tex
 Subject: Re: Using `\collect@body` with commands that take >1 argument
 Date: Fri, 08 Aug 2003 09:03:20 +0200

eed132@psu.edu (Evan) wrote:
 > I'm trying to make a new Latex environment that acts like the
 > `\colorbox` command that is part of the color package. I looked through
 > the FAQ and ran across this bit about using the `\collect@body` command
 > that is part of AMSLaTeX:
 > <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=cmdasenv>
 >
 > It almost works. If I do something like the following:
 > `\newcommand{\redbox}[1]{\colorbox{red}{#1}}`
 >
 > `\makeatletter`
 > `\newenvironment{redbox}{\collect@body \redbox}{}`

You will get an error message: Command `\redbox` already defined.
Thus you must rename either the command `\redbox` or the environment name.

```
> \begin{coloredbox}{blue}
>   Yadda yadda yadda... this is on a blue background...
> \end{coloredbox}
> and can't figure out how to make the \collect@body take this.

> \collect@body \colorbox{red}
> \collect@body {\colorbox{red}}
```

The argument of `\collect@body` has to be one token exactly.

```
\documentclass{article}
\usepackage{color}
\usepackage{amsmath}

\newcommand{\redbox}[1]{\colorbox{red}{#1}}
\makeatletter
\newenvironment{coloredbox}[1]{%
  \def\next@{\colorbox{#1}}%
  \collect@body\next@
}{%

% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}[1]{%
  \def\next@{\mycoloredbox{#1}}%
  \collect@body\next@
}{%
\newcommand{\mycoloredbox}[2]{%
  \colorbox{#1}{\ignorespaces#2\unskip}%
}

% support of optional color model argument
\newcommand\coloredboxIII\endcsname{}
\def\coloredboxIII#1#2{%
  \@coloredboxIII{#1}%
}
\def\@coloredboxIII#1#2{%
  \def\next@{\mycoloredboxIII{#1}{#2}}%
  \collect@body\next@
}
\newcommand{\mycoloredboxIII}[3]{%
  \colorbox{#1}{#2}{\ignorespaces#3\unskip}%
}

\makeatother
```

```

\begin{document}
  Black text before
  \begin{coloredbox}{blue}
    Hello World
  \end{coloredbox}
  Black text after

  Black text before
  \begin{coloredboxII}{blue}
    Hello World
  \end{coloredboxII}
  Black text after

  Black text before
  \begin{coloredboxIII}[rgb]{0,0,1}
    Hello World
  \end{coloredboxIII}
  Black text after

\end{document}

Yours sincerely
Heiko <oberdiedk@uni-freiburg.de>

```

XXX.2 Tabular environments

This is based on the work by Herbert Breger in developing `tabmac.tex`.

The original `tabmac.tex` file was void of comments or any explanatory text other than the above notice. The algorithm is Breger's. Peter Wilson have made some cosmetic changes to the original code and reimplemented some things so they are more LaTeX-like. All the commentary are from Peter Wilson, as are any mistake or errors.

However, Maïeul Rouquette has modified code in order to add new features of `eledmac` and `reledmac`.

XXX.2.1 Disabling and restoring commands

`\l@dtabnoexpands` More no expansion for critical and familiar footnotes in tabular environment.

```

7833 \newcommand*{\l@dtabnoexpands}{%
7834   \let\rtab=0%
7835   \let\ctab=0%
7836   \let\ltab=0%
7837   \let\rtabtext=0%
7838   \let\ltabtext=0%
7839   \let\ctabtext=0%
7840   \let\edbeforetab=0%
7841   \let\edaftertab=0%
7842   \let\edatleft=0%

```

```

7843 \let\edatright=0%
7844 \let\edvertline=0%
7845 \let\edvertdots=0%
7846 \let\edrowfill=0%
7847 }
7848
7849 %

```

\disable@familiarnotes Macros to disable and restore familiar notes, to prevent them from printing multiple times in edtabularx and edarrayx environments.

\restore@familiarnotes

```

7850 \newcommand{\disable@familiarnotes}{%
7851   \unless\ifnofamiliar{%
7852     \def\do##1{%
7853       \csletcs{footnote@##1}{footnote##1}%
7854       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
7855         \protected@csxdef{theftnmark##1}{\csuse{thefootnote##1}}%
7856         \csuse{footnotemark##1}%
7857       }%
7858     }%
7859     \dolistloop{\@series}%
7860   \fi%
7861 }%
7862 \newcommand{\restore@familiarnotes}{%
7863   \unless\ifnofamiliar{%
7864     \def\do##1{%
7865       \csletcs{footnote##1}{footnote@##1}%
7866     }%
7867     \dolistloop{\@series}%
7868   \fi%
7869 }%
7870
7871 %

```

\disable@sidenotes The same, for side notes.

\restore@sidenotes

```

7872 \newcommand{\disable@sidenotes}{%
7873   \let\@@ledrightnote\ledrightnote%
7874   \let\@@ledleftnote\ledleftnote%
7875   \let\@@ledsidenote\ledsidenote%
7876   \let\ledrightnote\@gobble%
7877   \let\ledleftnote\@gobble%
7878   \let\ledsidenote\@gobble%
7879 }%
7880 \newcommand{\restore@sidenotes}{%
7881   \let\ledrightnote\@@ledrightnote%
7882   \let\ledleftnote\@@ledleftnote%
7883   \let\ledsidenote\@@ledsidenote%
7884 }%
7885 %

```

\disable@notes Disable/restore side and familiar notes.

\restore@notes

```
7886 \newcommand{\disable@notes}{%
7887   \disable@sidenotes%
7888   \disable@familiarnotes%
7889 }%
7890 \newcommand{\restore@notes}{%
7891   \restore@sidenotes%
7892   \restore@familiarnotes%
7893 }%
7894 %
```

\EDTEXT We need to be able to modify the `\edtext` macros and also restore their original definitions.

\xedtext

```
7895 \let\EDTEXT=\edtext
7896 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
7897 %
```

\EDLABEL We need to be able to modify and restore the `\edlabel` macro.

\xedlabel

```
7898 \let\EDLABEL=\edlabel
7899 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}
7900 %
```

\xedindex Macros supporting modification and restoration of `\edindex`.

\nulledindex

```
7901 \AtBeginDocument{\let\xedindex\edindex}%
7902 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
7903
7904 %
```

\@line@num Macro supporting restoration of `\linenum`.

```
7905 \let\@line@num=\linenum
7906 %
```

\l@dgobblearg `\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```
7907 \newcommand*{\l@dgobbleoptarg}[2][\relax]%
7908
7909 %
```

\Relax `\let\Relax=\relax`

\NEXT `\let\NEXT=\next`

```
7912
7913 %
```


`\l@dmodforedtext` Modify and restore various macros for when `\edtext` is used.
`\l@drestoreforedtext`

```

7914 \newcommand{\l@dmodforedtext}{%
7915   \let\edtext\relax
7916   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
7917   \dolistloop{\@series}%
7918   \let\edindex\nulledindex
7919   \let\linenum@gobble}
7920 \newcommand{\l@drestoreforedtext}{%
7921   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
7922   \dolistloop{\@series}%
7923   \let\edindex\xedindex}
7924 %

```

`\l@dnnullfills` Nullify and restore some column fillers, etc.
`\l@drestorefills`

```

7925 \newcommand{\l@dnnullfills}{%
7926   \def\edlabel##1{%
7927     \def\edrowfill##1##2##3{%
7928     }
7929   \newcommand{\l@drestorefills}{%
7930     \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
7931   }
7932 }
7933 %

```

`\letsforverteilen` Gathers some lets and other code that is common to the `*verteilen*` macros.

```

7934 \newcommand{\letsforverteilen}{%
7935   \let\edtext\xedtext
7936   \let\edindex\xedindex
7937   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
7938   \dolistloop{\@series}%
7939   \let\linenum@line@num
7940   \hilfsskip=\l@dcolwidth%
7941   \advance\hilfsskip by -\wd\hilfsbox
7942   \def\edlabel##1{\xedlabel{##1}}
7943 }
7944 %

```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for
`\enablel@dtabfeet` `\edtext`.

```

7945 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
7946 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
7947 %

```

XXX.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.

```

7948 \newcount\l@dampcount
7949   \l@dampcount=1\relax
7950 \newcount\l@dcolcount
7951   \l@dcolcount=0\relax
7952
7953 %

```

`\hilfsbox` Some (temporary) helper items.

```

\hilfsskip
7954 \newbox\hilfsbox
\Hilfsbox
7955 \newskip\hilfsskip
\hilfscount
7956 \newbox\Hilfsbox
7957 \newcount\hilfscount
7958
7959 %

```

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).

```

7960 \newdimen\dcoli
7961 \newdimen\dcolii
7962 \newdimen\dcoliii
7963 \newdimen\dcoliv
7964 \newdimen\dcolv
7965 \newdimen\dcolvi
7966 \newdimen\dcolvii
7967 \newdimen\dcolviii
7968 \newdimen\dcolix
7969 \newdimen\dcolx
7970 \newdimen\dcolxi
7971 \newdimen\dcolxii
7972 \newdimen\dcolxiii
7973 \newdimen\dcolxiv
7974 \newdimen\dcolxv
7975 \newdimen\dcolxvi
7976 \newdimen\dcolxvii
7977 \newdimen\dcolxviii
7978 \newdimen\dcolxix
7979 \newdimen\dcolxx
7980 \newdimen\dcolxxi
7981 \newdimen\dcolxxii
7982 \newdimen\dcolxxiii
7983 \newdimen\dcolxxiv
7984 \newdimen\dcolxxv
7985 \newdimen\dcolxxvi
7986 \newdimen\dcolxxvii

```

```

7987 \newdimen\dcollxxviii
7988 \newdimen\dcollxxix
7989 \newdimen\dcollxxx
7990 \newdimen\dcollerr % added for error handling
7991
7992 %

```

\l@dcollwidth This is a cunning way of storing the column widths indexed by the column number \l@dcollcount, like an array. (was \Dimenzuordnung)

```

7993 \newcommand{\l@dcollwidth}{\ifcase \the\l@dcollcount \dcoli %???
7994 \or \dcoli \or \dcolii \or \dcoliii
7995 \or \dcoliv \or \dcolv \or \dcolvi
7996 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
7997 \or \dcolxi \or \dcolxii \or \dcolxiii
7998 \or \dcolxiv \or \dcolxv \or \dcolxvi
7999 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
8000 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
8001 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
8002 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
8003 \else \dcollerr \fi}
8004
8005 %

```

\step1@dcollcount This increments the column counter, and issues an error message if it is too large.

```

8006 \newcommand*{\step1@dcollcount}{\advance\l@dcollcount\@ne
8007 \ifnum\l@dcollcount>30\relax
8008 \led@err@TooManyColumns
8009 \fi}
8010
8011 %

```

\l@dsetmaxcolwidth Sets the column width to the maximum value seen so far.

```

8012 \newcommand{\l@dsetmaxcolwidth}{%
8013 \ifdim\l@dcollwidth < \wd\hilfsbox
8014 \l@dcollwidth = \wd\hilfsbox
8015 \else \relax \fi}
8016
8017 %

```

\measurecell Measure (recursively) the width required for a math cell.

```

8018 \def\measurecell #1&{%
8019 \ifx #1\ \ifnum\l@dcollcount=0\let\NEXT\relax%
8020 \else\l@dcheckcols%
8021 \l@dcollcount=0%
8022 \let\NEXT\measurecell%
8023 \fi%

```

```

8024 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8025 \step1@dcolcount%
8026 \l@dsetmaxcolwidth%
8027 \let\NEXT\measuremcell%
8028 \fi\NEXT}
8029
8030 %

```

\measuretcell Measure (recursively) the width required for a text cell.

```

8031 \def\measuretcell #1{%
8032 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8033 \else\l@dcheckcols%
8034 \l@dcolcount=0%
8035 \let\NEXT\measuretcell%
8036 \fi%
8037 \else\setbox\hilfsbox=\hbox{#1}%
8038 \step1@dcolcount%
8039 \l@dsetmaxcolwidth%
8040 \let\NEXT\measuretcell%
8041 \fi\NEXT}
8042
8043 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

8044 \def\measuremrow #1{%
8045 \ifx #1&\let\NEXT\relax%
8046 \else\measuremcell #1\&\&\&%
8047 \let\NEXT\measuremrow%
8048 \fi\NEXT}
8049 %

```

\measuretrow Measure (recursively) the width required for a text row.

```

8050 \def\measuretrow #1{%
8051 \ifx #1&\let\NEXT\relax%
8052 \else\measuretcell #1\&\&\&%
8053 \let\NEXT\measuretrow%
8054 \fi\NEXT}
8055
8056 %

```

\edtabcolsep The length \edtabcolsep controls the distance between columns.

```

8057 \newskip\edtabcolsep
8058 \global\edtabcolsep=10pt
8059
8060 %

```

```

\variab861 \newcommand{\variab}{\relax}
8062
8063 %

```

\l@dccheckcols Check that the number of columns is consistent.

```

8064 \newcommand*{\l@dccheckcols}{%
8065   \ifnum\l@dcolcount=1\relax
8066   \else
8067     \ifnum\l@dampcount=1\relax
8068     \else
8069       \ifnum\l@dcolcount=\l@dampcount\relax
8070       \else
8071         \l@d@err@UnequalColumns
8072       \fi
8073     \fi
8074     \l@dampcount=\l@dcolcount
8075   \fi}
8076
8077 %

```

\edfilldimen A length.

```

8078 \newdimen\edfilldimen
8079 \edfilldimen=0pt
8080
8081 %

```

\c@addcolcount A counter to hold the number of a column. We use a roman number so that we can grab the column dimension from `\dcol`. We do not use the `\roman` \TeX command, because some packages, like `babel` can override it in some specific cases (Greek, for example).

\theadcolcount

```

8082 \newcounter{addcolcount}
8083 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
8084 %

```

XXX.2.3 Tabular typesetting

\setmcellright Typeset (recursively) cells of display math right justified.

```

8085 \def\setmcellright #1&{\def\edlabel##1}%
8086   \let\edindex\nulledindex
8087   \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
8088     \let\Next\relax%
8089   \else\l@dcolcount=0%
8090     \let\Next=\setmcellright%
8091   \fi%
8092 \else%
8093   \disablel@dtabfeet%

```

```

8094         \stepl@dc colcount%
8095         \disable@notes%
8096         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8097         \restore@notes%
8098         \letsforverteilen%
8099         \hskip\hilfsskip$\displaystyle{#1}$%
8100         \hskip\edtabcolsep%
8101         \let\Next=\setmcellright%
8102     \fi\Next}
8103
8104 %

```

\settcellright Typeset (recursively) cells of text right justified.

```

8105 \def\settcellright #1&{\def\edlabel##1{}%
8106     \let\edindex\nulledindex
8107     \ifx #1\\ \ifnum\l@dc colcount=0%\removelastskip
8108         \let\Next\relax%
8109     \else\l@dc colcount=0%
8110         \let\Next=\settcellright%
8111     \fi%
8112 \else%
8113     \disablel@dtabfeet%
8114     \stepl@dc colcount%
8115     \disable@notes%
8116     \setbox\hilfsbox=\hbox{#1}%
8117     \restore@notes%
8118     \letsforverteilen%
8119     \hskip\hilfsskip#1%
8120     \hskip\edtabcolsep%
8121     \let\Next=\settcellright%
8122 \fi\Next}
8123 %

```

\setmcellleft Typeset (recursively) cells of display math left justified.

```

8124 \def\setmcellleft #1&{\def\edlabel##1{}%
8125     \let\edindex\nulledindex
8126     \ifx #1\\ \ifnum\l@dc colcount=0 \let\Next\relax%
8127         \else\l@dc colcount=0%
8128         \let\Next=\setmcellleft%
8129     \fi%
8130 \else \disablel@dtabfeet%
8131     \stepl@dc colcount%
8132     \disable@notes%
8133     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8134     \restore@notes%
8135     \letsforverteilen%
8136     $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
8137     \let\Next=\setmcellleft%

```

```

8138 \fi\Next}
8139
8140 %

```

\settcclleft Typeset (recursively) cells of text left justified.

```

8141 \def\settcclleft #1&{\def\edlabel##1{}}%
8142 \let\edindex\nulledindex
8143 \ifx #1\\ \ifnum\l@dcclcount=0 \let\Next\relax%
8144 \else\l@dcclcount=0%
8145 \let\Next=\settcclleft%
8146 \fi%
8147 \else \disablel@dtabfeet%
8148 \stepl@dcclcount%
8149 \disable@notes%
8150 \setbox\hilfsbox=\hbox{#1}%
8151 \restore@notes%
8152 \letsforverteilen%
8153 #1\hskip\hilfsskip\hskip\edtabcolsep%
8154 \let\Next=\settcclleft%
8155 \fi\Next}
8156 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

8157 \def\setmcellcenter #1&{\def\edlabel##1{}}%
8158 \let\edindex\nulledindex
8159 \ifx #1\\ \ifnum\l@dcclcount=0\let\Next\relax%
8160 \else\l@dcclcount=0%
8161 \let\Next=\setmcellcenter%
8162 \fi%
8163 \else \disablel@dtabfeet%
8164 \stepl@dcclcount%
8165 \disable@notes%
8166 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8167 \restore@notes%
8168 \letsforverteilen%
8169 \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
8170 \hskip\edtabcolsep%
8171 \let\Next=\setmcellcenter%
8172 \fi\Next}
8173
8174 %

```

\settcclcenter Typeset (recursively) cells of text centered.

```

8175 \def\settcclcenter #1&{\def\edlabel##1{}}%
8176 \let\edindex\nulledindex
8177 \ifx #1\\ \ifnum\l@dcclcount=0 \let\Next\relax%
8178 \else\l@dcclcount=0%

```

```

8179             \let\Next=\settcellcenter%
8180             \fi%
8181     \else    \disablel@dtabfeet%
8182             \stepl@dcolcount%
8183             \disable@notes%
8184             \setbox\hilfsbox=\hbox{#1}%
8185             \restore@notes%
8186             \letsforverteilen%
8187             \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
8188             \hskip\edtabcolsep%
8189             \let\Next=\settcellcenter%
8190     \fi\Next}
8191
8192 %

```

```

\NEXT93 \let\NEXT=\relax

```

```

8194
8195 %

```

\setmrowright Typeset (recursively) rows of right justified math.

```

8196 \def\setmrowright #1\{%
8197     \ifx #1& \let\NEXT\relax
8198     \else \centerline{\setmcellright #1&\\&\\&}
8199     \let\NEXT=\setmrowright
8200     \fi\NEXT}
8201 %

```

\settroright Typeset (recursively) rows of right justified text.

```

8202 \def\settroright #1\{%
8203     \ifx #1& \let\NEXT\relax
8204     \else \centerline{\settcellright #1&\\&\\&}
8205     \let\NEXT=\settroright
8206     \fi\NEXT}
8207
8208 %

```

\setmrowleft Typeset (recursively) rows of left justified math.

```

8209 \def\setmrowleft #1\{%
8210     \ifx #1&\let\NEXT\relax
8211     \else \centerline{\setmcellleft #1&\\&\\&}
8212     \let\NEXT=\setmrowleft
8213     \fi\NEXT}
8214 %

```

\settrorleft Typeset (recursively) rows of left justified text.


```

8215 \def\settrorleft #1\{ %
8216   \ifx #1& \let\NEXT\relax
8217   \else \centerline{\settcclleft #1&\&\&}
8218         \let\NEXT=\settrorleft
8219   \fi\NEXT}
8220
8221 %

```

\setmrowcenter Typeset (recursively) rows of centered math.

```

8222 \def\setmrowcenter #1\{ %
8223   \ifx #1& \let\NEXT\relax %
8224   \else \centerline{\setmcclcenter #1&\&\&}
8225         \let\NEXT=\setmrowcenter
8226   \fi\NEXT}
8227 %

```

\settrorcenter Typeset (recursively) rows of centered text.

```

8228 \def\settrorcenter #1\{ %
8229   \ifx #1& \let\NEXT\relax
8230   \else \centerline{\settcclcenter #1&\&\&}
8231         \let\NEXT=\settrorcenter
8232   \fi\NEXT}
8233
8234 %

```

\nullsetzen 35 \newcommand{\nullsetzen}{ %

```

8236   \step1@dcolcount %
8237   \l@dcolwidth=0pt %
8238   \ifnum\l@dcolcount=30\let\NEXT\relax %
8239         \l@dcolcount=0\relax
8240   \else\let\NEXT\nullsetzen %
8241   \fi\NEXT}
8242
8243 %

```

\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.

```

8244 \newcommand{\edatleft}[3][\@empty]{ %
8245   \ifx#1\@empty
8246     \vbox to 10pt{\vss\hbox{\$ \left#2\vrule width0pt height #3
8247                               depth 0pt \right. \$\hss}\vfil}
8248   \else
8249     \vbox to 4pt{\vss\hbox{\$#1\left#2\vrule width0pt height #3
8250                               depth 0pt \right. \$}\vfil}
8251   \fi}
8252 %

```

\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.

```

8253 \newcommand{\edatright}[3][\@empty]{%
8254   \ifx#1\@empty
8255     \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
8256               depth 0pt \right#2 $\hss}\vfil}
8257   \else
8258     \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
8259               depth 0pt \right#2 #1 $\vfil}
8260   \fi}
8261
8262 %

```

\edvertline `\edvertline{ $\langle len \rangle$ }` vertical line $\langle len \rangle$ high.

```

8263 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
8264
8265 %

```

\edvertdots `\edvertdots{ $\langle len \rangle$ }` vertical dotted line $\langle len \rangle$ high.

```

8266 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
8267   {\cleaders\hbox{$\math\hbox{.}\vbox to 0.5em{ }$}\vfil}}}
8268
8269 %

```

\l@dtabaddcols `\l@dtabaddcols{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ }` adds the widths of the columns $\langle startcol \rangle$ through $\langle endcol \rangle$ to `\edfilldimen`. It is a \LaTeX style reimplementation of the original `\@add@`.

```

8270 \newcommand{\l@dtabaddcols}[2]{%
8271   \l@dcheckstartend{#1}{#2}%
8272   \ifl@dstartendok
8273     \setcounter{addcolcount}{#1}%
8274     \@whilenum \value{addcolcount}<#2\relax \do
8275     {\advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
8276     \advance\edfilldimen by \edtabcolsep
8277     \stepcounter{addcolcount}}%
8278     \advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
8279   \fi
8280 }
8281
8282 %

```

\ifl@dstartendok `\l@dcheckstartend{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ }` checks that the values of $\langle startcol \rangle$ and $\langle endcol \rangle$ are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```

8283 \newif\ifl@dstartendok
8284 \newcommand{\l@dcheckstartend}[2]{%

```

```

8285 \l@dstartendoktrue
8286 \ifnum #1<\@ne
8287   \l@dstartendokfalse
8288   \led@err@LowStartColumn
8289 \fi
8290 \ifnum #2>30\relax
8291   \l@dstartendokfalse
8292   \led@err@HighEndColumn
8293 \fi
8294 \ifnum #1>#2\relax
8295   \l@dstartendokfalse
8296   \led@err@ReverseColumns
8297 \fi
8298 }
8299
8300 %

```

`\edrowfill` `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a \LaTeX style reimplementation and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktel` macros.

```

8301 \newcommand*{\edrowfill}[3]{%
8302   \l@dtabaddcols{#1}{#2}%
8303   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
8304 \let\@edrowfill=\edrowfill
8305 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
8306
8307 %

```

`\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`.

```

8308 \newcommand{\leftltab}[1]{%
8309   \hb@xt@ \z@{\vbox{\edtabindent%
8310     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8311
8312 %

```

`\leftrtab` `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

8313 \newcommand{\leftrtab}[2]{%
8314   #2\hb@xt@ \z@{\vbox{\edtabindent%
8315     \advance\Hilfsskip by\dcoli%
8316     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8317
8318 %

```

`\leftctab` `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

8319 \newcommand{\leftctab}[2]{%
8320     \hb@xt@#2{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8321     \advance\Hilfsskip by 0.5\dcoli%
8322     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8323     \disablel@dtabfeet$\displaystyle{#2}$}%
8324     \advance\Hilfsskip by -0.5\wd\hilfsbox%
8325     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
8326     #2}
8327
8328 %

```

`\rightctab` `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

8329 \newcommand{\rightctab}[2]{%
8330     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8331     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8332     #1\hb@xt@#2{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8333     \advance\Hilfsskip by 0.5\l@dcolwidth%
8334     \advance\Hilfsskip by -\wd\hilfsbox%
8335     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8336     \disablel@dtabfeet$\displaystyle{#1}$}%
8337     \advance\Hilfsskip by -0.5\wd\hilfsbox%
8338     \advance\Hilfsskip by \edtabcolsep%
8339     \moveright\Hilfsskip\hbox{ #2}}\hss}%
8340     }
8341
8342 %

```

`\rightltab` `\rightltab{<math>}{<text>}` for `\edaftertab` in `\ltab`.

```

8343 \newcommand{\rightltab}[2]{%
8344     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8345     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8346     #1\hb@xt@#2{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8347     \advance\Hilfsskip by\l@dcolwidth%
8348     \advance\Hilfsskip by-\wd\hilfsbox%
8349     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8350     \disablel@dtabfeet$\displaystyle{#1}$}%
8351     \advance\Hilfsskip by-\wd\hilfsbox%
8352     \advance\Hilfsskip by\edtabcolsep%
8353     \moveright\Hilfsskip\hbox{ #2}}\hss}%
8354     }
8355
8356 %

```

`\rightrtab` `\rightrtab{<math>}{<text>}` for `\edaftertab` in `\rtab`.

```

8357 \newcommand{\rightrtab}[2]{%

```

```

8358 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
8359 \disablel@dtabfeet#2}%
8360 #1\hb@xt@{z@{\vbox{\edtabindent%
8361 \advance\Hilfsskip by-\wd\hilfsbox%
8362 \advance\Hilfsskip by\edtabcolsep%
8363 \moveright\Hilfsskip\hbox{ #2}}\hss}%
8364 }
8365
8366 %

```

\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.

```

8367 \newcommand{\rtab}[1]{%
8368 \l@dnnullfills
8369 \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
8370 \def\edaftertab##1##2{\righttab{##1}{##2}}%
8371 \measurebody{#1}%
8372 \l@drestorefills
8373 \variab
8374 \setmrowright #1\&\&\&%
8375 \enablel@dtabfeet}
8376
8377 %

```

\measurebody `\measurebody{<body>}` measures the array `<body>`.

```

8378 \newcommand{\measurebody}[1]{%
8379 \disablel@dtabfeet%
8380 \l@dcolcount=0%
8381 \nullsetzen%
8382 \l@dcolcount=0
8383 \measuremrow #1\&\&\&%
8384 \global\l@dampcount=1}
8385
8386 %

```

\rtabtext `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```

8387 \newcommand{\rtabtext}[1]{%
8388 \l@dnnullfills
8389 \measuretbody{#1}%
8390 \l@drestorefills
8391 \variab
8392 \setthrowright #1\&\&\&%
8393 \enablel@dtabfeet}
8394
8395 %

```

`\measuretbody` `\measuretbody{<body>}` measures the tabular <body>.

```

8396 \newcommand{\measuretbody}[1]{%
8397   \disable@notes%
8398   \disablel@dtabfeet%
8399   \l@dcolcount=0%
8400   \nullsetzen%
8401   \l@dcolcount=0
8402   \measuretrrow #1\\&\\%
8403   \restore@notes%
8404   \global\l@dampcount=1}
8405
8406 %

```

`\ltab` Array with entries left justified.

```

\edbeforetab
\edaftertab
8407 \newcommand{\ltab}[1]{%
8408   \l@dnnullfills
8409   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
8410   \def\edaftertab##1##2{\rightltab{##1}{##2}}%
8411   \measuretbody{#1}%
8412   \l@drestorefills
8413   \variab
8414   \setmrowleft #1\\&\\%
8415   \enablel@dtabfeet}
8416
8417 %

```

`\ltabtext` Tabular with entries left justified.

```

8418 \newcommand{\ltabtext}[1]{%
8419   \l@dnnullfills
8420   \measuretbody{#1}%
8421   \l@drestorefills
8422   \variab
8423   \settrrowleft #1\\&\\%
8424   \enablel@dtabfeet}
8425
8426 %

```

`\ctab` Array with centered entries.

```

\edbeforetab
\edaftertab
8427 \newcommand{\ctab}[1]{%
8428   \l@dnnullfills
8429   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
8430   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
8431   \measuretbody{#1}%
8432   \l@drestorefills
8433   \variab
8434   \setmrowcenter #1\\&\\%

```

```

8435 \enablel@dtabfeet}
8436
8437 %

```

\ctabtext Tabular with entries centered.

```

8438 \newcommand{\ctabtext}[1]{%
8439 \l@dnnullfills
8440 \measuretbody{#1}%
8441 \l@drestorefills
8442 \variab
8443 \settrrowcenter #1\\&\\%
8444 \enablel@dtabfeet}
8445
8446 %

```

\spreadtext₄₇ \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
\hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
%
8449 %

\spreadmath₅₀ \newcommand{\spreadmath}[1]{%
\hb@xt@ \the\l@dcolwidth{\hbox{\$\displaystyle{#1}\$}\hss}}
%
8453 %

\HILFSskip More helpers.

\Hilfsskip
8454 \newskip\HILFSskip
8455 \newskip\Hilfsskip
8456
8457 %

\EDTABINDENT₅₈ \newcommand{\EDTABINDENT}{%
8459 \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
8460 \else\step\l@dcolcount%
8461 \advance\Hilfsskip by\l@dcolwidth%
8462 \ifdim\l@dcolwidth=0pt\advance\hilfscount\@ne
8463 \else\advance\Hilfsskip by \the\hilfscount\edtabcolsep%
8464 \hilfscount=1\fi%
8465 \let\NEXT=\EDTABINDENT%
8466 \fi\NEXT}%
8467 %

\edtabindent (was \tabindent)

```

8468 \newcommand{\edtabindent}{%
8469     \l@dcollectioncount=0\relax
8470     \Hilfsskip=0pt%
8471     \hilfscount=1\relax
8472     \EDTABINDENT%
8473     \hilfsskip=\hsize%
8474     \advance\hilfsskip -\Hilfsskip%
8475     \Hilfsskip=0.5\hilfsskip%
8476 }%
8477
8478 %

```

\EDTAB (was \TAB)

```

8479 \def\EDTAB #1|#2|{%
8480     \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
8481     \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
8482     \advance\tabelskip -\wd\tabhilfbox%
8483     \advance\tabelskip -\wd\tabHilfbox%
8484     \unhbox\tabhilfbox\hskip\tabelskip%
8485     \unhbox\tabHilfbox}%
8486
8487 %

```

\EDTABtext (was \TABtext)

```

8488 \def\EDTABtext #1|#2|{%
8489     \setbox\tabhilfbox=\hbox{#1}%
8490     \setbox\tabHilfbox=\hbox{#2}%
8491     \advance\tabelskip -\wd\tabhilfbox%
8492     \advance\tabelskip -\wd\tabHilfbox%
8493     \unhbox\tabhilfbox\hskip\tabelskip%
8494     \unhbox\tabHilfbox}%
8495 %

```

\tabhilfbox Further helpers.

\tabHilfbox

```

8496 \newbox\tabhilfbox
8497 \newbox\tabHilfbox
8498
8499 %

```

XXX.2.4 Environments

edarrayl edarrayc edarrayr The ‘environment’ forms for **\ltab**, **\ctab** and **\rtab**.

```

8500 \newenvironment{edarrayl}{\l@dcollection@body\ltab}{\}
8501 \newenvironment{edarrayc}{\l@dcollection@body\ctab}{\}
8502 \newenvironment{edarrayr}{\l@dcollection@body\rtab}{\}
8503
8504 %

```


edtabularl edtabularc edtabularr The ‘environment’ forms for \ltabtext, \ctabtext and \rtabtext.

```

8505 \newenvironment{edtabularl}{\l@collect@body\ltabtext}{\}
8506 \newenvironment{edtabularc}{\l@collect@body\ctabtext}{\}
8507 \newenvironment{edtabularr}{\l@collect@body\rtabtext}{\}
8508
8509 %

```

XXXI Quotation’s commands

`\initnumbering@quote` This macro, called at the beginning of any numbered section, locally redefines the quotation and quote environments, in order to allow their use inside of numbered sections.

```

\quotation \initnumbering@quote defines quotation environment.
\endquotation
\quote      8510 \newcommand{\initnumbering@quote}{\}
\endquote   8511 \ifnoquotation@else
            8512 \renewcommand{\quotation}{\par\leavevmode%
            \parindent=1.5em%
            \skipnumbering%
            \ifautopar%
            \vskip-\parskip%
            \else%
            \vskip\topsep%
            \fi%
            \global\leftskip=\leftmargin%
            \global\rightskip=\leftmargin%
            }
            \renewcommand{\endquotation}{\par%
            \global\leftskip=0pt%
            \global\rightskip=0pt%
            \leavevmode%
            \skipnumbering%
            \ifautopar%
            \vskip-\parskip%
            \else%
            \vskip\topsep%
            \fi%
            }
            \renewcommand{\quote}{\par\leavevmode%
            \parindent=0pt%
            \skipnumbering%
            \ifautopar%
            \vskip-\parskip%
            \else%
            \vskip\topsep%
            \fi%
            }

```

```

8542             \global\leftskip=\leftmargin%
8543             \global\rightskip=\leftmargin%
8544         }
8545         \renewcommand{\endquote}{\par%
8546             \global\leftskip=0pt%
8547             \global\rightskip=0pt%
8548             \leavevmode%
8549             \skipnumbering%
8550             \ifautopar%
8551                 \vskip-\parskip%
8552             \else%
8553                 \vskip\topsep%
8554             \fi%
8555         }
8556     \fi
8557 }
8558 %

```

XXXII Section's title commands

XXXII.1 Commands to disable some feature

\ledsectnotoc The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```

8559 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
8560 %

```

\ledsectnomark The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```

8561 \newcommand{\ledsectnomark}{%
8562     \let\chaptermark\@gobble%
8563     \let\sectionmark\@gobble%
8564     \let\subsectionmark\@gobble%
8565 }
8566 %

```

XXXII.2 General overview

The system of `\eledxxxx` commands to section text work like this:

1. When one of these commands is called, `reledmac` writes to an auxiliary files:
 - The section level.
 - The section title.
 - The side (when `eledpar` is used).
 - The `pstart` where the command is called.

- If we have starred version or not.
2. `reledmac` adds the title of the section to `pstart`, as normal content. This is to enable critical notes.
 3. When \TeX is run a other time, this file is read. That:
 - Adds the `pstart` number to a list of `pstarts` where a sectioning command is used.
 - Defines a command, the name of which contains the `pstart` number, and which calls the normal \TeX sectioning command.
 4. This last command is called when the `pstart` is effectively printed.

XXXII.3 `\beforeeledchapter` command

We do not define commands for `\eledsection` and related if the `noeledsec` option is loaded. We use `etoolbox` tests and not the `\ifxxx...\else...\fi` structure to prevent problem of expansions with command after the `\ifxxx` which contains `\fi`. As we patch command inside this test, we need to change the category code of `#` character *before* `\notbool` statement, because the second argument is read with the standard catcode (read *The TeXbook* to understand when the catcode's change has effect).

```
8567 \catcode`\#=12
8568 \notbool{@noeled@sec}{%
8569 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
8570 \ifl@dmemoir
8571 \newcommand\beforeeledchapter{%
8572 \clearforchapter%
8573 }
8574 \else
8575 \newcommand\beforeeledchapter{%
8576 \if@openright%
8577 \cleardoublepage%
8578 \else%
8579 \clearpage%
8580 \fi%
8581 }
8582 \fi
8583 %
```

XXXII.4 Auxiliary commands

`\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.

```

8584 \def\print@rightmargin@eledsection{%
8585   \if@eled@sectioning%
8586     \begingroup%
8587     \if@RTL%
8588       \let\llap\rlap%
8589       \let\leftlinenum\rightlinenum%
8590       \let\leftlinenumR\rightlinenumR%
8591       \let\l@drd@ta\l@dld@ta%
8592       \let\l@drsn@te\l@dlsn@te%
8593     \fi%
8594     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
8595     \endgroup%
8596   \fi%
8597 }%
8598
8599 \def\print@leftmargin@eledsection{%
8600   \if@eled@sectioning%
8601     \leavevmode%
8602     \begingroup%
8603     \if@RTL%
8604       \let\rlap\llap%
8605       \let\rightlinenum\leftlinenum%
8606       \let\rightlinenumR\leftlinenumR%
8607       \let\l@dld@ta\l@drd@ta%
8608       \let\l@dlsn@te\l@drsn@te%
8609     \fi%
8610     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
8611     \endgroup%
8612   \fi%
8613 }%
8614
8615 %

```

XXXII.5 Patching standard commands

`\M@sect`
`\@mem@old@ssect`
`\@makechapterhead`
`\@makechapterhead`
`\@makeschapterhead`
`\@sect`
`\@ssect`

We have to patch \LaTeX , book and memoir sectioning commands in order to:

- Disable `\edtext` inside.
- Disable page breaking (for `\chapter`).
- Add line numbers and sidenotes.

Unfortunately, Maïeul Rouquette was not able to try if memoir is loaded. That is why `eledmac` tries to define for both standard class and memoir class.

```

8616 \AtBeginDocument{%
8617
8618
8619 \pretocmd{\M@sect}

```

```

8620 {\let\old@edtext=\edtext%
8621 \let\edtext=\dummy@edtext@showlemma%
8622 }
8623 {}
8624 {}
8625
8626 \apptocmd{\M@sect}
8627 {\let\edtext=\old@edtext}
8628 {}
8629 {}
8630
8631 \patchcmd{\M@sect}
8632 { #9}
8633 { #9%
8634 \print@rightmargin@eledsection%
8635 }
8636 {}
8637 {}
8638
8639 \patchcmd{\M@sect}
8640 {\hskip #3\relax}
8641 {\hskip #3\relax%
8642 \print@leftmargin@eledsection%
8643 }
8644 {}
8645 {}
8646
8647 \patchcmd{\@mem@old@ssect}
8648 {#5}
8649 {#5%
8650 \print@leftmargin@eledsection%
8651 }
8652 {}
8653 {}
8654
8655 \patchcmd{\@mem@old@ssect}
8656 {\hskip #1}
8657 {\hskip #1%
8658 \print@rightmargin@eledsection%
8659 }
8660 {}
8661 {}
8662
8663
8664
8665 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8666 \if@eled@sectioning\else%
8667 \ifl@dprintingpages\else%
8668 \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a

```

```

\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
8669   \fi%
8670   \fi%
8671   }
8672   {}
8673   {}
8674
8675 \patchcmd{\@makechapterhead}
8676   {#1}
8677   {\print@leftmargin@eledsection%
8678     #1%
8679     \print@rightmargin@eledsection%
8680   }
8681   {}
8682   {}
8683
8684 \patchcmd{\@makechapterhead}% For BIDI
8685   {\if@RTL\raggedleft\else\raggedright\fi}%
8686   {\if@eled@sectioning\else%
8687     \if@RTL\raggedleft\else\raggedright\fi%
8688   \fi%
8689   }%
8690   {}%
8691   {}%
8692
8693 \patchcmd{\@makeschapterhead}
8694   {#1}
8695   {\print@leftmargin@eledsection%
8696     #1%
8697     \print@rightmargin@eledsection%
8698   }
8699   {}
8700   {}
8701
8702 \pretocmd{\@sect}
8703   {\let\old@edtext=\edtext
8704     \let\edtext=\dummy@edtext@showlemma%
8705   }
8706   {}
8707   {}
8708
8709 \apptocmd{\@sect}
8710   {\let\edtext=\old@edtext}
8711   {}
8712   {}
8713
8714 \pretocmd{\@ssect}
8715   {\let\old@edtext=\edtext%
8716     \let\edtext=\dummy@edtext@showlemma%

```

```

8717 }
8718 {}
8719 {}
8720
8721 \apptocmd{\@ssect}
8722 {\let\edtext=\old@edtext}
8723 {}
8724 {}
8725
8726 %

```

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.

```

8727 \@ifpackageloaded{nameref}{
8728
8729   \patchcmd{\NR@sect}
8730     {#8}
8731     {#8%
8732       \print@rightmargin@eledsection%
8733     }
8734     {}
8735     {}
8736
8737   \patchcmd{\NR@sect}
8738     {\hskip #3\relax}
8739     {\hskip #3\relax%
8740       \print@leftmargin@eledsection%
8741     }
8742     {}
8743     {}
8744
8745   \patchcmd{\NR@ssect}
8746     {#5}
8747     {#5%
8748       \print@rightmargin@eledsection%
8749     }
8750     {}
8751     {}
8752
8753   \patchcmd{\NR@ssect}
8754     {\hskip #1}
8755     {\hskip #1%
8756       \print@leftmargin@eledsection%
8757     }
8758     {}
8759     {}
8760   }%
8761   {
8762     \patchcmd{\@sect}
8763       {#8}

```

```

8764     {#8%
8765     \print@rightmargin@eledsection%
8766     }
8767     {}
8768     {}
8769
8770     \patchcmd{\@sect}
8771     {\hskip #3\relax}
8772     {\hskip #3\relax%
8773     \print@leftmargin@eledsection%
8774     }
8775     {}
8776     {}
8777
8778     \patchcmd{\@ssect}
8779     {#5}
8780     {#5%
8781     \print@rightmargin@eledsection%
8782     }
8783     {}
8784     {}
8785
8786     \patchcmd{\@ssect}
8787     {\hskip #1}
8788     {\hskip #1%
8789     \print@leftmargin@eledsection%
8790     }
8791     {}
8792     {}
8793     }%
8794 }%
8795 %

```

Close the `\notbool{@noeled@sec}` statement. Also, we have finished patching the commands, using `#` with a catcode equal to 12, so we are restoring the normal catcode for `#`.

```

8796 {}}%
8797 \protect\catcode`\#=6 %Space NEEDS by \catcode
8798 %

```

\chapter We patch the `\chapter` command even if the `noeledsec` option is called, because we can use `\chapter` in the optional argument of a `\pstart` in parallel typesetting.

```

8799 \AtBeginDocument{%
8800 \patchcmd{\chapter}{\clearforchapter}{%
8801 \if@eled@sectioning\else%
8802 \ifl@dprintingpages\else%
8803 \clearforchapter%
8804 \fi%
8805 \fi%

```



```

8806 }%
8807 {}%
8808 {}%
8809
8810 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8811   \if@eled@sectioning\else%
8812     \ifl@dprintingpages%
8813       \endgraf%
8814     \else%
8815       \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
8816   \fi%
8817 \fi%
8818 }%
8819 {}%
8820 {}%
8821 }%
8822 %

```

\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.

```

8823 \newif\if@eled@sectioning%
8824 %

```

We reopen a new \notbool{@noeled@sec} statement, as we will define the \elesection commands.

```

8825 \notbool{@noeled@sec}{%
8826 %

```

XXXII.6 Main code of \eledxxx commands

\eled@sectioning@out \eled@sectioning@out is the output file, to dump the pstarts where a sectioning command is used.

```

8827 \newwrite\eled@sectioning@out
8828 %

```

\eledchapter And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsection
\eledsubsection
8829 \eledsubsubsection \newcommand{\eledchapter}[2][{}]{%
8830   \disable@familiarnotes%
\eledchapter* 8831   #2%
\eledsection* 8832   \restore@familiarnotes%
\eledsubsection* 8833   \ifledRcol%
\eledsubsubsection*

```

```

8834 \immediate\write\eled@sectioningR@out{%
8835 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
8836 }%
8837 \else%
8838 \immediate\write\eled@sectioning@out{%
8839 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{L}
8840 }%
8841 \fi%
8842 }
8843
8844 \newcommand{\eledsection}[2][]{%
8845 \disable@familiarnotes%
8846 #2%
8847 \restore@familiarnotes%
8848 \ifledRcol%
8849 \immediate\write\eled@sectioningR@out{%
8850 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
8851 }%
8852 \else%
8853 \immediate\write\eled@sectioning@out{%
8854 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{L}
8855 }%
8856 \fi%
8857 }
8858
8859 \newcommand{\eledsubsection}[2][]{%
8860 \disable@familiarnotes%
8861 #2%
8862 \restore@familiarnotes%
8863 \ifledRcol%
8864 \immediate\write\eled@sectioningR@out{%
8865 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
8866 }%
8867 \else%
8868 \immediate\write\eled@sectioning@out{%
8869 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{L}
8870 }%
8871 \fi%
8872 }
8873 \newcommand{\eledsubsubsection}[2][]{%
8874 \disable@familiarnotes%
8875 #2%
8876 \restore@familiarnotes%
8877 \ifledRcol%
8878 \immediate\write\eled@sectioningR@out{%
8879 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
8880 }{R}
8881 }%
8882 \else%

```

```

8882 \immediate\write\eled@sectioning@out{%
8883 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL
8884 }{}{}
8885 }%
8886 \fi%
8887 }
8888
8889 \WithSuffix\newcommand\eledchapter*[2][]{%
8890 \disable@familiarnotes%
8891 #2%
8892 \restore@familiarnotes%
8893 \ifledRcol%
8894 \immediate\write\eled@sectioningR@out{%
8895 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8896 }%
8897 \else%
8898 \immediate\write\eled@sectioning@out{%
8899 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{R}
8900 }%
8901 \fi%
8902 }
8903
8904 \WithSuffix\newcommand\eledsection*[2][]{%
8905 \disable@familiarnotes%
8906 #2%
8907 \restore@familiarnotes%
8908 \ifledRcol%
8909 \immediate\write\eled@sectioningR@out{%
8910 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8911 }%
8912 \else%
8913 \immediate\write\eled@sectioning@out{%
8914 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{R}
8915 }%
8916 \fi%
8917 }
8918
8919 \WithSuffix\newcommand\eledsubsection*[2][]{%
8920 \disable@familiarnotes%
8921 #2%
8922 \restore@familiarnotes%
8923 \ifledRcol%
8924 \immediate\write\eled@sectioningR@out{%
8925 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8926 }%
8927 \else%
8928 \immediate\write\eled@sectioning@out{%
8929 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL

```

```

}{{*}}{}
8930   }%
8931   \fi%
8932 }
8933
8934 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
8935   \disable@familiarnotes%
8936   #2%
8937   \restore@familiarnotes%
8938   \ifledRcol%
8939     \immediate\write\eled@sectioningR@out{%
8940       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR
8941     }}{{*}}{R}
8942     }%
8943     \else%
8944       \immediate\write\eled@sectioning@out{%
8945         \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
8946       }}{{*}}{}
8947       }%
8948       \fi%
8949     }
8950   %

```

XXXII.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.

```

8949 \def\eled@chapter#1#2#3#4#5{%
8950   \ifstrempy{#4}%
8951   {%
8952     \ifstrempy{#1}%
8953     {%
8954       \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}%
8955       \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}}%
8956     }%Need for \pairs, because of using parbox.
8957     {%
8958       \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%

```

```

8959 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
8960 {#2}}%Need for \pairs, because of using parbox.
8961 }%
8962 {%
8963 \ifstrempy{#1}%
8964 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
8965 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#1}{#2}}}%Bug in LaTeX!
8966 }%
8967 \listcsgadd{eled@sections#5@@}{#3}%
8968 }
8969 \def\eled@section#1#2#3#4#5{%
8970 \ifstrempy{#4}%
8971 {\ifstrempy{#1}%
8972 {%
8973 \csgdef{eled@sectioning@#3#5}{\section{#2}}%
8974 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
8975 {#2}}%Need for \pairs, because of using parbox.
8976 {%
8977 \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
8978 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
8979 {#1}}%Need for \pairs, because of using parbox.
8980 }%
8981 {\ifstrempy{#1}%
8982 {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
8983 {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
8984 }
8985 \listcsgadd{eled@sections#5@@}{#3}%
8986 }
8987 \def\eled@subsection#1#2#3#4#5{%
8988 \ifstrempy{#4}%
8989 {\ifstrempy{#1}%
8990 {%
8991 \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
8992 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{
subsectionmark}{#2}}}%Need for \pairs, because of using parbox. \csuse in
case of \subsectionmark is not defined (book)
8993 }%
8994 {%
8995 \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
8996 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{
subsectionmark}{#1}}}%Need for \pairs, because of using parbox. \csuse in
case of \subsectionmark is not defined (book)
8997 }%
8998 }%
8999 {\ifstrempy{#1}%

```

```

9000     {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
9001     {\csgdef{eled@sectioning@#3#5}{\subsection*[#1]{#2}}}%Bug in LaTeX!
9002   }
9003   \listcsgadd{eled@sections#5@@}{#3}%
9004   }
9005 \def\eled@subsubsection#1#2#3#4#5{%
9006   \ifstrempy{#4}%
9007   {\ifstrempy{#1}%
9008     {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
9009     {\csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
9010   }%
9011   {\ifstrempy{#1}%
9012     {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
9013     {\csgdef{eled@sectioning@#3#5}{\subsubsection*[#1]{#2}}}%Bug in
LaTeX!
9014   }
9015   \listcsgadd{eled@sections#5@@}{#3}%
9016   }
9017
9018 %

```

End of the conditional test about noeledsec option.

```

9019 }{}
9020 %

```

XXXIII Page breaking or no page breaking depending of specific lines

By default, page breaks are automatic. However, the user can define lines which will force page breaks, or prevent page breaks around one specific line. On the first run, the line-list file records the line number of where the page break is being changed (either forced, or prevented). On the next run, page breaks occur either before or after this line, depending on how the user sets the command. The default setting is after the line.

\normal@page@break \normal@page@break is an etoolbox list which contains the absolute line number of the last line, for each page.

```

9021 \def\normal@page@break{}
9022 %

```

\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.

```

9023 \def\l@prev@pb{}
9024 \def\l@prev@nopb{}
9025 %

```

`\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.

```
9026 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
9027 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
9028 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
9029 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
9030 %
```

`\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.

```
9031 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
9032 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
9033 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
9034 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
9035 %
```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

```
9036 \def\led@pb@setting{before}
9037 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
9038 %
```

`\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`.

```
9039 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
9040 \newcommand{\led@check@nopb}{%
9041   \IfStrEq{\led@pb@setting}{before}{%
9042     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
9043       {\numdef{\abs@prevline}{\the\absline@num-1}%
9044       \xifinlist{\abs@prevline}{\normal@page@break}%
9045       {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9046       {}}%
9047     }%
9048   }%
9049 }%
9050 \IfStrEq{\led@pb@setting}{after}{%
9051   \xifinlist{\the\absline@num}{\l@prev@nopb}{%
9052     \xifinlist{\the\absline@num}{\normal@page@break}%
9053     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9054     {}}%
9055 }%
```

```

9056     {}}%
9057     {}%
9058     {}%
9059 }
9060 %

```

XXXIV Long verse: prevents being separated by a page break

`\iflednopbinverse` The `\lednopbinverse` boolean is set to false by default. If set to true, `reledmac` will automatically prevent page breaks inside verse. The declaration is made at the beginning of the file, because it is used as a package option.

`\check@pb@in@verse` The `\check@pb@in@verse` checks if a verse is broken in two page. If true, it adds:

- The absolute line number of the first line of the verse -1 in the `\led@pb` list, if the page break must occur before the verse.
- The absolute line number of the first line of the verse -1 in the `\led@nopb` list, if the page break must occur after the verse.

```

9061 \newcommand{\check@pb@in@verse}{%
9062     \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
9063     \ifnum\page@num=\last@page@num\else%If we have change page
9064         \IfStrEq{\led@pb@setting}{before}{%
9065             \numgdef{\abs@line@verse}{\the\absline@num-1}%
9066             \ledpbnum{\abs@line@verse}%
9067         }{}%
9068         \IfStrEq{\led@pb@setting}{after}{%
9069             \numgdef{\abs@line@verse}{\the\absline@num-1}%
9070             \lednopbnum{\abs@line@verse}%
9071         }{}%
9072     \fi%
9073 \fi\fi\fi%
9074 }
9075 %

```

XXXV Tools for hyperref package

`\Hy@raisedlink@left` The `hyperref` package provides a `\Hy@raisedlink` command, to be used to add an anchor to the top of a line and not to the bottom of it.³⁵

³⁵<http://tex.stackexchange.com/a/17138/7712>.

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.³⁶

```

9076 \def\Hy@raisedlink@left#1{%
9077   \ifvmode
9078     #1%
9079   \else
9080     \Hy@SaveSpaceFactor
9081     \llap{\smash{%
9082       \begingroup
9083         \let\HyperRaiseLinkLength\@tempdima
9084         \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
9085         \HyperRaiseLinkHook
9086       \expandafter\endgroup
9087       \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
9088         \Hy@RestoreSpaceFactor
9089         #1%
9090         \Hy@SaveSpaceFactor
9091       }%
9092     }}%
9093     \Hy@RestoreSpaceFactor
9094     \penalty\@M\hskip\z@\relax
9095   \fi
9096 }
9097 %

```

XXXVI Compatibility with `eledmac`

Here, we define some commands for the `eledmac-compat` option.

```

9098 \ifeledmaccompat@%
9099
9100 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
9101 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
9102 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
9103 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
9104
9105 \unless\ifnocritical@
9106   \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
9107   \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
9108   \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
9109   \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
9110   \let\hsizetwocol\Xhsizetwocol
9111   \let\hsizethreecol\Xhsizethreecol
9112   \let\hookXnote\Xhooknote

```

³⁶The 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-coupure-de-ligne/801>.

```

9113 \let\boxsymlinenum\Xboxsymlinenum
9114 \let\symlinenum\Xsymlinenum
9115 \let\beforenumberinfootnote\Xbeforenumber
9116 \let\afternumberinfootnote\Xafternumber
9117 \let\beforeXsymlinenum\Xbeforesymlinenum
9118 \let\afterXsymlinenum\Xaftersymlinenum
9119 \let\inplaceofnumber\Xinplaceofnumber
9120 \let\Xlemmaseparator\lemmaseparator
9121 \let\afterlemmaseparator\Xafterlemmaseparator
9122 \let\beforelemmaseparator\Xbeforelemmaseparator
9123 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
9124 \let\txtbeforeXnotes\Xtxtbeforenotes
9125 \let\afterXrule\Xafterrule
9126 \let\numberonlyfirstinline\Xnumberonlyfirstinline
9127 \let\numberonlyfirstintwoline\Xnumberonlyfirstintwoline
9128 \let\nonnumberinfootnote\Xnonnumberinfootnote
9129 \let\pstartinfootnote\Xpstart
9130 \let\pstartinfootnoteeverytime\Xpstarteverytime
9131 \let\onlyXpstart\Xonlypstart
9132 \let\Xnonnumberinfootnote\Xnonnumber
9133 \let\Xnonbreakableafternumber\Xnonbreakableafternumber
9134 \let\maxhXnotes\Xmaxhnotes
9135 \let\beforeXnotes\Xbeforenotes
9136 \let\boxlinenum\Xboxlinenum
9137 \let\boxlinenumalign\Xboxlinenumalign
9138 \let\boxstartlinenum\Xboxstartlinenum
9139 \let\boxendlinenum\Xboxendlinenum
9140 \let\twoline\Xtwoline
9141 \let\morethantwoline\Xmorethantwoline
9142 \let\twolinebutnotmore\Xtwolinebutnotmore
9143 \let\twolineonlyinsamepage\Xtwolineonlyinsamepage
9144 \fi
9145
9146 \unless\ifnofamiliar@
9147 \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
9148 \fi
9149 \newcommandx{\parafootsep}[2][1,usedefault]{%
9150 \Xparafootsep[#1]{#2}%
9151 \parafootsepX[#1]{#2}
9152 }%
9153
9154 \newcommandx{\afternote}[2][1,usedefault]{%
9155 \Xafternote[#1]{#2}%
9156 \afternoteX[#1]{#2}%
9157 }%
9158
9159 \unless\ifnoend@
9160 \let\XendXtwoline\Xendtwoline
9161 \let\XendXmorethantwoline\Xendmorethantwoline
9162 \let\hookXendnote\Xendhooknote

```

```

9163 \let\boxXendlinenum\Xendboxlinenum%
9164 \let\boxXendlinenumalign\Xendboxlinenumalign%
9165 \let\boxXendstartlinenum\Xendboxstartlinenum%
9166 \let\boxXendendlinenum\Xendboxendlinenum%
9167 \let\XendXlemmaseparator\Xendlemmaseparator
9168 \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
9169 \let\XendXafterlemmaseparator\Xendafterlemmaseparator
9170 \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
9171 \fi
9172
9173 \AtBeginDocument{%
9174 \ifdef\lineref{}\let\lineref\edlineref}%
9175 }%
9176
9177
9178 \fi%
9179 %

```

</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 changed³⁷ 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
```

³⁷A 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. 122 to find out why.) Aliases of the second kind may be nested without any problem.

If you really have to use `\critext` in any of the tabular or array environments, then `\edtext` must not be used in the same environment. If you use `\critext` in one of these environments then you have to issue the declaration `\usingcritext` beforehand. The declaration `\usingedtext` must be issued to revert to the default assumption that `\edtext` will be used.

Appendix A.2 Migration from ledmac to eledmac

In `eledmac`, some changes were made in the code to allow easy customization. This may cause problems for people who have already made their own. The next sections explain how to handle this.

If you have created your own series using `\addfootins` and `\addfootinsX`, you must use instead the `\newseries` command (see 6.7.1 p. 34), 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. 35) or `\Xfootnote` options (6.2.2 p. 25). Otherwise please add a new ticket on Github to request a new function for doing this.³⁸

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]
```

³⁸<https://github.com/maieul/ledmac/issues>

If you do not make that, you will get a spurious [X], where X is series letter.

If you used a `\protect` command inside a `\footnote` command inside a numbered section, you must change the `\protect` to `\noexpand`. Otherwise the command after the `\protect` will be discarded.

Appendix A.3 Migration to eledmac 1.5.1

The version 1.5.1 corrects a bug in `stanzaindentsrepetition` (cf. 9.3 p. 50). This bug had two consequences:

1. `stanzaindentsrepetition` did not work when its value was greater than 2.
2. `stanzaindentsrepetition` worked wrong when its value was equal to 2.

So, if you used `stanzaindentsrepetition` with a value equal to 2, you had to change your `\setstanzaindents`. Explanation:

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

This code, in versions prior to 1.5.1, made the first line have an indentation of 0, the second line of 1, the third verse of 0, the fourth verse of 1 and so forth.

But this code should have instead achieved quite the contrary: the first line would have an indentation of 1, the second line of 0, the third line of 1, the fourth line of 0 and so forth.

So version 1.5.1 corrected this bug. If you want to keep the former presentation, you must change:

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

to:

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

Appendix A.4 Migration to eledmac 1.12.0

The migration to eledmac 1.12.0 is easy:

- You must first delete all the auxiliary files, then compile your document three times as usual.
- If you have modified `\l@reg`, which is not advisable, you must rename it to `\@nl@reg`.

There is an additional problem. If you have put text into brackets just after `\pstart` or `\pend`, this text will be considered to be an optional argument of `\pstart` or `\pend` (see 5.2.3 p. 18). If so, add a `\relax` between `\pstart`/`\pend` and the first bracket.

The version 1.12.0 also introduce a better way to handle sectional divisions inside numbered text. Please read 16.2 p. 66.

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. 68 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 `\Xlinerangeseparator` or `/` and `\Xendlinerangeseparator` (7.2.4 p. 38).

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. 41.

`\xlineref` does not print anymore the side flag (R for right side), because it is incompatible with numerical test. Use `\xflagref` to obtain it.

The `\printlines` and `\printendlines` commands take now an eighth argument, which is the side flag. It is strongly recommended to NEVER redefine these two commands and to use the setting commands instead (or to ask for new setting commands if the actual does not answer to your needs). However, if you have done it, just change your redefinition to have a new argument.

It is strongly recommended to stop redefining `\fullstop` and to use `\Xsublinesep` instead.

Appendix A.15 Migration to reledmac 2.7.0

`\SErefonlypage` (introduced in reledmac 2.5.0) added an parenthesis after the page number. This was just an error, linked to a bad imitation of `\SErefwithpage`. That has been deleted. And so, the `\XendafterpagenumberSErefonlypage` to set it was also deleted.

`\rigidbalance` is split to two new commands: `\Xrigidbalance` for critical footnotes and `\rigidbalanceX` for familiar footnotes. If you have redefined it — but why should you have ? —, you should split your single redefinition in two redefinitions.

Appendix A.16 Migration to reledmac 2.7.2

`\Xhsize` is already defined in the `floatrow` package. It becomes `\Xwidth`, and, consequently, `\hsizeX` becomes `\widthX`.

The ancient names are temporarily maintained as aliases.

Appendix A.17 Migration to reledmac 2.8.0

Reledmac 2.8.0 fix spurious indents for paragraphed critical and familiar footnotes in `ledgroup` and `minipage`. You can re-establish the indent with `\Xparinden` and `\parindentX`.

Appendix A.18 Migration to reledmac 2.13.1

Reledmac 2.5.0 added a bug, which makes the right flag to be printed on the right side of critical footnotes, even if not explicitly requested by using `\Xlineflag`.

Version 2.13.1 solves this issue. Please use `\Xlineflag` if you want to add the right flag.

Appendix A.19 Migration to reledmac 2.18.0

After updating reledmac, and before any new compilation, you need to clean your `.aux` files, if you use `\edlabel` or related.

References

- [Bre96] Herbert Breger. `tabmac`. October 1996. (Available from CTAN in `macros/plain/contrib/tabmac`)
- [Bur01] John Burt. ‘Typesetting critical editions of poetry’. *TUGboat*, **22**, 4, pp 353–361, December 2001. (Code available from CTAN in `macros/latex/contrib/poemscol`)
- [Eck03] Matthias Eckermann. *The Parallel-Package*. April 2003. (Available from CTAN in `macros/latex/contrib/parallel`)
- [Fai03] Robin Fairbairns. *footmisc—a portmanteau package for customising footnotes in L^AT_EX*. February 2003. (Available from CTAN in `macros/latex/contrib/footmisc`)
- [LW90] John Lavagnino and Dominik Wujastyk. ‘An overview of edmac: a PLAIN TeX format for critical editions’. *TUGboat*, **11**, 4, pp. 623–643, November 1990. (Code available from CTAN in `macros/plain/contrib/edmac`)
- [Lüc03] Uwe Lück. ‘ednotes—critical edition typesetting with LaTeX’. *TUGboat*, **24**, 2, pp. 224–236, 2003. (Code available from CTAN in `macros/latex/contrib/ednotes`)
- [Sul92] Wayne G. Sullivan. *The file edstanza.doc*. June 1992. (Available from CTAN in `macros/plain/contrib/edmac`)
- [Wil02] Peter Wilson. *The memoir class for configurable typesetting*. November 2002. (Available from CTAN in `macros/latex/contrib/memoir`)
- [Wil04] Peter Wilson and Maïeul Rouquette. *Parallel typesetting for critical editions: the eledpar package*. December 2004. (Available from CTAN in `macros/latex/contrib/ledmmac`)

Index

Symbols

<code>\&</code>	49
<code>\@EDROWFILL@</code>	1
<code>\@adv</code>	1
<code>\@advancestanzanumber</code>	1
<code>\@beforeinsertofthisedtext</code>	1
<code>\@doclearpage</code>	1
<code>\@doreinfeetX</code>	1
<code>\@edindex@fornote@</code>	1
<code>\@edindex@hyperref</code>	1
<code>\@edrowfill@</code>	1
<code>\@edtext@level</code>	1

\@emptytoks	1
\@fnpos	1
\@footnotemark	1
\@footnotetext	1
\@getfirstseries	1
\@gobblefour	1
\@gobbleseven	1
\@gobblethree	1
\@h	1
\@hangingsymbol	1
\@iiiminipage	1
\@insertstanzaumber	1
\@k	1
\@l@dttempcnta	1
\@l@dttempcntb	1
\@lab	1
\@led@testifnofoot	1
\@lemma	1
\@line@num	1
\@lock	1
\@lopL	1
\@lopR	1
\@makechapterhead	1
\@makeschapterhead	1
\@mem@extranofeet	1
\@mem@old@ssect	1
\@mpfnpos	1
\@msd	1
\@msd@c	1
\@msd@options@iffullpage	1
\@msdata@list	1
\@nl	1
\@nl@reg	1
\@opXfeet	1
\@pend	1
\@pendR	1
\@ref	1
\@ref@reg	1
\@ref@reg@parse	1
\@sect	1
\@series	1
\@set	1
\@sidenotesep	1
\@ssect	1
\@startstanza	1
\@stopmsd	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1

\@xloop	1
\@xympar	1
CLASSarticle	68
CLASSbook	68, 340
CLASSmemoir	196, 197, 257, 258, 291, 340, 360, 421, 425
CLASSscrbook	425
COMMAND*footnote	70
COMMAND\...@footnotemark...	200
COMMAND\...d@ta	148
COMMAND\<hook	
@<series	245
COMMAND\<hookname	
<pseudoserries	247, 248
COMMAND\<type	
footfmt	187
COMMAND\@@line	178
COMMAND\@MM	164, 422
COMMAND\@Rlineflag	293, 422
COMMAND\@Serefprefix	271
COMMAND\@Serefprefixmore	271
COMMAND\@add@	330
COMMAND\@adv	107
COMMAND\@apprefprefixmore	271
COMMAND\@apprefprefixsingle	271
COMMAND\@beforeinsertofthisedtext	129
COMMAND\@bsphack	260
COMMAND\@doclearpage	258, 415, 425
COMMAND\@doreinfeetX	425
COMMAND\@dprintingcolumns	422
COMMAND\@edindex@hyperref	293, 294
COMMAND\@edtext@level	126
COMMAND\@esphack	260
COMMAND\@fnpos	218, 254
COMMAND\@footnotemark	197, 415, 425
COMMAND\@footnotetext	197, 198, 415
COMMAND\@gobble	124, 125, 240
COMMAND\@gobblefive	424
COMMAND\@gobblefour	421
COMMAND\@gobbleseven	242
COMMAND\@gobblethree	414
COMMAND\@h	181
COMMAND\@hangingsymbol	298
COMMAND\@iiiminipage	283, 285, 414, 425
COMMAND\@iiiminpage	283
COMMAND\@l	420
COMMAND\@l@dtmncnta	150, 152, 159
COMMAND\@l@dtmncntb	152
COMMAND\@l@reg	420
COMMAND\@lab	103, 260, 263, 267, 414
COMMAND\@ldunboxmpfoot	285

COMMAND\@led@extranofeet	361
COMMAND\@ledinnote@command	289, 290
COMMAND\@lemma	129, 131
COMMAND\@lock	97, 299
COMMAND\@lopL	415
COMMAND\@lopR	415
COMMAND\@makecol	253, 254, 257, 425
COMMAND\@mpfnpos	218
COMMAND\@msd	308
COMMAND\@msd@c	308
COMMAND\@msd@options@iffullpage	314
COMMAND\@msdata@list	308, 309
COMMAND\@nl	103–106, 108, 117, 263, 414, 415
COMMAND\@nl@reg	104, 358, 415, 420
COMMAND\@opXfeet	415
COMMAND\@opfeetX	425
COMMAND\@opxtrafeeti	425
COMMAND\@page	105, 263
COMMAND\@pend	415
COMMAND\@pendR	415
COMMAND\@ref	103, 113–115, 118, 119, 124
COMMAND\@ref@later	114, 120
COMMAND\@ref@reg	113, 415
COMMAND\@ref@reg@parsearg	114
COMMAND\@reinserts	253, 255, 257, 425
COMMAND\@secondoftwo	70
COMMAND\@sect	343
COMMAND\@series	244
COMMAND\@set	107
COMMAND\@sidenotesep	282
COMMAND\@stopmsd	308
COMMAND\@sw	115, 132, 135, 136
COMMAND\@tag	125, 127, 130
COMMAND\@tempcnta	83
COMMAND\@tempcntb	83
COMMAND\@toksa	89
COMMAND\@toksb	89
COMMAND\@xloop	160
COMMAND\@xympar	275, 425
COMMAND\Aendnote	16, 25
COMMAND\Afootfmt	164
COMMAND\Afootgroup	164
COMMAND\Afootnote	8, 16, 24, 25, 28, 127, 172, 196, 219, 234, 424
COMMAND\Afootstart	164
COMMAND\AtBeginDocument	257
COMMAND\AtEndEveryPend	19, 431
COMMAND\AtEveryPend	18, 52, 142, 422, 423, 425, 431
COMMAND\AtEveryPend*	18
COMMAND\AtEveryPstart	18, 52, 422, 423, 425, 428, 431
COMMAND\AtEveryPstart*	18

COMMAND\AtEveryStanza	51, 429, 431
COMMAND\AtEveryStopStanza	52, 429, 431
COMMAND\AtStartEveryPstart	19, 431
COMMAND\AtStartEveryStanza	52, 431
COMMAND\BeforeEveryStopStanza	431
COMMAND\Bendnote	16, 24
COMMAND\Bfootnote	8, 16, 196, 219, 234
COMMAND\Centering	45
COMMAND\Cfootnote	196
COMMAND\Columns	84, 169
COMMAND\Dfootnote	196
COMMAND\Efootnote	196
COMMAND\Gls	62
COMMAND\Hy@raisedlink	352
COMMAND\Hy@raisedlink@left	353
COMMAND\LTR	45
COMMAND\NR	343
COMMAND\Pages	84, 255, 256
COMMAND\ProcessOptionsX	74
COMMAND\RL	44
COMMAND\RaggedLeft	45
COMMAND\RaggedRight	45
COMMAND\SEonlypage	268, 427
COMMAND\Seref	56–58, 268, 271, 428, 430
COMMAND\SErefonlypage	56–58, 364, 427
COMMAND\SErefwithpage	56–58, 268, 271, 364, 427, 429
COMMAND\Stanza	420
COMMAND\Waklam	331
COMMAND\X@doreinfeet	255, 425
COMMAND\XR@prefix	275
COMMAND\XR@test	274, 275
COMMAND\XR@test@mac	274
COMMAND\XR@test@mac@test	275
COMMAND\XXXXXXfmt	357
COMMAND\XXXXXXfmt	357
COMMAND\Xafterlemmaseparator	42, 361
COMMAND\Xafternote	45, 46, 360, 361
COMMAND\Xafternumber	40, 361
COMMAND\Xafterrule	47, 219, 361, 420, 423
COMMAND\Xaftersymlinenum	40, 361
COMMAND\Xarrangement	36, 46, 70, 165, 166, 246, 361
COMMAND\Xarrangement@footparagraph	171
COMMAND\Xarrangement@normal	166
COMMAND\Xarrangement@paragraph	170
COMMAND\Xbeforeinserting	44, 45
COMMAND\Xbeforelemmaseparator	42, 361
COMMAND\Xbeforenotes	46, 219, 361, 420, 423
COMMAND\Xbeforenumber	38, 40, 361
COMMAND\Xbeforesymlinenum	40, 361
COMMAND\Xhookendnote	362

COMMAND\Xbhookgroup	46, 427, 428
COMMAND\Xbhooknote	44, 362, 363, 425, 426
COMMAND\Xboxendlinenum	41, 362, 424
COMMAND\Xboxlinenum	40, 41, 362
COMMAND\Xboxlinenumalign	41, 362, 424
COMMAND\Xboxstartlinenum	41, 362, 424
COMMAND\Xboxsymlinenum	40, 41, 362
COMMAND\Xcolalign	45, 423
COMMAND\Xdo@feet	425, 430
COMMAND\Xend	242
COMMAND\XendXafterlemmaseparator	362
COMMAND\XendXbeforelemmaseparator	362
COMMAND\XendXinplaceoflemmaseparator	362
COMMAND\XendXlemmaseparator	362
COMMAND\XendXmorethantwolines	362
COMMAND\XendXtwolines	362
COMMAND\Xendafterenumber	40, 426
COMMAND\Xendafterlemmaseparator	42, 362
COMMAND\Xendafternote	48, 428
COMMAND\Xendafternumber	42
COMMAND\Xendafterpagenumbe	429
COMMAND\Xendafterpagenumber	41, 57
COMMAND\XendafterpagenumberSereonlypage	364
COMMAND\Xendaftersymlinenum	40, 42, 426
COMMAND\Xendahookinplaceofnumber	42, 426
COMMAND\Xendahooklinenum	42, 426
COMMAND\Xendbeforelemmaseparator	42, 362
COMMAND\Xendbeforelinenum	41
COMMAND\Xendbeforenumber	40, 426
COMMAND\Xendbeforepagenumber	41, 57, 58
COMMAND\XendbeforepagenumberSereonlypage	57
COMMAND\Xendbeforesymlinenum	40, 42, 426
COMMAND\Xendbhookinplaceofnumber	42, 426
COMMAND\Xendbhooklinenum	41, 426
COMMAND\Xendbhooknote	44
COMMAND\Xendboxendlinenum	41, 424
COMMAND\Xendboxlinenum	41, 362, 422
COMMAND\Xendboxlinenumalign	41, 362, 424
COMMAND\Xendboxstartlinenum	41, 424
COMMAND\Xendboxsymlinenum	41, 426
COMMAND\Xendhangindent	44, 426, 428
COMMAND\Xendinplaceoflemmaseparator	25, 42, 362
COMMAND\Xendinplaceofnumber	40, 425
COMMAND\Xendinplaceofpagenumber	38, 430
COMMAND\Xendinsertsep@	225
COMMAND\Xendlemmadisablefontselection	43
COMMAND\Xendlemmafont	43, 427
COMMAND\Xendlemmaseparator	26, 42, 362
COMMAND\Xendletboxendlinenum	362
COMMAND\Xendlineflag	58

COMMAND\Xendlineprefixmore	41, 58
COMMAND\Xendlineprefixsingle	41, 58
COMMAND\Xendlinerangeseparator	38, 58, 185, 364, 426
COMMAND\Xendmorethantwolines	25, 39, 58, 362, 423, 424
COMMAND\Xendnonumber	39, 425
COMMAND\Xendnote	222, 240–242, 423
COMMAND\Xendnotefontsize	43, 361
COMMAND\Xendnotenumfont	41–43, 361
COMMAND\Xendnotes	225
COMMAND\Xendnumberonlyfirstinline	37, 426
COMMAND\Xendnumberonlyfirstintwolines	37, 426
COMMAND\Xendpagenumberonlyfirst	37, 430
COMMAND\Xendpagenumberonlyfirstifsingle	37, 430
COMMAND\Xendpagenumberonlyfirstintwo	37, 430
COMMAND\Xendparagraph	48, 420
COMMAND\Xendsep	48
COMMAND\Xendsublinesep	39, 58, 185
COMMAND\Xendsymlinenum	37, 426
COMMAND\Xendsympagenum	38, 430
COMMAND\Xendtwolines	25, 38, 39, 58, 362, 423, 424
COMMAND\Xendtwolinesbutnotmore	39, 58, 423, 424
COMMAND\Xendtwolinesonlyinsamepage	39, 58, 423, 424
COMMAND\Xendwrapcontent	44, 429
COMMAND\Xendwraplemma	44, 429
COMMAND\Xfootfmt	363
COMMAND\Xfootgroup	170
COMMAND\Xfootins	169
COMMAND\Xfootnote	54, 60, 125, 357, 417, 421, 423, 427, 429
COMMAND\Xfootstarts	170
COMMAND\Xgroupbyline	46, 158, 195
COMMAND\Xgroupbylines	430
COMMAND\Xgroupbylineseparetwolines	46
COMMAND\Xhangindent	44, 426
COMMAND\Xhsize	364, 427, 428
COMMAND\Xhsizethreecol	45, 48, 362
COMMAND\Xhsizetwocol	45, 48, 247, 362
COMMAND\Xinplaceoflemmaseparator	25, 42, 362
COMMAND\Xinplaceofnumber	40, 362, 423, 424
COMMAND\Xinsertparafootsep	175, 177
COMMAND\Xledsetnormalparstuff	359, 360, 423
COMMAND\Xlemmadisablefontselection	43
COMMAND\Xlemmafont	43, 427
COMMAND\Xlemmaseparator	42, 185, 248, 251, 253, 362
COMMAND\Xlineflag	57, 364, 365, 429
COMMAND\Xlinerangeseparator	38, 57, 185, 364, 426
COMMAND\Xmaxhnotes	47, 68, 70, 361, 362, 420, 422
COMMAND\Xmorethantwolines	25, 38, 39, 57, 362, 422
COMMAND\Xnoindent	426
COMMAND\Xnolemmaseparator	42, 253, 362
COMMAND\Xnonbreakableafternumber	40, 362, 418

COMMAND\Xnonumber	39, 362
COMMAND\Xnonumberinfootnote	362
COMMAND\Xnotefontsize	43, 361
COMMAND\Xnotefontsize@⟨s⟩	175, 179, 181
COMMAND\Xnotenumfont	43, 361
COMMAND\Xnoteswidthliketwocolumns	48, 421
COMMAND\Xnumberonlyfirstinline	37, 38, 46, 100, 186, 247, 248, 251, 362, 417, 422, 431
COMMAND\Xnumberonlyfirstintwolines	37, 46, 362, 417, 431
COMMAND\Xonlypstart	39, 362, 417, 422
COMMAND\Xpagelinesep	40, 430
COMMAND\Xparafootsep	46, 100, 361, 362, 429, 431
COMMAND\Xparafootsep@series	175
COMMAND\Xparinden	364
COMMAND\Xparindent	44, 423, 426, 428
COMMAND\Xprenotes	47, 219, 429
COMMAND\Xprenotes@	169, 219, 417
COMMAND\Xpstart	39, 359, 362, 417, 422
COMMAND\Xpstarteverytime	39, 359, 362, 422
COMMAND\Xragged	46
COMMAND\Xrigidbalance	178, 364, 427
COMMAND\Xstanza	39, 52
COMMAND\Xstanzaseparator	39
COMMAND\Xstorelineinfo	186
COMMAND\Xsublinesep	22, 39, 40, 57, 185, 364
COMMAND\Xsublinesepside	22, 39
COMMAND\Xsymlinenum	37, 46, 361, 362, 424, 431
COMMAND\Xtoendnotes	26, 242
COMMAND\Xtwolines	25, 38, 39, 57, 191, 192, 247, 362, 422
COMMAND\Xtwolinesappref	247
COMMAND\Xtwolinesbutnotmore	38, 39, 57, 362, 423
COMMAND\Xtwolinesbutnotmoreappref	248
COMMAND\Xtwolinesonlyinsamepage	38, 39, 57, 362, 423
COMMAND\Xtxtbeforenotes	46, 362, 429, 430
COMMAND\Xunvxh	173, 363
COMMAND\Xwidth	48, 364, 428
COMMAND\Xwrapcontent	44, 429
COMMAND\Xwraplemma	43–45, 429
COMMAND\&	361
COMMAND\⟨XXX⟩vfootnote	195
COMMAND\absline@num	97, 149
COMMAND\accent	124
COMMAND\actionlines@list	98, 151
COMMAND\actions@list	98
COMMAND\add@Xgroupbyline	158
COMMAND\add@inserts	98, 157, 158
COMMAND\add@inserts@next	157, 158
COMMAND\add@msd@	308
COMMAND\add@msdata	308, 309
COMMAND\add@msdata@firstlineofpage	311
COMMAND\add@msddata	308

COMMAND\add@penalties	149, 159
COMMAND\addcontentsline	338
COMMAND\addfootins	357, 360
COMMAND\addfootinsX	357, 360
COMMAND\addtoendnotes	242
COMMAND\advancelabel@refs	262
COMMAND\advanceline	22, 99, 107, 121, 425
COMMAND\affixlin@num	281
COMMAND\affixline@num	152, 154, 156, 415
COMMAND\affixpstart@num	156
COMMAND\afterXrule	361
COMMAND\afterXsymlinenum	361
COMMAND\afterenumber	40
COMMAND\aftergroup	124, 128
COMMAND\afterlemmaseparator	361
COMMAND\afternote	361
COMMAND\afternoteX	46, 360, 361
COMMAND\afternumberinfootnote	361
COMMAND\afterruleX	47, 420, 423
COMMAND\applabel	56, 263, 265, 423, 429
COMMAND\appref	54, 56–58, 268, 271, 427, 428
COMMAND\apprefprefixmore	56, 361
COMMAND\apprefprefixsingle	56, 361
COMMAND\apprefwithpage	56–58, 268, 271, 424, 427
COMMAND\arrangementX	36, 70, 200, 246, 361
COMMAND\arrangementX@normal	206
COMMAND\article	15
COMMAND\at@every@pend	142
COMMAND\autopar	17, 18, 139, 143, 144, 216, 416, 418, 419, 423
COMMAND\ballast	69
COMMAND\ballast@count	149, 159
COMMAND\baselineskip	36, 171, 175
COMMAND\beforeXnotes	361
COMMAND\beforeXsymlinenum	361
COMMAND\beforeelectedchapter	10, 66, 339
COMMAND\beforeinsertingX	44
COMMAND\beforelemmaseparator	361
COMMAND\beforenotesX	47, 363, 419, 420, 423
COMMAND\beforenumberinfootnote	361
COMMAND\begin	316
COMMAND\beginnumbering	16, 17, 19, 84, 85, 87, 96, 101, 116, 143, 221, 307, 417, 420, 424, 425, 430, 431
COMMAND\bf	417
COMMAND\bfseries	43, 417
COMMAND\bhookXnote	362
COMMAND\bhookgroupX	46, 427
COMMAND\bhooknoteX	44, 363, 425, 426
COMMAND\body	299
COMMAND\bodyfootmarkA	33
COMMAND\book	15

COMMAND\boxXendlinenum	362
COMMAND\boxXendlinenumalign	362
COMMAND\boxXendstartlinenum	362
COMMAND\boxendlinenum	362
COMMAND\boxlinefootnote	189
COMMAND\boxlinenum	362
COMMAND\boxlinenumalign	362
COMMAND\boxstartlinenum	362
COMMAND\boxsymlinenum	362
COMMAND\break	36, 173
COMMAND\brokenpenalty	159
COMMAND\centering	45
COMMAND\ch@ck@l@ck	415
COMMAND\ch@cksub@l@ck	154, 415
COMMAND\chapter	66, 340, 344, 420, 423, 425, 429
COMMAND\chaptermark	338
COMMAND\check@pb@in@verse	352
COMMAND\colalignX	45, 423
COMMAND\collect@body	316
COMMAND\color	429
COMMAND\colorbox	70
COMMAND\columns	48
COMMAND\columnwidth	171, 421
COMMAND\command names	247
COMMAND\copyright	125
COMMAND\correct@Xfootins@box	422
COMMAND\correct@footinsX@box	422
COMMAND\count	179
COMMAND\critex	416
COMMAND\critext	131, 356, 357, 360
COMMAND\csize	75, 134
COMMAND\csquotes	238
COMMAND\ctab	332, 336
COMMAND\ctabtext	337
COMMAND\dccl	325
COMMAND\def	72
COMMAND\detokenize	134
COMMAND\dimen	179
COMMAND\dimexpr	48
COMMAND\discretionary	173
COMMAND\displaywidowpenalty	159
COMMAND\do@Xfeet	254, 415, 425, 430
COMMAND\do@actions	149–151, 415
COMMAND\do@actions@fixedcode	415
COMMAND\do@actions@next	150
COMMAND\do@ballast	149, 159
COMMAND\do@feet@custom@order	254
COMMAND\do@insidelinehook	418
COMMAND\do@line	98, 123, 141, 144, 147, 148, 157, 159, 299, 415, 416, 418, 420
COMMAND\do@linehook	415

COMMAND\do@lockoff	99
COMMAND\do@lockon	99
COMMAND\doereintrafeet	414
COMMAND\doendnotes	25, 225, 226, 363, 424, 431
COMMAND\doendnotesbysection	26, 226, 242, 424, 431
COMMAND\doennotes	431
COMMAND\doinsidelinehook	23, 421
COMMAND\dolinehook	23, 421
COMMAND\doreintrafeeti	425
COMMAND\doreintrafeetii	425
COMMAND\doxtrafeet	254, 414
COMMAND\doxtrafeeti	425
COMMAND\doxtrafeetii	425
COMMAND\dummy@ref	124
COMMAND\edaftertab	65, 331, 332
COMMAND\edatleft	64, 329
COMMAND\edatright	64, 65, 330
COMMAND\edbeforetab	65, 331, 332
COMMAND\edfilldimen	330
COMMAND\edfont@info	130
COMMAND\edgls	62, 288
COMMAND\edglsadd	431
COMMAND\edgls...	428
COMMAND\edindex	60–62, 288, 289, 292, 294, 320, 418, 421, 422, 425, 426, 430, 431
COMMAND\edindexlab	62
COMMAND\edlabel	54– 56, 58, 125, 259, 260, 262, 263, 265, 267, 274, 288, 320, 353, 365, 414, 417–419, 422, 427
COMMAND\edlabelE	56, 265
COMMAND\edlabelS	56, 265
COMMAND\edlabelSE	56
COMMAND\edlineref	54, 259, 362, 422, 424, 427, 431
COMMAND\edmakelabel	55, 56, 274
COMMAND\edpageref	54, 259, 265, 274
COMMAND\edrowfill	331
COMMAND\edsublineref	54
COMMAND\edtabcolsep	324
COMMAND\edtext	6, 24, 27, 29, 30, 49, 54–56, 60, 63, 70, 98, 113, 115, 118, 119, 122–131, 133– 135, 137, 263–265, 268, 320, 321, 340, 353, 356, 357, 360, 414, 416, 418, 420–424, 429–431
COMMAND\edtext@level	424
COMMAND\edtextlater	115
COMMAND\edvertdots	65, 330
COMMAND\edvertline	65, 330
COMMAND\elechapter	66
COMMAND\eled@sectioning@out	345
COMMAND\eledchapter	66, 360, 421, 425
COMMAND\eledchapter*	66
COMMAND\eledmac@error	414
COMMAND\eledsection	7, 16, 66, 124, 147, 339, 360, 423, 431
COMMAND\eledsection*	66
COMMAND\eledsubsection	66, 360

COMMAND\eledsubsection*	66
COMMAND\eledsubsubsection	66, 360
COMMAND\eledsubsubsection*	66
COMMAND\eledxxx	10, 67, 345, 420
COMMAND\eledxxxx	338
COMMAND\elesection	345
COMMAND\else	287, 339
COMMAND\empty	83, 153, 260
COMMAND\end	315, 316
COMMAND\end@lemmas	124
COMMAND\endashchar	185
COMMAND\endgraf	141, 175, 216
COMMAND\endlock	22, 99, 122, 303
COMMAND\endminipage	283, 285, 414, 425
COMMAND\endmsdata	32
COMMAND\endnotes	423, 427
COMMAND\endnumbering	16, 17, 19, 84, 85, 87, 415, 424, 430, 431
COMMAND\endprint	222, 224, 242, 359
COMMAND\endstanzaextra	361
COMMAND\endsub	21, 99, 120
COMMAND\endverse	363
COMMAND\everypar	143
COMMAND\extensionchars	68, 84
COMMAND\externaldocument	58, 274
COMMAND\fix@l@cks	415
COMMAND>falseverse	360, 418, 420
COMMAND\fi	339
COMMAND\firstlinenum	20, 152, 416
COMMAND\firstsublinenum	20, 416
COMMAND\fix@page	104, 105, 415
COMMAND\flag@end	118, 119, 130, 420
COMMAND\flag@end@RTL	119
COMMAND\flag@end@later	120
COMMAND\flag@start	118, 119, 130, 420, 421
COMMAND\flag@start@RTL	119
COMMAND\flag@start@later	120
COMMAND\flagstanza	53
COMMAND\floatingpenalty	164, 422
COMMAND\flush@notes	160
COMMAND\fnpos	35, 218, 419, 430
COMMAND\footfmt	164, 166
COMMAND\footfmt...	201
COMMAND\footfootmarkA	33
COMMAND\footfudgefactor	173
COMMAND\footfudgefiddle	70, 171, 414
COMMAND\footgroup	164
COMMAND\footins	169
COMMAND\footnormal	246, 361, 414
COMMAND\footnormalX	361
COMMAND\footnote	33, 69, 196–198, 358, 415

COMMAND\footnote@lang	185
COMMAND\footnoteA	16, 33
COMMAND\footnoteB	16
COMMAND\footnoteC	24
COMMAND\footnoteE	33
COMMAND\footnoteX	8, 34, 238, 240, 429
COMMAND\footnoteX@reading	240
COMMAND\footnoteXmark	240, 431
COMMAND\footnoteXmk	253
COMMAND\footnoteXtext	240, 431
COMMAND\footnote⟨X⟩	125
COMMAND\footnote⟨X⟩mark	34
COMMAND\footnote⟨X⟩mk	34
COMMAND\footnote⟨X⟩nomk	34
COMMAND\footnote⟨X⟩text	34
COMMAND\footnotelang@lua	163
COMMAND\footnotelang@poly	163
COMMAND\footnotemark	34, 240
COMMAND\footnoteoption@	162, 426
COMMAND\footnoterule	179
COMMAND\footnotesize	43
COMMAND\footnotetext	34, 240
COMMAND\footparagraph	171, 246, 361, 420
COMMAND\footparagraphX	211, 361, 420
COMMAND\footplitskips	415, 422
COMMAND\footstart	164, 169, 179
COMMAND\footstrut	175
COMMAND\footthreecol	361
COMMAND\footthreecolX	361, 424
COMMAND\foottwocol	361
COMMAND\foottwocolX	361, 424
COMMAND\foreignlanguage	44
COMMAND\fullstop	364
COMMAND\get@edindex@hyperref	293
COMMAND\get@edindex@ledinnote@command	289
COMMAND\get@fnmark	198
COMMAND\get@index@command	419
COMMAND\get@linelistfile	415
COMMAND\get@thisfootnote	205
COMMAND\getline@num	148, 150
COMMAND\gl@p	89
COMMAND\global	103
COMMAND\globaldefs	103
COMMAND\gls	62, 296
COMMAND\hangindentX	44, 423, 426
COMMAND\hangingsymbol	361, 416
COMMAND\hbox	173
COMMAND\hfill	419
COMMAND\hidenumbering	23, 112, 423
COMMAND\hidenumberingonleftpage	23, 112, 429

COMMAND\hidenumberingonrightpage	23, 112, 429
COMMAND\hline	63
COMMAND\hrulefill	331
COMMAND\hsize	37, 169, 171–173, 180, 182, 217, 415, 421
COMMAND\hsizeX	364, 427, 428
COMMAND\hsizethreecol	362
COMMAND\hsizethreecolX	45, 48
COMMAND\hsizetwocol	362
COMMAND\hsizetwocolX	45, 48
COMMAND\hyperlinkR	293
COMMAND\hyperlinkformat	293
COMMAND\hyperlinkformatR	293
COMMAND\if@RTL	76, 128
COMMAND\if@edtext@	421, 424
COMMAND\if@eled@sectioning	345
COMMAND\if@firstlineofpage	76
COMMAND\if@firstlineofpageR	76
COMMAND\if@msd@options@fullpage	314
COMMAND\if@msdata@insertedfrompreviouspage	311
COMMAND\if@nobreak	142
COMMAND\if@noneed@Footnote	118
COMMAND\ifXnote@	84
COMMAND\ifbypage@	90
COMMAND\ifbypage@R	90
COMMAND\ifbypstart@	90
COMMAND\ifbypstart@R	90
COMMAND\iffirst@linenum@out@	116, 117
COMMAND\ifindtl@innote	84
COMMAND\ifindtl@notenumber	84
COMMAND\ifinserthangingsymbol	299
COMMAND\ifinstanza	298
COMMAND\ifstwofollowinglines	192, 193
COMMAND\ifl@d@Xmorethantwolines	190, 422
COMMAND\ifl@d@Xtwolines	190
COMMAND\ifl@d@dash	190
COMMAND\ifl@d@elin	190
COMMAND\ifl@d@esl	190
COMMAND\ifl@d@pnum	189
COMMAND\ifl@d@ssub	190
COMMAND\ifl@dend@X	241
COMMAND\ifl@dmemoir	414
COMMAND\ifl@dpaging	421
COMMAND\ifl@dpairing	83, 416
COMMAND\ifl@dprintingpages	422
COMMAND\ifl@dskipnumber	152
COMMAND\ifl@dstartendok	330
COMMAND\ifl@imakeidx	76
COMMAND\ifledRcol	84, 416
COMMAND\ifledRcol@	84, 420
COMMAND\iflemmacommand@	421

COMMAND\ifnoend@	226
COMMAND\ifnoledgroup@	287
COMMAND\ifnoteschanged@	100
COMMAND\ifnumberedpar@	138
COMMAND\ifnumbering	85, 87
COMMAND\ifnumberingR	84, 416
COMMAND\ifnumberline	129, 152
COMMAND\ifpst@rted	416
COMMAND\ifpst@rtedL	85
COMMAND\ifseriesbefore	244
COMMAND\ifstopmsdata@inserted@	307
COMMAND\ifsublines@	97, 109
COMMAND\iftrue	424
COMMAND\ifvmode	262
COMMAND\ifxxx	339
COMMAND\ignorespaces	127
COMMAND\imki@wrindexentry	76
COMMAND\immediate	116, 117, 221
COMMAND\indent	18, 143, 363
COMMAND\index	295, 296, 430
COMMAND\indtl@wrindexentry	76
COMMAND\initnumbering@quote	337, 425
COMMAND\initnumbering@reg	415
COMMAND\initnumbering@sectcmd	425
COMMAND\inplaceoflemmaseparator	362
COMMAND\inplaceofnumber	362
COMMAND\insert	129, 157, 164, 166, 167, 180, 195, 201
COMMAND\insert@Xtxtbeforenotes	161, 180
COMMAND\insert@count	113, 118, 127
COMMAND\insert@countR	127
COMMAND\insert@msdata	308, 314
COMMAND\inserthangingsymbol	419
COMMAND\insertlines@list	98, 113
COMMAND\insertparafootsepX	215
COMMAND\inserts@list	123, 139, 157, 172
COMMAND\interAfootnotelinepenalty	416
COMMAND\interfootnotelinepenalty	416
COMMAND\interlinepenalty	164
COMMAND\interparanoteglue	360
COMMAND\justifying	45
COMMAND\l@advance@parledegroupp@beforenormalnotes	425
COMMAND\l@d@@wrindexhyp	421
COMMAND\l@d@add	132
COMMAND\l@d@end	221, 241
COMMAND\l@d@nums	127, 130–132, 189, 190
COMMAND\l@d@section	222
COMMAND\l@d@set	108, 121
COMMAND\l@dampcount	322
COMMAND\l@dbfnote	198, 415
COMMAND\l@dcheckstartend	330

COMMAND\l@dchset@num	108
COMMAND\l@dcolcount	322, 323
COMMAND\l@dcollect@@body	316
COMMAND\l@dcollect@body	315
COMMAND\l@dcsnote	420
COMMAND\l@dcsnotetext	148, 279
COMMAND\l@dcsnotetext@l	148, 279
COMMAND\l@dcsnotetext@r	148, 279
COMMAND\l@ddodoreintrafeet	255, 414
COMMAND\l@ddoxtrafeet	255, 414
COMMAND\l@demptyd@ta	416
COMMAND\l@dend@close	221
COMMAND\l@dend@open	221
COMMAND\l@dend@stuff	221
COMMAND\l@denvbody	316
COMMAND\l@dfeetbeginmini	414
COMMAND\l@dfeetendmini	414
COMMAND\l@dgetline@margin	416
COMMAND\l@dgetlock@disp	416
COMMAND\l@dgetref@num	267
COMMAND\l@dgetsidenote@margin	275, 416
COMMAND\l@dgobbeloptarg	421
COMMAND\l@dgobblearg	421
COMMAND\l@dgobbleoptarg	320
COMMAND\l@dlabel@parse	267
COMMAND\l@dld@ta	152, 154
COMMAND\l@dlp@rbox	281
COMMAND\l@dlsn@te	416
COMMAND\l@dlsnote	420
COMMAND\l@dmake@labels	262, 275
COMMAND\l@dmake@labelsR	275
COMMAND\l@dnumpstartsL	85, 416
COMMAND\l@dp@rsefootspec	190
COMMAND\l@dparsefootspec	190
COMMAND\l@dpush@begins	316
COMMAND\l@drd@ta	152, 154
COMMAND\l@dref@undefined	267
COMMAND\l@drrsn@te	416
COMMAND\l@drrsnote	420
COMMAND\l@dtabaddcols	330
COMMAND\l@dtabnoexpands	414
COMMAND\l@dumboxmpfoot	425
COMMAND\l@dunboxmpfoot	416
COMMAND\l@dzeropenalties	416, 421
COMMAND\l@pb	351
COMMAND\l@prev@nopb	350
COMMAND\l@prev@pb	350
COMMAND\l@reg	358
COMMAND\label	19, 56, 58, 62, 260, 268
COMMAND\label@refs	260

COMMAND\labelstarttrue	19, 417
COMMAND\labelref@list	259, 260, 263
COMMAND\language	173
COMMAND\last@page@num	415
COMMAND\lastbox	143
COMMAND\lastskip	120
COMMAND\latex@makecol	257
COMMAND\leavevmode	18, 143
COMMAND\led@check@nopb	351
COMMAND\led@check@pb	351
COMMAND\led@nopb	351, 352
COMMAND\led@nopbnum	351
COMMAND\led@pb	351, 352
COMMAND\led@pb@macro	351
COMMAND\led@pbnum	351
COMMAND\led@reinit@index@fornote	296
COMMAND\led@set@index@fornote	295
COMMAND\ledRflag	293
COMMAND\ledchapter	360, 418
COMMAND\ledfootinsdim	361
COMMAND\ledinnernote	58, 277, 420, 429
COMMAND\ledinnote	291, 359, 424
COMMAND\ledinnotemark	60, 359, 423
COMMAND\ledleftnote	59, 277
COMMAND\ledlinenum	95, 416
COMMAND\ledlfill	148
COMMAND\ledlsnotefontsetup	429
COMMAND\ledlsnotesep	59
COMMAND\ledlsnotewidth	59
COMMAND\lednopb	67, 351
COMMAND\lednopbinverse	352
COMMAND\lednopbinversetrue	51, 67
COMMAND\lednopbnum	351
COMMAND\ledouternote	58, 277, 420, 429
COMMAND\ledpb	67, 351, 360
COMMAND\ledpbnum	351
COMMAND\ledpbsetting	68, 351, 426
COMMAND\ledrightnote	59, 277
COMMAND\ledrsnotefontsetup	429
COMMAND\ledrsnotesep	59
COMMAND\ledrsnotewidth	59
COMMAND\ledsection	360
COMMAND\ledsectnomark	338
COMMAND\ledsectnotoc	338
COMMAND\ledsetnormalparstuff	359, 360, 423
COMMAND\ledsetnormalparstuff@common	216
COMMAND\ledsetnormalparstuffX	359, 360, 423
COMMAND\ledsidenote	59, 277, 279
COMMAND\ledsubsection	360
COMMAND\ledsubsubsection	360

COMMAND\ledxxx	420
COMMAND\left	64
COMMAND\leftctab	332
COMMAND\leftheadline	95
COMMAND\leftlinenum	21, 95, 414, 416
COMMAND\leftltab	331
COMMAND\leftnoteupfalse	59
COMMAND\leftstartnum	156
COMMAND\leftftab	331
COMMAND\leftsidenote	279
COMMAND\leftskip	169, 172, 173
COMMAND\lemma	3, 24, 27, 29, 30, 123, 126–128, 130, 131, 133, 356, 416, 417, 424, 425, 427
COMMAND\lemmaseparator	362
COMMAND\let	28, 49, 125, 129, 240, 257, 303, 414
COMMAND\letboxXendendlinenum	362
COMMAND\line	178, 181
COMMAND\line@list	98, 114, 130
COMMAND\line@list@stuff	85, 101, 116, 414, 416
COMMAND\line@list@version	103
COMMAND\line@margin	91, 154, 275
COMMAND\line@num	96, 97, 99, 152, 414
COMMAND\line@set	131, 132
COMMAND\lineation	21, 90
COMMAND\linebreak	36
COMMAND\linenum	24, 27, 54–56, 123, 131, 265, 268, 274, 356, 428
COMMAND\linenum@out	116, 260, 263
COMMAND\linenumberlist	20, 83, 153, 414
COMMAND\linenumberstyle	22, 95, 414, 431
COMMAND\linenumincrement	20, 416
COMMAND\linenummargin	21, 91, 275
COMMAND\linenumr@p	95, 414, 416
COMMAND\linenumrep	95, 416
COMMAND\linenumsep	21, 59, 95, 276
COMMAND\linerrangesep@	253
COMMAND\lineref	259, 266, 274, 362, 422
COMMAND\list@clear	89
COMMAND\list@clearing@reg	416
COMMAND\list@create	89
COMMAND\lock@disp	94
COMMAND\lock@off	110
COMMAND\lock@on	109
COMMAND\lockdisp	22, 94
COMMAND\loop	160, 161, 299
COMMAND\ltab	331, 332, 336
COMMAND\ltabtext	337
COMMAND\m@mmf@prepare	197
COMMAND\makeatletter	148
COMMAND\makehboxofhboxes	174, 175
COMMAND\makeindex	60, 292
COMMAND\makelabel	274

COMMAND\managestanza@modulo	300
COMMAND\marginpar	58, 69, 275, 415
COMMAND\marginparwidth	59, 276
COMMAND\markboth	147
COMMAND\mathchardef	300
COMMAND\maxhXnotes	362
COMMAND\maxhnotesX	47, 70, 361, 419, 420, 422–424
COMMAND\maxlinesinpar@list	101
COMMAND\measurebody	333
COMMAND\measuretbody	334
COMMAND\memorybreak	20
COMMAND\morenoexpands	70, 71, 123, 125
COMMAND\morethantwolines	362
COMMAND\mpfnpos	35, 218, 419, 430
COMMAND\mpnormalfootgroup	415
COMMAND\mpnormalvfootnote	415
COMMAND\msdata	32, 306–308, 431
COMMAND\msdataposition	33
COMMAND\multfootsep	33, 197
COMMAND\multiplefootnotemarker	197
COMMAND\musixtex	420
COMMAND\n@num	416, 423
COMMAND\n@num@ref	423
COMMAND\new@line	117, 415
COMMAND\new@series	125
COMMAND\newcommand	28, 72, 196, 262
COMMAND\newcommandx	28
COMMAND\newhookarg@specific	252
COMMAND\newhookcommand@series	247, 248, 423
COMMAND\newhookcommand@series@reload	248
COMMAND\newhookcommand@toggle@reload	248, 421
COMMAND\newhooktoggle@series	247, 248, 423
COMMAND\newhooktoggle@specific	252
COMMAND\newif	423
COMMAND\newline	36
COMMAND\newlinechar	241
COMMAND\newseries	34, 357, 360, 361
COMMAND\newseries@	232, 245
COMMAND\newverse	52, 53, 360, 420, 431
COMMAND\next	299
COMMAND\next@action	102
COMMAND\next@actionline	102
COMMAND\next@insert	158
COMMAND\nl@regR	104
COMMAND\no@expands	70, 125, 130, 240, 414
COMMAND\noXlemmaseparator	362
COMMAND\nobreak	189
COMMAND\nocritical	233
COMMAND\noeledsec	67, 360
COMMAND\noendnotes	360

COMMAND\noexpand	358
COMMAND\nofamiliar	250
COMMAND\noindent	18, 51, 52, 143, 426
COMMAND\noindentX	426
COMMAND\nomk@	253
COMMAND\nonbreakableafternumber	362
COMMAND\nonnumberinfootnote	362
COMMAND\normal@footnotemarkX	201
COMMAND\normal@page@break	350
COMMAND\normal@pars	216
COMMAND\normalbfnoteX	416
COMMAND\normalbodyfootmarkX	201
COMMAND\normalfont	430
COMMAND\normalfootfmt	49, 168, 175, 185, 222, 428
COMMAND\normalfootfmtX	202
COMMAND\normalfootfootmarkX	202
COMMAND\normalfootgroup	170
COMMAND\normalfootgroupX	203
COMMAND\normalfootnoterule	165
COMMAND\normalfootstart	168, 172
COMMAND\normalfootstartX	202
COMMAND\normalvfootnote	166, 167
COMMAND\normalvfootnote@inserted	167
COMMAND\normalvfootnoteX	201
COMMAND\notbool	339
COMMAND\notefontsetup	361
COMMAND\notefontsizeX	43, 361
COMMAND\notenumfont	361
COMMAND\notenumfontX	43, 361
COMMAND\notesXwidthliketwocolumns	362
COMMAND\noteswidthliketwocolumnsX	48, 362, 421, 423
COMMAND\num@lines	138, 159
COMMAND\numberlinefalse	20
COMMAND\numberlinetrue	20
COMMAND\numberonlyfirstinline	245, 362
COMMAND\numberonlyfirstintwolines	362
COMMAND\numberpstartfalse	19
COMMAND\numberpstarttrue	19, 39, 359, 417, 425, 431
COMMAND\numberstanza	39
COMMAND\numberstanzafalse	52
COMMAND\numberstanzatrue	52
COMMAND\numlabfont	21, 48, 95
COMMAND\one@line	138
COMMAND\onehalfspacing	426
COMMAND\onlyXpstart	362
COMMAND\onlysideX	239
COMMAND\page@action	99, 108
COMMAND\page@start	99, 416
COMMAND\pagecontents	99
COMMAND\pageparbreak	360

COMMAND\pageref	56, 265
COMMAND\par	27, 36, 143, 216
COMMAND\par@line	138, 159
COMMAND\para@footgroup	172
COMMAND\para@footgroupX	214
COMMAND\para@footsetup	171, 414
COMMAND\para@footsetupX	211, 414, 421
COMMAND\para@vfootnoteX	212
COMMAND\parafootfmt	174, 175, 428
COMMAND\parafootfmtX	213
COMMAND\parafootftm	177
COMMAND\parafootftmX	215
COMMAND\parafootftmsep	361
COMMAND\parafootsep	362, 419, 424
COMMAND\parafootsepX	46, 100, 361, 362, 429
COMMAND\parafootstart	172
COMMAND\parafootstartX	212
COMMAND\paravfootnote	172, 175
COMMAND\parfillskip	175
COMMAND\parindent	426
COMMAND\parindentX	44, 364, 426, 428
COMMAND\parshape	69
COMMAND\parskip	18, 51, 52, 143
COMMAND\pausenumbering	19, 87, 88, 101, 103, 144, 419, 421, 428
COMMAND\penalty	175
COMMAND\pend	2, 6, 17–22, 66, 67, 121, 123, 126, 132, 138–144, 156, 157, 358, 419, 420, 429–431
COMMAND\preXnotes	423, 429
COMMAND\prenotesX	47, 220, 423
COMMAND\prepare@Xgroupbyline	195
COMMAND\prepare@Xprenotes	219
COMMAND\prev@nopb	351
COMMAND\prev@pb	351
COMMAND\prevlineX	100
COMMAND\prevpageX@num	100
COMMAND\print@Xfootnoterule	423
COMMAND\print@Xnotes	255, 256
COMMAND\print@Xnotes@forpages	422
COMMAND\print@eledsection	147
COMMAND\print@footnoteXrule	423
COMMAND\print@leftmargin@eledsection	339
COMMAND\print@lemma	186
COMMAND\print@line	145
COMMAND\print@notesX@forpages	422
COMMAND\print@rightmargin@eledsection	339
COMMAND\printendlines	226, 232, 271, 364, 414, 416
COMMAND\printlinefootnote	187, 188, 422
COMMAND\printlinefootnotearea	188, 189, 422
COMMAND\printlinefootnotenotenumbers	187
COMMAND\printlines	168, 185, 189, 191, 226, 271, 364, 414, 416, 422, 427
COMMAND\printnpnum	359, 364

COMMAND\printpstart	185
COMMAND\protect	125, 358
COMMAND\providecommand	196, 414
COMMAND\pstart	2, 6, 17–22, 66, 67, 108, 121, 126, 132, 138, 139, 141–143, 147, 157, 344, 358, 416–418, 420, 421, 423–425, 428–431
COMMAND\pstartinfootnote	362
COMMAND\pstartinfootnoteeverytime	362
COMMAND\pstartnum	156
COMMAND\pstartref	54, 259, 266, 419, 431
COMMAND\pstarts	417
COMMAND\raggedX	46
COMMAND\raggedleft	45
COMMAND\raggedright	45
COMMAND\raw@text	138, 139
COMMAND\rbracket	42
COMMAND\read@linelist	101–103
COMMAND\ref	56, 58, 62
COMMAND\reformatted@	271
COMMAND\reformattedwithpage	271
COMMAND\relax	18, 108, 129, 150, 157, 303, 320, 358
COMMAND\renewcommand	70, 361, 364
COMMAND\reset@msd@options@iffullpage	314
COMMAND\resetprevline@	100
COMMAND\resetprevpage@	100
COMMAND\resumenumbering	19, 84, 87, 88, 101, 103, 144, 416, 420, 421, 428
COMMAND\righth	64
COMMAND\righthctab	332
COMMAND\rightrightnum	21, 95, 414, 416
COMMAND\righthltab	332
COMMAND\rightrightupfalse	59
COMMAND\rightrightab	332
COMMAND\rightrightnote	279
COMMAND\rightrightskip	169, 172, 173, 175
COMMAND\rightrightstartnum	156
COMMAND\rightrightbalance	177–179, 181, 364, 427
COMMAND\rightrightbalanceX	178, 364, 427
COMMAND\robustify	37
COMMAND\roman	325, 427
COMMAND\rtab	331–333, 336
COMMAND\rtabtext	333, 337
COMMAND\sameword	28–30, 115, 132–134, 137, 422, 424, 426, 429
COMMAND\sameword@inedtext	133, 134
COMMAND\saweword	133
COMMAND\scriptsize	95
COMMAND\section	66, 416, 430
COMMAND\section@num	84
COMMAND\sectionmark	338
COMMAND\select@lemmafont	48, 49, 162
COMMAND\series	232
COMMAND\series@	232

COMMAND\seriesatbegin	34, 244, 423
COMMAND\seriesatend	34, 244, 423
COMMAND\set@Xtxtbeforenotes	161
COMMAND\set@line	130
COMMAND\set@line@action	99, 109
COMMAND\setSErefonlypageprefixmore	57, 271, 428, 431
COMMAND\setSErefonlypageprefixsingle	57, 271, 428, 431
COMMAND\setSErefprefixmore	57
COMMAND\setSErefprefixsingle	57
COMMAND\setapprefprefixmore	57, 361
COMMAND\setapprefprefixsingle	57, 361, 427
COMMAND\setcommand@series	246
COMMAND\sethangingsymbol	51, 298, 361, 426
COMMAND\sethanginsymbol	49
COMMAND\setistwofollowinglines	192
COMMAND\setl@dlprbox	281
COMMAND\setline	22, 99, 104, 107, 121, 125, 141, 425
COMMAND\setlinenum	22, 104, 108, 121, 414
COMMAND\setmsdatalabel	33
COMMAND\setmsdataposition	431
COMMAND\setmsdataseries	33
COMMAND\setprintendlines	227, 228, 416
COMMAND\setprintlines	191, 192, 227, 416
COMMAND\setsidenotesepp	59
COMMAND\setsidenotsep	361
COMMAND\setstanzaindent	300
COMMAND\setstanzaindents	50, 300, 358
COMMAND\setstanzapenalties	300
COMMAND\setstanzavalues	300
COMMAND\settoggle@series	245, 417, 421
COMMAND\showlemma	124, 414, 415
COMMAND\showwordrank	31, 134
COMMAND\sidenote@margin	415
COMMAND\sidenotemargin	59, 415, 420
COMMAND\sidenotesepp	361
COMMAND\sidesstartnumtrue	19
COMMAND\skip	169
COMMAND\skipnumbering	23, 111, 122, 416, 423, 424
COMMAND\skipnumbering@reg	423
COMMAND\small	43
COMMAND\special	12
COMMAND\splitmaxdepth	164, 179
COMMAND\splitoff	178
COMMAND\splittopskip	164, 179, 181
COMMAND\stanza	22, 23, 51–53, 303, 361, 363, 426, 431
COMMAND\stanza@hang	302
COMMAND\stanza@line	302
COMMAND\stanzaindent	50, 300, 422
COMMAND\stanzaindent*	50
COMMAND\stanzaindentbase	299, 300

COMMAND\stanzanumwrapper	52
COMMAND\startlock	22, 99, 122, 303
COMMAND\startstanzahook	361
COMMAND\startsub	21, 99, 120
COMMAND\stopmsd	308
COMMAND\stopmsdata	32, 306, 307
COMMAND\strip@pt	172
COMMAND\strutbox	179
COMMAND\sub@action	99, 109
COMMAND\sub@lock	97
COMMAND\sub@off	106, 263
COMMAND\sub@on	106, 263
COMMAND\subline@num	96, 97, 99
COMMAND\sublinenum@rep	414
COMMAND\sublinenumstyle	23, 95, 414
COMMAND\sublinenumincrement	20
COMMAND\sublinenumr@p	95, 414, 416
COMMAND\sublinenumrep	95, 416
COMMAND\sublineref	54, 259, 266
COMMAND\subsectionmark	338
COMMAND\sw@inthisedtext	127
COMMAND\sw@list@inedtext	130, 137
COMMAND\symlinenum	362
COMMAND\symplinum	361
COMMAND\sza@penalty	302
COMMAND>tag	422
COMMAND\text	356
COMMAND\text<language>	44
COMMAND\textcolor	70
COMMAND\textheight	70
COMMAND\the	414
COMMAND\thefootnoteA	33
COMMAND\thefootnoteX	418
COMMAND\thelabidx	294
COMMAND\thepage	104
COMMAND\thepstart	19
COMMAND\thepstartL	417
COMMAND\thepstartR	417
COMMAND\thestanza	52
COMMAND\this@line@list@version	116
COMMAND\thisfootnote	205
COMMAND\threecol@begin@insert	180
COMMAND\threecolfootfmt	180, 428
COMMAND\threecolfootfmtX	210
COMMAND\threecolfootgroup	179
COMMAND\threecolfootgroupX	210
COMMAND\threecolfootsetup	179
COMMAND\threecolfootsetupX	209
COMMAND\threecolvfootnote	179
COMMAND\threecolvfootnote@inserted	179

COMMAND\threecolvfootnoteX	209
COMMAND\toendnotes	26, 225, 429
COMMAND\twocolfootfmt	428
COMMAND\twocolfootfmtX	207
COMMAND\twocolfootgroupX	208
COMMAND\twocolfootsetupX	207
COMMAND\twocolvfootnoteX	207
COMMAND\twolines	245, 362
COMMAND\twolines@A	245
COMMAND\twolines@B	245
COMMAND\twolines@C	245
COMMAND\twolinesbutnotmore	362
COMMAND\twolinesonlyinsamepage	362
COMMAND\xtbeforeXnotes	362
COMMAND\uline	34
COMMAND\unhbox	173
COMMAND\unpenalty	174, 175
COMMAND\unskip	175
COMMAND\unvxh	175, 363
COMMAND\unvxhX	363
COMMAND\upbracefill	331
COMMAND\usingcritext	357, 360
COMMAND\usingedtext	357, 360
COMMAND\VAfootnote	164
COMMAND\vadjust	119
COMMAND\variant	28
COMMAND\vbox	141, 143, 173, 178, 218
COMMAND\vfootnote	164, 169, 172, 179
COMMAND\vl@dbfnote	198, 415
COMMAND\vl@disnote	279
COMMAND\vl@dlsnote	279
COMMAND\vl@dosnote	279
COMMAND\vl@drsnote	279
COMMAND\vnumfootnoteX	416
COMMAND\vsizer	47, 70
COMMAND\vsplit	159
COMMAND\waklam	331
COMMAND\waklamec	331
COMMAND\wapunktel	331
COMMAND\wastricht	331
COMMAND\widthX	48, 364, 428
COMMAND\wrap@edcrossref	265, 421
COMMAND\wrapcontentX	44, 429
COMMAND\wrapped@bodyfootmarkX	215
COMMAND\wrapped@footfootmarkX	215
COMMAND\x...	55
COMMAND\xabslineref	266
COMMAND\xdef	89, 303
COMMAND\xflagref	55, 267, 364, 427
COMMAND\xleft@appenditem	89, 124

COMMAND\xlineref	55, 364, 427
COMMAND\xpageref	55
COMMAND\xpstartref	55, 419
COMMAND\xr	58
COMMAND\xright@appenditem	89
COMMAND\xsublineref	55
COMMAND\xxref	55, 268, 274, 419, 422, 429
COMMAND\zz@@@	414
ENVIRONMENTastanza	427
ENVIRONMENTedarrayc	336
ENVIRONMENTedarrayl	336
ENVIRONMENTedarrayr	336
ENVIRONMENTedtabularc	337
ENVIRONMENTedtabularl	337
ENVIRONMENTedtabularr	337
ENVIRONMENTledgroup	75, 286, 364, 427
ENVIRONMENTledgroupsize	286
PACKAGE(r)(e)ledmac	35
PACKAGEEledmac	11, 72, 99, 291, 359, 360, 422, 423
PACKAGEEledpar	423
PACKAGEEtoolbox	75
PACKAGEParallel	366
PACKAGEReledmac	363, 364
PACKAGEamsgen	315
PACKAGEamsmath	315
PACKAGEbabel	44, 71, 325, 427
PACKAGEbiblatex	69
PACKAGEbidi	44, 45, 76, 426
PACKAGEccaption	83
PACKAGEcolor	70
PACKAGEcsquotes	431
PACKAGEedmac	1, 6, 10, 12, 13, 72, 189, 196, 260, 300, 356, 366, 414
PACKAGEedstanza	1, 13, 298
PACKAGEeledmac	1, 10, 14–16, 61, 196, 288, 291, 318, 340, 353, 357, 359–361, 418, 420, 422
PACKAGEeledpar	83, 164, 338, 366, 416, 420–422
PACKAGEetex	426
PACKAGEetoolbox	88, 89, 132, 232, 245, 253, 279, 339, 350
PACKAGEfancyhdr	257, 430
PACKAGEfloatrow	68, 364
PACKAGEfootmisc	33, 71, 76, 196, 366, 430
PACKAGEgeometry	15
PACKAGEglossaries	62, 296, 427
PACKAGEhandout	421
PACKAGEhyperlink	240
PACKAGEhyperref	55, 126, 215, 260, 293, 343, 352, 419–421, 428, 431
PACKAGEifluatex	75
PACKAGEifxetex	75
PACKAGEimakeidx	60, 69, 76, 288, 291, 360, 418–420, 422
PACKAGEindextols	295
PACKAGEindextool	360

PACKAGEindextools	60, 69, 76, 84, 288, 291, 295, 360, 422, 427, 430
PACKAGEinputenc	134
PACKAGEledarab	71
PACKAGEledmac	1, 10, 13, 14, 71, 88, 291, 356, 357, 360, 363
PACKAGEledpar	71
PACKAGEMemoir	75, 291, 360, 366, 421
PACKAGEMorewrites	69
PACKAGEMusixtex	420
PACKAGEperpage	427
PACKAGEpolyglossia	42, 71, 163, 185, 427
PACKAGEragged2e	45, 75
PACKAGEreledmac	1, 2, 10–12, 14–17, 19, 20, 23, 25, 27–29, 31, 33, 34, 37, 38, 41, 44, 45, 47, 49, 51, 53, 54, 56, 58, 60–62, 67–73, 90, 92, 98, 99, 103, 106, 116, 125, 128, 129, 132, 157, 165, 169, 173, 185, 196, 222, 233, 237, 245, 253, 254, 265, 266, 269, 274, 291, 311, 318, 338, 339, 352, 353, 360, 361, 363–365, 425, 428, 430
PACKAGEreledpar	1, 4, 6, 8, 15, 18, 34, 48, 55, 57, 58, 67–69, 71, 73, 83, 90, 101, 106, 127, 129, 166, 169, 217, 232, 238, 253, 255, 256, 287, 298, 426, 427, 429, 430
PACKAGESuffix	75
PACKAGETabmac	1, 13, 366
PACKAGEulem	34
PACKAGEuninormalize	29
PACKAGExargs	28, 75
PACKAGExkeyval	73, 253
PACKAGEXR	5, 58, 274, 275, 428
PACKAGEXref	274
PACKAGEXstring	75, 293

A

\absline@num	1
Abu Kamil Shuja' b. Aslam	13
\actionlines@list	1
\actions@list	1
\add@inserts	1
\add@inserts@next	1
\add@msd@	1
\add@msdata	1
\add@msdata@firstlineofpage	1
\add@penalties	1
\add@Xgroupbyline	1
\addtol@denvbody	1
Adelard II	13
\advancelabel@refs	1
\advanceline	1, 22
\Aendnote	25
\affixline@num	1
\affixpstart@num	1
\affixside@note	1
\Afootnote	24
\afternoteX	45
\afterruleX	47

<code>\ampersand</code>	1, 53
<code>\append@notesX</code>	1
<code>\append@Xnotes</code>	1
<code>\applabel</code>	1, 56
<code>\appref</code>	1, 56
<code>\apprefwithpage</code>	1, 56
<code>\arrangementX</code>	1, 36
<code>\arrangementX@normal</code>	1
<code>\arrangementX@threecol</code>	1
<code>\arrangementX@twocol</code>	1
<code>\at@every@pend</code>	1
<code>\AtEndEveryPend</code>	1, 19
<code>\AtEveryPend</code>	1, 18
<code>\AtEveryPstart</code>	1, 18
<code>\AtEveryStanza</code>	1, 51
<code>\AtEveryStopStanza</code>	1, 52
<code>\AtStartEveryPstart</code>	1, 19
<code>\AtStartEveryStanza</code>	1, 52
<code>\autopar</code>	1, 17

B

<code>\ballast</code>	69
<code>\ballast@count</code>	1
Beeton, Barbara Ann Neuhaus Friend	18
<code>\beforeeledchapter</code>	1
<code>\BeforeEveryStopStanza</code>	1
<code>\beforeinsertingX</code>	44
<code>\beforenotesX</code>	47
<code>\beginnumbering</code>	1, 16
<code>\Bendnote</code>	25
<code>\Bfootnote</code>	24
<code>\bhookgroupX</code>	46
<code>\bhooknoteX</code>	44
<code>\bodyfootmarkA</code>	33
<code>\boxfootnotenumbers</code>	1
Bredon, Simon	13
Breger, Herbert	13, 318
Brey, Gerhard	13
Busard, Hubert L. L.	13
<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>	25
<code>\Cfootnote</code>	24
<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	13
Claassens, Geert H. M.	13
<code>\colalignX</code>	45
Copernicus, Nicolaus	13
<code>\critext</code>	356
<code>\ctab</code>	1
<code>\ctabtext</code>	1

D

Dekker, Dirk-Jan	70
<code>\Dendnote</code>	25
<code>\Dfootnote</code>	24
<code>\disable@familiarnotes</code>	1
<code>\disable@notes</code>	1
<code>\disable@sidenotes</code>	1
<code>\disablel@dtabfeet</code>	1
<code>\do@actions</code>	1
<code>\do@actions@fixedcode</code>	1
<code>\do@actions@next</code>	1
<code>\do@ballast</code>	1
<code>\do@feet@custom@order</code>	1
<code>\do@feetX</code>	1
<code>\do@insidelinehook</code>	1
<code>\do@line</code>	1
<code>\do@linehook</code>	1
<code>\do@lockoff</code>	1
<code>\do@lockoffL</code>	1
<code>\do@lockon</code>	1
<code>\do@lockonL</code>	1
<code>\do@Xfeet</code>	1
<code>\doedindexlabel</code>	1
<code>\doendnotes</code>	1, 25
<code>\doendnotesbysection</code>	1, 25
<code>\doinsidelinehook</code>	1, 23
<code>\dolinehook</code>	1, 23
<code>\dosplits</code>	1
Downes, Michael	69, 173, 175
<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, 65, 331
<code>edarrayc</code> (environment)	62
<code>edarrayl</code> (environment)	62
<code>edarrayr</code> (environment)	62
<code>\edatleft</code>	1, 64
<code>\edatright</code>	1, 64
<code>\edbeforetab</code>	1, 65, 331
<code>\edfilldimen</code>	1
<code>\edfont@info</code>	1
<code>\edindex</code>	1, 60
<code>\edindexlab</code>	1, 62
<code>\EDLABEL</code>	1
<code>\edlabel</code>	1, 54
<code>\edlabelE</code>	1, 56
<code>\edlabelS</code>	1, 56
<code>\edlabelSE</code>	1, 56
<code>\edlineref</code>	1, 54
<code>\edmakelabel</code>	1, 55
<code>\edpageref</code>	1, 54
<code>\edrowfill</code>	1, 63
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 63
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc</code> (environment)	62
<code>edtabularl</code> (environment)	62
<code>edtabularr</code> (environment)	62
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 24
<code>\edvertdots</code>	1, 65
<code>\edvertline</code>	1, 65
<code>\Eendnote</code>	25
<code>\Efootnote</code>	24
<code>\eled@chapter</code>	1
<code>\eled@section</code>	1
<code>\eled@sectioning@out</code>	1
<code>\eled@subsection</code>	1
<code>\eled@subsubsection</code>	1
<code>\eledchapter</code>	1
<code>\eledchapter*</code>	1
<code>\eledsection</code>	1
<code>\eledsection*</code>	1
<code>\eledsubsection</code>	1
<code>\eledsubsection*</code>	1
<code>\eledsubsubsection</code>	1
<code>\eledsubsubsection*</code>	1
<code>\enablel@dtabfeet</code>	1
<code>\end@lemmas</code>	1

<code>\endashchar</code>	1
<code>\endline@num</code>	1
<code>\endlock</code>	1, 22
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 16
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 21
<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	62
<code>edarrayl</code>	62
<code>edarrayr</code>	62
<code>edtabularc</code>	62
<code>edtabularl</code>	62
<code>edtabularr</code>	62
<code>ledgroup</code>	53
<code>ledgroupsize</code>	53
<code>minipage</code>	53
<code>Euclid</code>	13
<code>\extensionchars</code>	1, 68

F

<code>\f@x@l@cks</code>	1
Fairbairns, Robin	33
<code>\first@linenum@out@false</code>	1
<code>\first@linenum@out@true</code>	1
<code>\firstlinenum</code>	1, 20
<code>\firstseriesX@</code>	1
<code>\firstsublinenum</code>	1, 20
<code>\firstXseries@</code>	1
<code>\fix@page</code>	1
<code>\flag@end</code>	1
<code>\flag@end@later</code>	1
<code>\flag@end@RTL</code>	1
<code>\flag@start</code>	1
<code>\flag@start@later</code>	1
<code>\flag@start@RTL</code>	1
<code>\flagstanza</code>	1, 53
<code>\flush@notes</code>	1
<code>\fnpos</code>	1, 35
Folkerts, Menso	13
<code>\footfootmarkA</code>	33
<code>\footfudgefiddle</code>	1, 70
<code>\footnote</code>	1
<code>\footnoteA</code>	33
<code>\footnoteB</code>	33
<code>\footnoteC</code>	33

\footnoteD	33
\footnoteE	33
\footnotelang@lua	1
\footnotelang@poly	1
\footnoteoptions@	1
\footnoteXmark	34
\footnoteXtext	34
\footsplitskips	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
\get@thisfootnote	1
\get@thisfootnoteX	1
\getline@num	1
\gl@p	1

H

\h@num	1
\hangindentX	44
\hidenumbering	1, 23
\hidenumberingonleftpage	1, 23
\hidenumberingonrightpage	1
\Hilfsbox	1
\hilfsbox	1
\hilfscount	1
\HILFSskip	1
\Hilfsskip	1
\hilfsskip	1
\hsizethreecolX	45
\hsizetwocolX	45
\Hy@raisedlink@left	1
\hyperlinkformat	1
\hyperlinkformatR	1
\hyperlinkR	1

I

\if@addsw	1
\if@edtext@secondarg@	1
\if@eled@sectioning	1
\if@firstlineofpage	1
\if@led@nofoot	1
\if@ledgroup	1

<code>\if@lemmacommand@</code>	1
<code>\if@msdata@insertedfrompreviouspage</code>	1
<code>\if@noeled@sec</code>	1
<code>\if@noneed@Footnote</code>	1
<code>\if@RTL</code>	1
<code>\ifat@every@pend@star@</code>	1
<code>\ifat@every@pstart@star@</code>	1
<code>\ifautopar@pause</code>	1
<code>\ifbypage@</code>	1
<code>\ifbypage@R</code>	1
<code>\ifbypstart@</code>	1
<code>\ifbypstart@R</code>	1
<code>\ifeledmaccompat@</code>	1
<code>\iffirst@linenum@out@</code>	1
<code>\ifindtl@innote</code>	1
<code>\ifindtl@notenumber</code>	1
<code>\ifinserthangingsymbol</code>	1
<code>\ifinstanza</code>	1
<code>\ifl@d@dash</code>	1
<code>\ifl@d@elin</code>	1
<code>\ifl@d@esl</code>	1
<code>\ifl@d@pnum</code>	1
<code>\ifl@d@ssub</code>	1
<code>\ifl@d@Xmorethantwolines</code>	1
<code>\ifl@d@Xtwolines</code>	1
<code>\ifl@dend@X</code>	1
<code>\ifl@dhidenumber</code>	1
<code>\ifl@dmemoir</code>	1
<code>\ifl@dpaging</code>	1
<code>\ifl@dpairing</code>	1
<code>\ifl@dprintingcolumns</code>	1
<code>\ifl@dprintingpages</code>	1
<code>\ifl@dskipnumber</code>	1
<code>\ifl@dskipversenumber</code>	1
<code>\ifl@dstartendok</code>	1
<code>\ifl@footmisc</code>	1
<code>\ifl@imakeidx</code>	1
<code>\ifl@indextools</code>	1
<code>\ifledfinal</code>	1, 68
<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>\ifstopmsdata@inserted@</code>	1
<code>\ifsublines@</code>	1
<code>\ifwidthliketwocolumns</code>	1
<code>\ifXendinsertsep@</code>	1
<code>\ifxindy@</code>	1
<code>\ifxindyhyperref@</code>	1
<code>\initnumbering@quote</code>	1
<code>\initnumbering@reg</code>	1
<code>\insert@count</code>	0, 1
<code>\insert@msdata</code>	1
<code>\insert@Xtxtbeforenotes</code>	1
<code>\insert@hangingymbol</code>	1
<code>\insert@lines@list</code>	1
<code>\insert@parafootsepX</code>	1
<code>\inserts@list</code>	1

J

Jayaditya	13
-----------------	----

K

Kabelschacht, Alois	161
---------------------------	-----

L

<code>\l@advance@parledgroup@beforenormalnotes</code>	1
<code>\l@d@add</code>	1
<code>\l@d@nums</code>	1
<code>\l@d@section</code>	1
<code>\l@d@set</code>	1
<code>\l@d@Xend</code>	1
<code>\l@dampcount</code>	1
<code>\l@dbfnote</code>	1
<code>\l@dcheckcols</code>	1
<code>\l@dcheckstartend</code>	1
<code>\l@dchset@num</code>	1
<code>\l@dcolcount</code>	1
<code>\l@dcollect@body</code>	1
<code>\l@dcollect@body</code>	1
<code>\l@dcolwidth</code>	1
<code>\l@dcsnote</code>	1
<code>\l@dcsnotetext</code>	1

<code>\l@dcstotetext@l</code>	1
<code>\l@dcstotetext@r</code>	1
<code>\l@ddodoreinextrafeet</code>	1
<code>\l@dedbeginmini</code>	1
<code>\l@dedendmini</code>	1
<code>\l@emptyd@ta</code>	1
<code>\l@dend@close</code>	1
<code>\l@dend@open</code>	1
<code>\l@dend@stuff</code>	1
<code>\l@dend@Xfalse</code>	1
<code>\l@dend@Xtrue</code>	1
<code>\l@denbody</code>	1
<code>\l@desnote</code>	1
<code>\l@dfambeginmini</code>	1
<code>\l@dfamendmini</code>	1
<code>\l@dfeetbeginmini</code>	1
<code>\l@dfeetendmini</code>	1
<code>\l@dgetline@margin</code>	1
<code>\l@dgetlock@disp</code>	1
<code>\l@dgetref@num</code>	1
<code>\l@dgetsidenote@margin</code>	1
<code>\l@dobblearg</code>	1
<code>\l@disnote</code>	1
<code>\l@dlabel@parse</code>	1
<code>\l@dld@ta</code>	1
<code>\l@dlp@rbox</code>	1
<code>\l@dlsn@te</code>	1
<code>\l@dlsnote</code>	1
<code>\l@dmake@labels</code>	1
<code>\l@dmodforedtext</code>	1
<code>\l@dnullfills</code>	1
<code>\l@dnumstartsl</code>	1
<code>\l@doldold@footnotetext</code>	1
<code>\l@dp@rsefootspec</code>	1
<code>\l@dparsedendline</code>	1
<code>\l@dparsedendpage</code>	1
<code>\l@dparsedendsub</code>	1
<code>\l@dparsedstartline</code>	1
<code>\l@dparsedstartpage</code>	1
<code>\l@dparsedstartsub</code>	1
<code>\l@dparsefootspec</code>	1
<code>\l@dpush@begins</code>	1
<code>\l@drd@ta</code>	1
<code>\l@dref@undefined</code>	1
<code>\l@drestorefills</code>	1
<code>\l@drestoreforedtext</code>	1
<code>\l@drp@rbox</code>	1
<code>\l@drsn@te</code>	1
<code>\l@drsnote</code>	1
<code>\l@dsetmaxcolwidth</code>	1

<code>\l@dskipnumberfalse</code>	1
<code>\l@dskipnumbertrue</code>	1
<code>\l@dtabaddcols</code>	1
<code>\l@dtabnoexpands</code>	1
<code>\l@dunboxmpfoot</code>	1
<code>\l@dunhbox@line</code>	1
<code>\l@dzeropenalties</code>	1
<code>\label</code>	56
<code>\labelpstartfalse</code>	1
<code>\labelpstarttrue</code>	1, 19
<code>\labelref@list</code>	1
<code>\labelrefsparseline</code>	1
<code>\labelrefsparsesubline</code>	1
<code>\last@page@num</code>	1
Lavagnino, John	12
<code>\led@check@nopb</code>	1
<code>\led@check@pb</code>	1
<code>\led@err@AutoparNotNumbered</code>	1
<code>\led@err@BadAction</code>	1
<code>\led@err@edtextoutsidepstart</code>	1
<code>\led@err@EdtextWithoutFootnote</code>	1
<code>\led@err@FootnoteNotInSecondArgEdtext</code>	1
<code>\led@err@HighEndColumn</code>	1
<code>\led@err@LineationInNumbered</code>	1
<code>\led@err@LowStartColumn</code>	1
<code>\led@err@ManyLeftnotes</code>	1
<code>\led@err@ManyRightnotes</code>	1
<code>\led@err@ManySidenotes</code>	1
<code>\led@err@NumberingNotStarted</code>	1
<code>\led@err@NumberingShouldHaveStarted</code>	1
<code>\led@err@NumberingStarted</code>	1
<code>\led@err@NumberingWithoutPstart</code>	1
<code>\led@err@PendNoPstart</code>	1
<code>\led@err@PendNotNumbered</code>	1
<code>\led@err@PstartInPstart</code>	1
<code>\led@err@PstartNotNumbered</code>	1
<code>\led@err@ReverseColumns</code>	1
<code>\led@err@toendnotes@outsidenumbering</code>	1
<code>\led@err@TooManyColumns</code>	1
<code>\led@err@UnequalColumns</code>	1
<code>\led@error@fail@patch@@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@PackageAfterEledmac</code>	1
<code>\led@mess@NotesChanged</code>	1
<code>\led@mess@SectionContinued</code>	1
<code>\led@nopb</code>	1
<code>\led@nopbnum</code>	1

<code>\led@pb</code>	1
<code>\led@pb@setting</code>	1
<code>\led@pbnum</code>	1
<code>\led@reinit@index@fornote</code>	1
<code>\led@set@index@fornote</code>	1
<code>\led@toksa</code>	1
<code>\led@toksb</code>	1
<code>\led@warn@AppLabelOutSecondArgEdtext</code>	1
<code>\led@warn@BadAction</code>	1
<code>\led@warn@BadAdvancelineLine</code>	1
<code>\led@warn@BadAdvancelineSubline</code>	1
<code>\led@warn@BadLineation</code>	1
<code>\led@warn@BadLinenummargin</code>	1
<code>\led@warn@BadLockdisp</code>	1
<code>\led@warn@BadSetline</code>	1
<code>\led@warn@BadSetlinenum</code>	1
<code>\led@warn@BadSidenotemargin</code>	1
<code>\led@warn@BadSublockdisp</code>	1
<code>\led@warn@DuplicateLabel</code>	1
<code>\led@warn@edinde@outsidenumbering</code>	1
<code>\led@warn@LineFileObsolete</code>	1
<code>\led@warn@NoFile</code>	1
<code>\led@warn@NoIndexFile</code>	1
<code>\led@warn@NoMarginpars</code>	1
<code>\led@warn@RefUndefined</code>	1
<code>\led@warn@SeriesStillExist</code>	1
<code>\led@warning@hsizeX@deprecated</code>	1
<code>\led@warning@msdatawithoutstop</code>	1
<code>\led@warning@preXnotes@deprecated</code>	1
<code>\led@warning@Xhsize@deprecated</code>	1
<code>ledgroup (environment)</code>	53
<code>ledgroupsize (environment)</code>	53
<code>\ledinnernote</code>	1, 58
<code>\ledinnote</code>	1
<code>\ledinnotehyperpage</code>	1
<code>\ledinnotemark</code>	1
<code>\ledleftnote</code>	1, 59
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 59
<code>\ledlsnotesep</code>	1, 59
<code>\ledlsnotewidth</code>	1, 59
<code>\lednopb</code>	1, 67
<code>\lednopbinversetrue</code>	67
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	1, 58
<code>\ledpb</code>	1, 67
<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 67
<code>\ledrightnote</code>	1, 59

<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 59
<code>\ledrsnotesep</code>	1, 59
<code>\ledrsnotewidth</code>	1, 59
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 59
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 21
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	59
<code>\leftpstartnum</code>	1
<code>\leftftab</code>	1
Leibniz	13
<code>\lemma</code>	1, 27
<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, 21
<code>\linenum</code>	1, 27
<code>\linenum@out</code>	1
<code>\linenumberlist</code>	1, 20
<code>\linenumberstyle</code>	1, 22
<code>\linenumincrement</code>	1, 20
<code>\linenummargin</code>	1, 21
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 21
<code>\linrangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 22
Lorch, Richard	13
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	74

M

<code>\m@mmf@check</code>	1
<code>\m@mmf@prepare</code>	1

<code>\M@sect</code>	1
<code>\makehboxofhboxes</code>	1
<code>\managestanza@modulo</code>	1
<code>\maxhnotesX</code>	47
Mayer, Gyula	13
<code>\measurebody</code>	1
<code>\measurecell</code>	1
<code>\measuremrow</code>	1
<code>\measurtbody</code>	1
<code>\measuresetcell</code>	1
<code>\measuresetrow</code>	1
Middleton, Thomas	13, 97
<code>minipage</code> (environment)	53
Mittelbach, Frank	12
<code>\morenoexpands</code>	1, 70
<code>\mp@append@notesX</code>	1
<code>\mp@append@Xnotes</code>	1
<code>\mpfnpos</code>	1, 35
<code>\mpnormalfootgroup</code>	1
<code>\mpnormalfootgroupX</code>	1
<code>\mpnormalvfootnote</code>	1
<code>\mpnormalvfootnote@inserted</code>	1
<code>\mpnormalvfootnoteX</code>	1
<code>\mppara@footgroupX</code>	1
<code>\mppara@vfootnoteX</code>	1
<code>\mpparafootgroup</code>	1
<code>\mpparavfootnote</code>	1
<code>\mpthreecolfootgroup</code>	1
<code>\mpthreecolfootgroupX</code>	1
<code>\mpthreecolfootsetup</code>	1
<code>\mpthreecolfootsetupX</code>	1
<code>\mptwocolfootgroup</code>	1
<code>\mptwocolfootgroupX</code>	1
<code>\mptwocolfootsetup</code>	1
<code>\mptwocolfootsetupX</code>	1
<code>\msdata</code>	1, 32
<code>\msdata@c</code>	1
<code>\msdata@cR</code>	1
<code>\multfootsep</code>	1, 33
<code>\multiplefootnotemarker</code>	1

N

<code>\n@num</code>	1
<code>\n@num@stanza</code>	1
<code>\new@line</code>	1
<code>\newhookarg@specific</code>	1
<code>\newhookcommand@series</code>	1
<code>\newhookcommand@series@reload</code>	1
<code>\newhooktoggle@series</code>	1
<code>\newhooktoggle@series@reload</code>	1

<code>\newhooktoggle@specific</code>	1
<code>\newseries@</code>	1
<code>\newverse</code>	1
<code>\NEXT</code>	1
<code>\no@expands</code>	1
<code>\noeledsec</code>	67
<code>\nomk@</code>	1
<code>\normal@footnotemarkX</code>	1
<code>\normal@page@break</code>	1
<code>\normal@pars</code>	1
<code>\normalbfnoteX</code>	1
<code>\normalbodyfootmarkX</code>	1
<code>\normalfootfmt</code>	1
<code>\normalfootfmtX</code>	1
<code>\normalfootfootmarkX</code>	1
<code>\normalfootgroup</code>	1
<code>\normalfootgroupX</code>	1
<code>\normalfootnoterule</code>	1
<code>\normalfootnoteruleX</code>	1
<code>\normalfootstart</code>	1
<code>\normalfootstartX</code>	1
<code>\normalvfootnote</code>	1
<code>\normalvfootnote@inserted</code>	1
<code>\normalvfootnoteX</code>	1
<code>\notefontsizeX</code>	43
<code>\notenumfontX</code>	43
<code>\noteschanged@false</code>	1
<code>\noteschanged@true</code>	1
<code>\noteswidthliketwocolumnsX</code>	48
<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>	20
<code>\numberlinetrue</code>	20
<code>\numberpstartfalse</code>	1, 19
<code>\numberpstarttrue</code>	1, 19
<code>\numberstanzafalse</code>	52
<code>\numberstanzatrue</code>	52
<code>\numlabfont</code>	1, 48

O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>optionauxdir</code>	15, 429
<code>optioncontinuousnumberingwithcolumns</code>	428
<code>optioninnnote</code>	427

optioninnote	427
optionlinrangesep	253
optionnocritical	427
optionnoeledsec	344, 429
optionnoend	427
optionnopenalties	69
optionnotenumber	427

P

\page@action	1
\page@num	1
\pagelinesep	1, 60
\pageref	56
\par@line	1
\para@footgroupX	1
\para@footsetup	1
\para@footsetupX	1
\para@vfootnoteX	1
\parafootfmt	1
\parafootfmtX	1
\parafootgroup	1
\parafootsepX	46
\parafootstart	1
\parafootstartX	1
\paravfootnote	1
\parindentX	44
\pausenumbering	1, 19
\pend	1, 17
Plato of Tivoli	13
\postbodyfootmark	1
\prebodyfootmark	1
\prenotesX	47
\prepare@edindex@fornote	1
\prepare@prenotesX	1
\prepare@Xgroupbyline	1
\prepare@Xprenotes	1
\prev@nopb	1
\prev@pb	1
\prevpage@num	1
\print@eledsection	1
\print@footnoteXrule	1
\print@leftmargin@eledsection	1
\print@lemma	1
\print@line	1
\print@notesX	1
\print@rightmargin@eledsection	1
\print@Xfootnoterule	1
\print@Xnotes	1
\printendlines	1
\printlineendnote	1

<code>\printlineendnotearea</code>	1
<code>\printlinefootnote</code>	1
<code>\printlinefootnotearea</code>	1
<code>\printlinefootnotenumbers</code>	1
<code>\printlines</code>	1
<code>\printnpnum</code>	1
<code>\printpstart</code>	1
<code>\printsymlineendnotearea</code>	1
<code>\printsymlinefootnotearea</code>	1
<code>\printXafternumber</code>	1
<code>\printXbeforenumber</code>	1
<code>\pstart</code>	1, 17
<code>\pstarteref</code>	1
<code>\pstartnum</code>	1
<code>\pstartref</code>	54

Q

<code>\quotation</code>	1
<code>\quote</code>	1

R

<code>\raggedX</code>	46
<code>\raw@text</code>	1
<code>\rbracket</code>	1
<code>\read@linelist</code>	1
<code>\ref</code>	56
<code>\ref@reg@later</code>	1
<code>\Relax</code>	1
<code>\reledmac@error</code>	1
<code>\reledmac@warning</code>	1
<code>\removehboxes</code>	1
<code>\reset@msd@options@iffullpage</code>	1
<code>\resetprevline@</code>	1, 100
<code>\resetprevpage@</code>	1
<code>\resetprevpage@num</code>	100
<code>\restore@familiarnotes</code>	1
<code>\restore@notes</code>	1
<code>\restore@sidenotes</code>	1
<code>\resumenumbering</code>	1, 19
<code>\rightctab</code>	1
<code>\rightlinenum</code>	1, 21
<code>\rightltab</code>	1
<code>\rightnoteupfalse</code>	59
<code>\rightrtab</code>	1
<code>\rightstartnum</code>	1
<code>\rigidbalance</code>	1
<code>\rigidbalanceX</code>	1
<code>\rtab</code>	1
<code>\rtabtext</code>	1

S

Sacrobosco	13
\sameword	<u>1</u> , 28
\sameword@inedtext	<u>1</u>
Schöpf, Rainer	12
\section@num	<u>1</u>
\select@lemmafонт	<u>1</u>
\select@lemmafонт	<u>1</u> , 48
\Seref	<u>1</u> , 56
\Serefonlypage	56
\Serefwithpage	<u>1</u> , 56
\series	<u>1</u>
\seriesatbegin	<u>1</u> , 34
\seriesatend	<u>1</u> , 34
\set@line	<u>1</u>
\set@line@action	<u>1</u>
\set@Xtxtbeforenotes	<u>1</u>
\setapprefprefixmore	56
\setapprefprefixsingle	56
\setcommand@series	<u>1</u>
\sethangingsymbol	<u>1</u> , 51
\setistwofollowinglines	<u>1</u>
\setl@dlp@rbox	<u>1</u>
\setl@drpr@box	<u>1</u>
\setline	<u>1</u> , 22
\setlinenum	<u>1</u> , 22
\setmcellcenter	<u>1</u>
\setmcellleft	<u>1</u>
\setmcellright	<u>1</u>
\setmrowcenter	<u>1</u>
\setmrowleft	<u>1</u>
\setmrowright	<u>1</u>
\setmsdatalabel	<u>1</u> , 33
\setmsdataposition	<u>1</u> , 33
\setmsdataseries	<u>1</u> , 33
\setnoteswidthliketwocolumnsX@	<u>1</u>
\setnotesXpositionliketwocolumns@	<u>1</u>
\setprintendlines	<u>1</u>
\setprintlines	<u>1</u>
\setSErefonlypageprefixmore	57
\setSErefonlypageprefixsingle	57
\setSErefprefixmore	57
\setSErefprefixsingle	57
\setsidenotesep	59
\setstanzaindents	<u>1</u> , 49
\setstanzapenalties	<u>1</u> , 50
\setstanzavalues	<u>1</u>
\settcclcenter	<u>1</u>
\settcclleft	<u>1</u>
\settcclright	<u>1</u>

<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, 68
<code>\showwordrank</code>	1, 31
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 59
<code>\sidepstartnumtrue</code>	19
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 23
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 63
<code>\spreadtext</code>	1, 63
<code>\stanza</code>	1, 49
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 50
<code>\stanzaindent*</code>	1, 50
<code>\stanzaindentbase</code>	1, 49
<code>\stanzanumwrapper</code>	1, 52
<code>\startlock</code>	1, 22
<code>\startsub</code>	1, 21
<code>\stepl@colcount</code>	1
<code>\stopmsdata</code>	1, 32
<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, 22
<code>\sublinenumincrement</code>	1, 20
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 54
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	12, 13, 69, 83, 87, 173, 174, 260, 298
<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>	33
<code>Theodosius</code>	13
<code>\thepageline</code>	1
<code>\thepstart</code>	1, 19
<code>\thestanza</code>	1, 52
<code>\thestartpageline</code>	1
<code>\this@line@list@version</code>	1
<code>\threecol@begin@insert</code>	1
<code>\threecolfootfmt</code>	1
<code>\threecolfootfmtX</code>	1
<code>\threecolfootgroup</code>	1
<code>\threecolfootgroupX</code>	1
<code>\threecolfootsetup</code>	1
<code>\threecolfootsetupX</code>	1
<code>\threecolvfootnote</code>	1
<code>\threecolvfootnote@inserted</code>	1
<code>\threecolvfootnoteX</code>	1
<code>\toendnotes</code>	1, 26
<code>\toendnotes*</code>	1
<code>\twocolfootfmt</code>	1
<code>\twocolfootfmtX</code>	1
<code>\twocolfootgroup</code>	1
<code>\twocolfootgroupX</code>	1
<code>\twocolfootsetup</code>	1
<code>\twocolfootsetupX</code>	1
<code>\twocolvfootnote</code>	1
<code>\twocolvfootnote@inserted</code>	1
<code>\twocolvfootnoteX</code>	1

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@dcsnote</code>	1
<code>\vl@disnote</code>	1
<code>\vl@dlsnote</code>	1
<code>\vl@dosnote</code>	1
<code>\vl@drsnote</code>	1
<code>\vnumfootnoteX</code>	1

W

<code>Whitney, Ron</code>	12
<code>\widthX</code>	48
<code>\wrap@edcrossref</code>	1

\wrapcontentX	44
\wrapped@bodyfootmarkX	1
\wrapped@footfootmarkX	1
Wujastyk, Dominik	12

X

\X@atbegininsertion	1
\X@beforeinsertion	1
\X@doreinfeet	1
\xabslineref	1
\Xafterlemmaseparator	42
\Xafternote	45
\Xafternumber	40
\Xafterrule	47
\Xaftersymlinenum	40
\Xarrangement	1, 36
\Xarrangement@normal	1
\Xarrangement@paragraph	1
\Xarrangement@threecol	1
\Xarrangement@twocol	1
\Xbeforeinserting	44
\Xbeforelemmaseparator	42
\Xbeforenotes	46
\Xbeforenumber	38, 40
\Xbeforesymlinenum	40
\Xhookgroup	46
\Xhooknote	44
\Xboxlinenum	40
\Xboxlinenumalign	41
\Xboxsymlinenum	40
\Xcolalign	45
\xedindex	1
\xedlabel	1
\xedtext	1
\Xendafterenumber	40
\Xendafterlemmaseparator	42
\Xendafternote	48
\Xendafterpagenumber	41
\Xendaftersymlinenum	40
\Xendahookinplaceofnumber	42
\Xendahooklinenumber	42
\Xendbeforelemmaseparator	42
\Xendbeforenumber	40
\Xendbeforepagenumber	41
\Xendbeforesymlinenum	40
\Xendbhookinplaceofnumber	42
\Xendbhooklinenumber	41
\Xendbhooknote	44
\Xendboxendlinenumalign	41
\Xendboxlinenum	41

<code>\Xendboxlinenumalign</code>	41
<code>\Xendboxstartlinenumalign</code>	41
<code>\Xendboxsymlinenum</code>	41
<code>\Xendhangindent</code>	44
<code>\Xendinplaceoflemmaseparator</code>	42
<code>\Xendinplaceofnumber</code>	40
<code>\Xendinplaceofpagenumber</code>	38
<code>\Xendlemmadisablefontselection</code>	43
<code>\Xendlemmafont</code>	43
<code>\Xendlemmaseparator</code>	42
<code>\Xendlineprefixmore</code>	41
<code>\Xendlineprefixsingle</code>	41
<code>\Xendlinangeseparator</code>	38
<code>\Xendmorethantwolines</code>	39
<code>\Xendnonumber</code>	39
<code>\Xendnotefontsize</code>	43
<code>\Xendnotenumfont</code>	43
<code>\Xendnumberonlyfirstinline</code>	37
<code>\Xendnumberonlyfirstintwolines</code>	37
<code>\Xendpagenumberonlyfirst</code>	37
<code>\Xendparagraph</code>	48
<code>\Xendsep</code>	48
<code>\Xendsublinesep</code>	39
<code>\Xendsymlinenum</code>	37
<code>\Xendsympagenum</code>	38
<code>\Xendtwolines</code>	39
<code>\Xendtwolinesbutnotmore</code>	39
<code>\Xendwrapcontent</code>	44
<code>\xflagref</code>	1
<code>\Xgroupbyline</code>	46
<code>\Xgroupbylineseparetwolines</code>	46
<code>\Xhangindent</code>	44
<code>\Xhsizethreecol</code>	45
<code>\Xhsizetwocol</code>	45
<code>\Xinplaceoflemmaseparator</code>	42
<code>\Xinplaceofnumber</code>	40
<code>\Xinsertparafootsep</code>	1
<code>\Xledsetnormalparstuff</code>	1
<code>\xleft@appenditem</code>	1
<code>\Xlemmadisablefontselection</code>	43
<code>\Xlemmafont</code>	43
<code>\Xlemmaseparator</code>	42
<code>\Xlinangeseparator</code>	38
<code>\xlineref</code>	1, 54
<code>\Xmaxhnotes</code>	47
<code>\Xmorethantwolines</code>	38
<code>\Xnolemmaseparator</code>	1, 42
<code>\Xnonbreakableafternumber</code>	40
<code>\Xnonumber</code>	39
<code>\Xnotefontsize</code>	43

\Xnotenumfont	43
\Xnoteswidthliketwocolumns	48
\Xnumberonlyfirstinline	37
\Xnumberonlyfirstintwolines	37
\Xonlypstart	39
\Xpagelinesep	40
\xpageref	<u>1</u> , 54
\Xparafootsep	46
\Xparindent	44
\Xprenotes	<u>1</u> , 47
\Xprenotes@	<u>1</u>
\Xpstart	39
\Xpstarteverytime	39
\xpstartref	<u>1</u> , 54
\XR@test	<u>1</u>
\XR@test@mac	<u>1</u>
\XR@test@mac@test	<u>1</u>
\Xragged	46
\xright@appenditem	<u>1</u>
\Xrigidbalance	<u>1</u>
\Xstanza	39
\Xstanzaseparator	39
\Xstorelineinfo	<u>1</u>
\xsublineref	<u>1</u> , 54
\Xsublinesep	39
\Xsymlinenum	37
\Xtoendnotes	26
\Xtwolines	38
\Xtwolinesonlyinsamepage	38
\Xtxtbeforenotes	46
\Xunvxh	<u>1</u>
\Xwidth	48
\Xwrapcontent	44
\Xwrapendlemma	44
\Xwraplemma	43
\xxref	<u>1</u> , 55
Z	
\zz@@@	<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	75
\morenoexpands: Added \l@dtabnoexpands to \no@expands	125
\reledmac@error: Added \eledmac@error and replaced error messages	77
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	263
\doxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	254
\edlabel: Tweaked \edlabel to get correct page numbers	260
\l@ddodoreinxtrafeet: Renamed \dodoreinxtrafeet to \l@ddodoreinxtrafeet	255
\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	125
\zz@@@: Minor change to \zz@@@	260
General: Added text about normal labeling	56
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	257
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	171
\line@list@stuff: Added initial write of page number in \line@list@stuff	117
\para@footsetup: Added \footfudgefiddle to \para@footsetup	171
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	211
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	83
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	263
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	104
\ledlinenum: Added \linenumr@p and \sublinenum@rep to \leftlinenum and \rightlinenum	95
\linenumberlist: Added \linenumberlist mechanism	83
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	228
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	194
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	95
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	285
\Xarrangement@normal: Added minpage footnote setup to \footnormal	166
\edtext: Added \showlemma to \edtext	126
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	283

\mpnormalfootgroup: Added \mpnormalfootgroup	170
\mpnormalvfootnote: Added \mpnormalvfootnote	167
\showlemma: Added \showlemma	83
General: Added final/draft options	73
Added ledgroup environment	286
Added ledgroupsize environment	286
Added minipage, etc., support	1
v0.4.1.	
\do@Xfeet: Changed \do@Xfeet code for easier extensions	254
\edindex: Let eledmac take advantage of memoir's indexing	291
\print@Xnotes: Added \op@Xfeet	255
General: Added code for changing \doclearpage	258
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	197
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	198
\@xympar: Eliminated \marginpar disturbance	275
General: Added left and right side notes	275
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	152
\l@edgetsidenote@margin: Added \sidenotemargin and \sidenote@margin	276
General: Added moveable side note	275
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	286
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	106
\@nl@reg: Added \fix@page to \@nl	104
Extended \@nl to include the page number	104
\fix@page: Added \last@page@num and \fix@page	105
\get@thisfootnote: Changed \l@dbfnote and \vl@dbfnote as originals could give incorrect markers in the footnotes	198
\new@line: Extended \new@line to output page numbers	117
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	104
\@ref@reg: Added \@ref@reg	113
\affixline@num: Added skipnumering to \affixline@num	152
\do@actions@fixedcode: Added \do@actions@fixedcode	151
\do@actions@next: Added number skipping to \do@actions	150
\do@insidelinehook: Added \do@linehook for use in \do@line	147
\endnumbering: Changed \endnumbering for eledpar	87
\fix@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fix@l@cks	154
\footplitskips: Added \footplitskips for use in many footnote styles	164
\get@linelistfile: Added \get@linelistfile	103
\initnumbering@reg: Added \initnumbering@reg	85

\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	285
\l@dcsnotetext@r: Added \l@emptyd@ta	148
\l@dgetline@margin: Added \l@dgetline@margin	91
\l@dgetlock@disp: Added \l@dgetlock@disp	94
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	276
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	83
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	148
\l@dzeropenalties: Added \l@dzeropenalties	143
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	95
\line@list@stuff: Deleted \page@start from \line@list@stuff	117
\list@clearing@reg: Added \list@clearing@reg	102
\n@num: Added \n@num	111
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	204
\resumenumbering: Changed \resumenumbering for eledpar	88
\setprintendlines: Added \setprintendlines for use by \printendlines	227
\setprintlines: Added \setprintlines for use by \printlines	191
\skipnumbering: Added \skipnumbering and supports	122
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	93
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	95
Using \sublinenumrep instead of \sublinenumr@p	95
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	205
General: eledmac having been available for 2 years, deleted the commented out original edmac texts	1
Maïeul Rouquette new maintainer	1
Made macros of all messages	76
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	83
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	19

Possibility to number the pstart with the commands <code>\numberpstarttrue</code>	1
v0.12.1.	
General: Don't number <code>\pstarts</code> of stanza.	1
The numbering of <code>\pstarts</code> restarts on each <code>\beginnumbering</code>	1
v0.13.0.	
<code>\managestanza@modulo</code> : New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	300
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	50
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.	260
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.	152
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	299
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code>	193
General: Line numbering can be reset at each <code>pstart</code>	90
Possibility to print <code>\pstart</code> number inside.	19
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	299
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	125
General: <code>\lemma</code> can contain commands.	27
Debug in lineation command	21
New generic commands to customize footnote display.	35, 245
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code>	25
Options of <code>\Xfootnotes</code>	162
Possibility to have commands in sidenotes.	59
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.	37
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code>	219
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	245
General: Add <code>\labelpstarttrue</code>	19
Add <code>\Xnumberonlyfirstintwolines</code>	37
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	39
New hook to add arbitrary code at the beginning of the notes	44
New options for block of notes.	46
New package option: <code>parapparatus</code>	1
New tools to change order of series	244
Sectioning commands.	66

v1.2.0.	
\Xprenotes:	Debug in familiar footnotes (bug introduced by v1.1). 219
\endquote:	Compatibility of \ledchapter with the <i>memoir</i> class. 337
v1.3.0.	
\endquote:	<i>Quotation</i> and quote environment inside numbered sections. 337
v1.4.0.	
\edtext:	Compatibility of \edtext with the right-to-left direction (with Polyglossia). 126
\ledsetnormalparstuffX:	Direction of footnotes with polyglossia. 216
\newseries@:	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. 234
\rbracket:	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). 185
General:	Compatibility with LuaTeX of RTL notes. 1
v1.4.1.	
\affixside@note:	Remove spurious spaces. 282
\endquote:	New option <i>noquotation</i> 337
\get@thisfootnote:	Compatibility of standard footnotes with eledmac when these footnotes contain any commands. 198
\labelrefsparsesubline:	Fix bug with \edlabel. 262
v1.4.2.	
General:	Debug with some special classes. 1
v1.4.3.	
General:	Add \Xnonbreakableafternumber. 40
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. 337
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. 204
v1.5.0.	
\do@insidelinehook:	Added \do@insidelinehook for use in \do@oline 147
\edindex:	Compatibility with imakeidx package, and possibility to use multiple index with \edindex. 291
General:	Correct indexing when the call is made in critical notes. 287
v1.5.1.	
\managestanza@modulo:	Correct stanzaindentsrepetition counter 300
\normalvfootnoteX:	Fix bug with normal familiar footnotes when mixing RTL and LTR text. 201
v1.6.0.	
\newverse:	Add \falseverse macro. 303
v1.6.1.	
\AtStartEveryPstart:	Spurious space in \pstart. 139

\ifinserthangingsymbol: Hang verse is now not automatically flush right.	299
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill inside.	144
\pend: Spurious space in \pend.	140
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.	67
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts.	337
\get@index@command: Debug \get@index@command and compatibility with hyperref package.	290
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work.	248
\prevpage@num: Correct \parafootsep when using with ledgroup.	177
General: Compatibility with parledgroup option ofeledpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	287
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	221
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.	287
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes.	254
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup.	283
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	337
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.	43
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.	104
\AtStartEveryPstart: New optional argument for \pstart, to execute code before it.	139
\edindex: Use correctly default index when imakeidx is loaded.	291
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	337
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar.	85
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	276
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in eledmac, including needs for eledpar case.	277
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	83
\l@dunhbox@line: \do@line is split in more little commands.	145
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	248
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	248
\pend: New optional argument for \pend, to execute code after it.	140
\stanza: &can have an optional argument: content to be printed after.	303
\Stanza can have an optional argument: content to be printed before.	303
Add \newverse macro, \falseverse deprecated.	303
General: Add \ledinnernote and \ledouternote commands.	58
Add \Xendparagraph and related settings.	48
Add hyperlink to crossref (needs hyperref package).	54
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	288
New hooks: \Xafterrule and \afterruleX	47
New options for ragged-paragraph notes	46
New sectioning commands.	66
Optional arguments for \pstart and \pend.	18

v1.12.1.

\wrap@edcrossref: Fix spurious spaces.	265
--	-----

v1.12.2.

\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	144
--	-----

v1.12.3.

\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	118
\flag@start send a error message when a \edtext is done without insert (note)	118
\reledmac@error: Replaced error messages	77
General: Add macros for new messages since v0.7	76
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.	58
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	293
\@ssect: Debug \eledchapter in parallel with memoir	340
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	148
\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	87
\l@dgobblearg: \l@dgobblearg becomes \l@gobbeloptarg	320
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	321
\resumenumbering: Debug \resumenumbering	88
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	265
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	118
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	248
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	211
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	245
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	48
Added widthliketwocolumns option	73
v1.13.1.	
\ifat@every@pstart@star@: Add \l@dzeroopenalties in \pstart	139
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	83
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	291
General: Debug spurious characters before endnotes.	221
Delete previous override of \l@d@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	295
Move gobbling command	83
Provide \@gobblefour	83
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref	
package.	343
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	340
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	126
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	211
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	120
\if@lemmacommand@: New boolean \iflemmacommand@.	131

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	28
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.	117
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.	126
Compatibility of standard footnotes with some biblatex styles.	1
New <code>\stanzaindent</code> command.	1
v1.16.1.	
<code>\xlineref</code> : <code>\lineref</code> is not defined if defined by some other package, like <code>lineno</code> . <code>Eledmac</code> provides <code>\edlineref</code> instead.	266
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph. . . .	126
v1.18.0.	
<code>\@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools + hyperref</code>	293
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code> . . .	260
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes.	222
<code>\l@dnumpstartsL</code> : Add <code>\ifl@dprintingpages</code> and <code>\@dprintingcolumns</code> for <code>eledpar</code>	83
<code>\print@line</code> : Compatibility with <code>Lua\TeX</code> RTL languages.	145
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code>	187
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns (<code>eledpar</code>)	185
General: Add <code>\Xpstarteverytime</code>	39
Compatibility with <code>Lua\TeX</code> RTL languages.	1
Debug <code>\Xonlypstart</code> when using <code>\Xnumberonlyfirstinline</code> and the current line number differs from the previous.	39
v1.19.0.	
<code>\footssplitskips</code> : <code>\footssplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages.	164
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty. .	268
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>	194
<code>\stanza</code> : <code>&</code> and <code>&</code> can be preceded by spaces.	303
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (fix bug added in 1.19.0).	268
General: Add <code>\Xendboxlinenum</code>	41
Add <code>\Xtwolines</code> and <code>\Xmorethantwolines</code> hooks	38
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>	25
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	293
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	222
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code>	291
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code>	216
<code>\n@num</code> : <code>\n@num@ref</code> deleted	111
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	111
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	247
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	248
<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>	218
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	244
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	122
Correct <code>\skipnumbering</code> for stanza.	122
Delete <code>\skipnumbering@reg</code>	122
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	244
Add <code>\applabel</code> and related	56
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column.	1
Add <code>\hidenumbering</code>	23
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks	45
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code>	39
Add <code>\Xparindent</code> and <code>\hangindentX</code>	44
Add <code>\Xtwolinesbutnotmore</code> and <code>\Xtwolinesonlyinsamepage</code>	1
Add <code>nocritical</code> , <code>noend</code> , <code>nofamiliar</code> and <code>noledgroup</code> options.	1
Add <code>noeledsec</code> package option	1
Debug <code>\beforenotesX</code> <code>\maxhnotesX</code> <code>\noteswidthliketwocolumnsX</code> and <code>\afterruleX</code> with footnotes set in two and three columns.	1
Fix bug when a <code>\Xfootnote</code> follows a <code>\Xendnote</code> in the second argument of <code>\edtext</code> (bug added in <code>eledmac 1.0.0</code>).	1

Fix bug with <code>\maxnotesX</code> when using <code>\foottwocolX</code> or <code>\footthreecolX</code>	1
Fix bug with space between columns with notes in two columns (bug added in v1.13.0).	1
Fix spurious space after first page number in <code>\doendnotes</code> . <code>oldprintnpnumspace</code> option allows to come back to previous setting	1
<code>parapparatus</code> option works now with familiar footnotes.	1
Provide <code>\@gobblefive</code>	83
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	291
General: Add <code>\doendnotesbysection</code> command.	25
Add option for lemma separator inside endnotes	42
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.	25
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>	177
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>	126
<code>\Serefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code>	271
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	130
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	234
<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.	134
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code>	41
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code>	41
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.	28
Fix bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code>	28
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.	28
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.	285
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	275
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	285
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	337
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some conde of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	285
<code>\newseries@</code> : One endnotes file by series.	240
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined	257
<code>\doxtrafeeti</code> becomes <code>\Xdo@feet</code> ; <code>\doxtrafeetii</code> becomes <code>\do@Xfeet</code> ; <code>\@opxtrafeeti</code> becomes <code>\@opfeetX</code> ; <code>\doreinxtrafeetii</code> becomes <code>\X@doreinfeet</code> ; <code>\doreinxtrafeeti</code> becomes <code>\@doreinfeetX</code>	257
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	197
Suppress indexing command specific to <code>memoir</code>	291
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.	216
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.	162
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>	38

Option <code>linrangesep</code> for critical footnotes and endnotes.	38
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>	271
<code>\edlabel</code> : Fix bug when calling <code>\edlabel</code> in a footnotes of the rightside	260
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments.	222
<code>\printlines</code> : <code>\printlines</code> takes an eighth argument: the line flag	193
<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.	266
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	56
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>	178
<code>\l@d@section</code> : <code>\endnotes</code> take seven arguments.	222
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>	48
New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</code>	43

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>	48
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.	222
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> .	186

General: Add <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> .	1
Fix bug in <code>\ledlsnotefontsetup</code> and <code>\ledrsnotefontsetup</code> which could not handle <code>\color</code> command properly.	1
More specific error messages.	1
New hooks: <code>\Xwrapcontent</code> , <code>\Xendwrapcontent</code> and <code>\wrapcontentX</code> .	44
New hooks: <code>\Xwraplemma</code> and <code>\Xendwraplemma</code> .	43
v2.10.1.	
General: Add ‘nopenalties’ option.	1
Fix bug introduced in v. 1.4: not paragraphed critical footnotes could prevent marginal line number from being displayed	1
v2.11.0.	
<code>\do@actions@fixedcode</code> : Add action 1010	151
General: Add new tools to produce an apparatus of manuscripts	1
Fix bug in <code>\Xparafootsep</code> in parallel typesetting	1
Make <code>\parafootsepX</code> work	1
Prevent <code>\Xtxtbeforenotes</code> hook from causing notes to go beyond the bottom margin	1
v2.12.0.	
General: <code>\preXnotes</code> becomes <code>\Xprenotes</code> (naming convention)	1
Add <code>\hidenumberingonleftpage</code> and <code>\hidenumberingonrightpage</code>	1
Add <code>\toendnotes</code> and related.	1
Add <code>auxdir</code> option.	1
Fix bug in critical and familiar footnotes when using uppercase letters with accent mark	1
Fix bug when using <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting in combination with the <code>noeledsec</code> option.	1
Fix bug with <code>\ledinnernote</code> and <code>\ledouternote</code> in parallel typesetting	1
Fix bug with familiar footnote number in optional argument of <code>\pstart</code> or <code>\pend</code> in parallel typesetting	1
Fix spurious vertical space in <code>\chapter</code> when used as optional argument of <code>\pstart</code> in parallel typesetting.	1
Make endnote compatible with <code>\sameword</code> mechanism	1
More accurate message to control the position of <code>\Xfootnote</code> and <code>\applabel</code> in the \TeX code	1
v2.13.0.	
General: Version 2.13.0 never existed.	1
v2.13.1.	
General: In critical footnotes, the right side flag is printed only if requested explicitly with <code>\Xlineflag</code> (bug added in v. 2.5.0).	1
v2.13.2.	
General: Fix bug added in v. 11.2 which could make parallel typesetting not work.	1
v2.13.3.	
General: Makes <code>\Xendafterpagenumbe</code> affecting <code>\Serefwithpage</code>	1
v2.14.0.	
General: Hyperref with the line number inside critical footnotes is correct when using <code>\xxref</code>	1
Some internal changes for new features of <code>reledpar</code> .	1
v2.14.1.	
General: Fix bug when using <code>\footnoteX</code> in the first argument of <code>\edtext</code> .	1
v2.14.1a.	
General: Fix problematic typos in the handbook.	1

v2.15.0.	
General: Add ‘byline’ arrangement.	1
Fix <code>\Xtxtbeforenotes</code> in <code>ledgroup</code>	1
v2.15.1.	
General: Fix <code>\edindex</code> in tabular environments.	1
v2.15.2.	
General: Fix bug with <code>fancyhdr</code> package 3.8 and later.	1
v2.15.3.	
General: Fix bug with <code>\section</code> in optional argument of <code>\pstart</code> and empty line before <code>\pend</code> (bug added in v2.8.2).	1
Simplification of the sectioning command code.	1
v2.16.0.	
General: <code>\Xdo@feet</code> becomes <code>\do@Xfeet</code>	257
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1
Add <code>\Xpagelinesep</code> hook.	40
Compatibility with new features of <code>reledpar</code>	1
Deleted dead code.	1
Display a warning message if using a version of \TeX that is too old.	1
Fix bug with <code>\Xgroupbylines</code> for notes in two columns	1
Fix bug with <code>\Xtxtbeforenotes</code> for notes in three or two columns	1
Fix bug with ‘notenumber’ option of <code>indextools</code> package when indexing texts in familiar footnotes.	1
Fix potential bug when using <code>\edindex</code> in critical footnotes.	1
More explicit error message in case the stanza indentation is not defined.	257
New options for <code>\fnpos</code> and <code>\mpfnpos</code> to set a customized order for familiar and critical footnotes.	1
When <code>\edindex</code> is called outside of a <code>\beginnumbering... \endnumbering</code> structure, it is automatically switched to <code>\index</code> , with a warning message.	1
When indexing texts in familiar footnotes with <code>\edtext</code> , refer to the line number where the footnote is called.	1
When indexing texts in sidenotes with <code>\edtext</code> , refer to the line number where the sidenote is called.	1
v2.16.1.	
General: Fix bug with redefinition of the style of the footnote number (bug added in v2.12.0)	1
v2.16.2.	
General: Error message if <code>footmisc</code> is loaded after <code>reledmac</code>	1
Fix bug introduced by v2.16.1 when using non-expandable control sequence, like <code>\normalfont</code> , in the footnote number style.	1
v2.16.3.	
General: Fix bug with <code>\Seref</code> (bug added in v2.7.0).	1
v2.16.4.	
General: Fix bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.15.3).	1
v2.16.5.	
General: Fix potential spurious spaces in endnotes.	1

v2.16.6.	
General: Fix bug with the line number style in <code>\doennotes</code> when referring to right side line in parallel typesetting.	1
Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code>	1
v2.16.7.	
General: Fix bug with <code>\msdata</code> when using multiple <code>\beginnumbering... \endnumbering</code>	1
Fix bug with <code>\numberpstarttrue</code> when using multiple <code>\beginnumbering... \endnumbering</code>	1
v2.16.8.	
General: Fix bug with <code>\edindex</code> in footnotes in parallel typesetting.	1
v2.17.0.	
General: Add <code>\edglsadd</code> command.	1
Add <code>\setmsdataposition</code> setting.	1
v2.17.1.	
General: Fix spurious space in paragraphed footnotes when using <code>Lua[®]TeX</code> without using Right-To-Left text.	1
v2.17.2.	
General: Change log message when numbered files still don't exist, in order to improve compatibility with <code>latexmk</code>	1
v2.17.3.	
General: Fix bug with <code>\doendnotesbysection</code> and <code>\doendnotes</code>	1
v2.17.4.	
General: Fix bug with <code>\setSerefonlypageprefixsingle</code> and <code>\setSerefonlypageprefixmore</code>	1
v2.17.5.	
General: Fix bug with <code>\pstartref</code> when referring to the left side in parallel typesetting. . .	1
v2.18.0.	
General: Fix bug when using a <code>\edtext</code> in two lines or more in right-to-left typesetting with <code>X_gTeX</code>	1
Fix bug when using both <code>\Xnumberonlyfirstintwolines</code> or <code>\Xnumberonlyfirstinline</code> and <code>\Xparafootsep</code> and <code>\Xsymlinenum</code>	1
v2.18.1.	
General: Fix bug when using <code>\msdata</code> with <code>Lua[®]TeX</code> or with the <code>hyperref</code> package. . . .	1
v2.19.0.	
General: Add <code>\footnoteXmark</code> and <code>\footnoteXtext</code> commands.	1
Add better compatibility with the <code>csquotes</code> package when using familiar footnotes. . .	1
Fix bug with paragraph indent after sectioning command.	1
v2.20.0.	
General: Add <code>\AtStartEveryStanza</code> , <code>\BeforeEveryStopStanza</code> , <code>\AtEndEveryPend</code> , <code>\AtStartEveryPstart</code>	1
Add second optional argument of <code>\pstart</code> , <code>\pend</code> and <code>\stanza</code>	1
Add starred version of <code>\AtEveryPstart</code> , <code>\AtEveryPend</code> , <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code>	1
Add third and fourth optional argument of <code>\newverse</code>	1
Fix bug when using familiar footnotes in <code>\eledsection</code> and related.	1
Reset font specification at the beginning of familiar footnotes.	1