

reledmac

Typeset scholarly editions with L^AT_EX*

Maïeul Rouquette[†]

based on the original ledmac by

Peter Wilson

Herries Press

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

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

Abstract

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

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

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

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

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

Contents

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

*This file (**reledmac.dtx**) has version number v2.13.2, last revised 2016/06/24.

[†]maieul at maieul dot net

1.2.4 <code>reledmac</code>	14
1.3 List of works edited with (r)(e)ledmac	14
2 How the package works	14
3 Compatibility warning	14
4 Options	15
4.1 Specific features	15
4.2 Optimizing package performance	15
5 Text lines and paragraphs numbering	16
5.1 Text lines numbering	16
5.2 Paragraphs	16
5.2.1 Basics	16
5.2.2 Automatically producing <code>\pstart ... \pend</code>	17
5.2.3 Content before specific <code>\pstart</code> and after specific <code>\pend</code>	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>)	18
5.2.6 Languages written in Right to Left	18
5.2.7 Memory limits	18
5.3 Lineation commands	19
5.3.1 Disabling lineation	19
5.3.2 Setting lineation start and step	19
5.3.3 Setting lineation reset	20
5.3.4 Setting line number margin	20
5.3.5 Other settings	20
5.4 Changing the line numbers	21
5.4.1 Sublineation	21
5.4.2 Locking lineation	21
5.4.3 Setting and changing line number	21
5.4.4 Line number style	22
5.4.5 Skipping and hiding number	22
5.4.6 Execute code at each line	22
6 Apparatus commands	23
6.1 Terminology	23
6.2 Critical notes	23
6.2.1 The lemma	23
6.2.2 Footnotes	24
6.2.3 Endnotes	24
6.2.4 Paragraph in critical apparatus	26
6.2.5 Change lemma and line number	26
6.2.6 Changing the names of commands for critical apparatus	27
6.3 Disambiguation of identical words in the apparatus	27
6.3.1 Basic use	27

6.3.2 Notes about input encoding with UTF-8 processor	28
6.3.3 Use with <code>\lemma</code> command	28
6.3.4 Customizing	29
6.4 Apparatus of Manuscripts	30
6.4.1 Marking sections of text	30
6.4.2 Layout of the apparatus of manuscripts	30
6.4.3 Settings	31
6.5 Familiar notes	31
6.5.1 Basic use	31
6.5.2 Customizing mark	31
6.5.3 Separator for multiple footnotes	32
6.6 Changing series	32
6.6.1 Create a new series	32
6.6.2 Delete series	32
6.6.3 Series order	32
6.7 Position of critical and familiar footnotes	32
7 Critical apparatus appearance	33
7.1 Notes arrangement in a series	33
7.2 Control line number printing	34
7.2.1 Print line number only at first time	34
7.2.2 Arbitrary text before line number	34
7.2.3 Separator for line range	34
7.2.4 Abbreviate line range	34
7.2.5 Disable line number	35
7.2.6 Printing pstart number	35
7.2.7 Printing stanza number	36
7.2.8 Separator between line and subline numbers	36
7.2.9 Space around number	36
7.2.10 Space around line symbol	36
7.2.11 Space in place of number	37
7.2.12 Boxing line number and line symbol	37
7.3 For endnotes	38
7.4 Arbitrary code around line number	38
7.5 Separator between the lemma and the note	38
7.5.1 For footnotes	38
7.5.2 For endnotes	39
7.6 Font style	39
7.6.1 For line number	39
7.6.2 For the lemma	39
7.6.3 For all notes	40
7.7 Wrapping notes	40
7.7.1 Wrapping lemmas	40
7.7.2 Wrapping contents	40
7.8 Indent of notes content	40
7.9 Arbitrary code at the beginning of notes	41

7.10 Arbitrary code before inserting note	41
7.11 Options for footnotes in columns	41
7.11.1 Alignment	41
7.11.2 Size of the columns	42
7.12 Options for paragraphed footnotes	42
7.12.1 Mark separation of notes	42
7.12.2 Ragged text	42
7.13 Options for block of notes	42
7.13.1 Text before notes	42
7.13.2 Code before notes	42
7.13.3 Spacing	43
7.13.4 Rule	43
7.13.5 Maximum height	43
7.13.6 Width	44
7.14 Footnotes and the <code>reledpar</code> columns	44
7.15 Endnotes in one paragraph	44
8 Fonts	44
9 Verse	45
9.1 Basic	45
9.2 Define stanza indents	45
9.3 Repeating stanza indents	46
9.4 Manual stanza indent	46
9.5 Stanza breaking	47
9.6 Hanging symbol	47
9.7 Long verse and page break	47
9.8 Content before/after verses	47
9.9 Numbering stanza	48
9.10 Various tools	48
9.11 Notes on empty lines	48
10 Grouping	49
11 Cross referencing	49
11.1 Basic use	49
11.2 Cross-referencing to a critical note	50
11.3 Cross-referencing which return a number in any case	50
11.3.1 Cross-referencing in order to define line number of a critical note	51
11.4 Not automatic cross-referencing	51
11.5 Normal \LaTeX cross-referencing	51
11.6 References to start and end lines	51
11.6.1 Reference to main text lines	51
11.6.2 References to lines that are commented on in the apparatus	52
11.6.3 Settings	52
11.7 Compatibility with <code>xr</code> package	54

12 Side notes	54
12.1 Basics	54
12.2 Setting	54
12.2.1 Width	54
12.2.2 Vertical position	54
12.2.3 Distance to the main text	55
12.2.4 Separator between notes	55
13 Indexing	55
13.1 Basics	55
13.2 Referring to critical notes	55
13.3 Separator between page and line numbers	56
13.4 Using xindy	56
13.5 Advanced setting	57
14 Glossary	57
14.1 Preamble setting	57
14.2 Commands	57
15 Tabular material	58
16 Sectioning commands	61
16.1 Sectioning commands without line numbers or critical notes	61
16.2 Sectioning commands with line numbering and critical notes	61
16.3 Optimization	62
17 Quotation environments	62
18 Page breaks	62
18.1 Control page breaking	62
18.2 Prevent page break in a long verses	63
19 Miscellaneous	63
19.1 Known and suspected limitations	64
19.1.1 Non-standard geometry	64
19.1.2 floatrow package compatibility	64
19.1.3 ‘No room for a new’	64
19.1.4 Marginal notes	64
19.1.5 Paragraph shape	65
19.1.6 Paragraphed footnotes	65
19.1.7 Use with other packages	65
19.1.8 Parallel typesetting	66
I Implementation overview	67

II Preliminaries	67
II.1 Links with original edmac	67
II.2 Package declaration	67
II.3 Package options	68
II.4 Loading packages	70
II.5 Compatibility with Lua \TeX	70
II.6 Boolean flags	70
II.7 Messages	71
II.8 Gobbling	77
II.9 Miscellaneous commands	77
II.10 Prepare reledpar	78
II.11 Booleans provided by other optional packages which are required in any case	79
III Sectioning commands	79
IV List macros	83
V Line counting	84
V.1 Choosing the system of lineation	84
V.2 Line number margin	86
V.3 Line number initialization and increment	87
V.4 Line number locking	88
V.5 Line number style	89
V.6 Line number printing	89
V.7 Line number counters and lists	90
V.8 Line number locking counter	92
V.9 Line number associated to lemma	92
V.10 Reading the line-list file	95
V.11 Commands within the line-list file	97
V.12 Writing to the line-list file	109
VI Marking text for notes	115
VI.1 <code>\edtext</code> itself	116
VI.2 Substitute lemma	122
VI.3 Substitute line numbers	123
VI.4 Lemma disambiguation	124
VII Paragraph decomposition and reassembly	130
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	130
VII.2 Processing one line	135
VII.2.1 General process	135
VII.2.2 Process for “normal” line	136
VII.2.3 Process for line containing <code>\eledsection</code> command	137
VII.2.4 Hooks	138
VII.2.5 Sidenotes and marginal line number initialization	138

VIII Line and page number computation	139
IX Line number printing	142
X Pstart number printing in side	146
XI Restoring footnotes and penalties	147
XI.1 Add insertions to the vertical list	148
XI.2 Penalties	149
XI.3 Printing leftover notes	149
XI.4 Text before notes	150
XII Critical footnotes	151
XII.1 Fonts	151
XII.2 Individual note options	152
XII.3 Notes language	152
XII.4 General survey of the way we manage notes	153
XII.5 General setup	154
XII.6 Footnotes arrangement	154
XII.6.1 User level macro	154
XII.6.2 Normal footnote	155
XII.6.3 Paragraphed footnotes	159
XII.6.4 Columnar footnotes	166
XII.7 Critical notes presentation	172
XII.7.1 Font tools	172
XII.7.2 Pstart number in footnote	173
XII.7.3 Lemma printing	173
XII.7.4 Line number printing	174
XIII Familiar footnotes	183
XIII.1 Adjacent footnotes	183
XIII.2 Regular footnotes for numbered texts	184
XIII.3 Footnote formats	186
XIII.4 Footnote arrangement	187
XIII.4.1 User level macro	187
XIII.4.2 Normal footnotes	187
XIII.4.3 Two columns footnotes	193
XIII.4.4 Three columns footnotes	195
XIII.4.5 Paragraphed footnotes	197
XIII.5 Wrapping footnote marks in hyperlink	201
XIV Code common to both critical and familiar footnote in normal arrangement	202
XV Footnotes' width for two columns	203
XVI Footnotes' order	204

XVII Footnotes' rule	204
XVIII Specific skip for first series of footnotes	205
XVIII.0.1 Overview	205
XVIII.0.2 User level command	206
XVIII.0.3 Internal commands	206
XIX Endnotes	207
XIX.1 Internal commands	207
XIX.2 User level commands	211
XIX.2.1 Inserting contents to endnotes	211
XIX.2.2 Printing endnotes	212
XX Generate series of notes	216
XX.1 Test if series is still existing	217
XX.2 Init specific to <code>reledpar</code>	217
XX.3 For critical footnotes	217
XX.3.1 Options	217
XX.3.2 Create inserts, needed to add notes in foot	219
XX.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc.	219
XX.3.4 Set standard display	221
XX.4 For familiar footnotes	222
XX.4.1 Options	222
XX.4.2 Create tools for familiar footnotes (<code>\footnotex</code>)	222
XX.5 The endnotes	224
XX.5.1 The auxiliary file	224
XX.5.2 The main macro	224
XX.5.3 Tools	225
XX.5.4 Internal commands	225
XX.5.5 The options	226
XX.6 Init standards series (A,B,C,D,E)	227
XXI Setting series display	227
XXI.1 Change series order	227
XXI.2 Test series order	228
XXI.2.1 Get the first series	228
XXI.3 Series setting	228
XXI.3.1 General way of working	228
XXI.3.2 Tools to set options	229
XXI.3.3 Tools to generate options commands	230
XXI.3.4 Options for critical notes	232
XXI.3.5 Options for familiar notes	233
XXI.3.6 Options for endnotes	234
XXI.4 Hooks for a particular footnote	235
XXI.5 Alias	236

XXII Output routine	237
XXII.1 Page number management	237
XXII.2 Extra footnotes output	237
XXII.3 Patching standard output of commands	239
XXIII Cross referencing	242
XXIII.1 Compatibility with xref	255
XXIV Side notes	255
XXV Minipages and such	263
XXVI Indexing	268
XXVI.1 Looking on package order	268
XXVI.2 Auxiliary macros for \edindex	268
XXVI.3 Code specific to \edindexin critical footnotes	269
XXVI.4 Analysis of command in indexed text	270
XXVI.5 Code for the formatted index	271
XXVI.6 Main code	271
XXVI.7 Hyperlink	272
XXVI.8 ‘innote’ and ‘notenumber’ option of indextols package	275
XXVII Glossaries	276
XXVIII Verse	277
XXVIII.1 Hanging symbol management	277
XXVIII.2 Using & character	278
XXVIII.3 Code category setting	278
XXVIII.4 Stanza count and indent	279
XXVIII.5 Numbering stanza	280
XXVIII.6 Stanza number in note	281
XXVIII.7 Main work	282
XXVIII.8 Restore catcode and penalties	284
XXIX Apparatus of Manuscripts	284
XXIX.1 User level macro	284
XXIX.2 Setting macro	285
XXIX.3 Counters and lists	286
XXIX.4 Auxiliary file macros	286
XXIX.5 Action macro	287
XXIX.6 Inserting footnote	291
XXIX.7 Other	292

XXX Arrays and tables	292
XXX.1 Preamble: macro as environment	292
XXX.2 Tabular environments	295
XXX.2.1 Disabling and restoring commands	296
XXX.2.2 Counters, boxes and lengths	299
XXX.2.3 Tabular typesetting	303
XXX.2.4 Environments	314
XXXI Quotation's commands	314
XXXII Section's title commands	315
XXXII.1 Commands to disable some feature	315
XXXII.2 General overview	316
XXXII.3 \beforeeledchapter command	316
XXXII.4 Auxiliary commands	317
XXXII.5 Patching standard commands	318
XXXII.6 Main code of \eledxxx commands	323
XXXII.7 Macros written in the auxiliary file	325
XXXIII Page breaking or no page breaking depending of specific lines	327
XXXIV Long verse: prevents being separated by a page break	329
XXXV Tools for hyperref package	330
XXXVI Compatibility with eledmac	330
Appendix A Things to do when changing versions	333
Appendix A.1 Migrating from edmac to ledmac	333
Appendix A.2 Migration from ledmac to eledmac	334
Appendix A.3 Migration to eledmac 1.5.1	335
Appendix A.4 Migration to eledmac 1.12.0	335
Appendix A.5 Migration to eledmac 17.1	336
Appendix A.6 Migration to eledmac 1.21.0	336
Appendix A.6.1 \Xledsetnormalparstuffand\ledsetnormalparstuffX	336
Appendix A.6.2 Endnotes	336
Appendix A.7 Migration to eledmac 1.22.0	336
Appendix A.8 Migration to eledmac 1.23.0	336
Appendix A.9 Migration from eledmac to reledmac	337
Appendix A.9.1 Risk of 'no room for a new'	337
Appendix A.9.2 Multiple indices with memoir	337
Appendix A.9.3 Deprecated commands and options	337
Appendix A.9.4 \renewcommandreplaced by command	338
Appendix A.9.5 Commands the names of which have been changed	338
Appendix A.9.6 Endnotes	340
Appendix A.9.7 Z Series	340
Appendix A.9.8 Internal commands	340

Appendix A.10 Migration to <code>reledmac</code> 2.1.0	340
Appendix A.11 Migration to <code>reledmac</code> 2.1.3	340
Appendix A.12 Migration to <code>reledmac</code> 2.3.0	340
Appendix A.13 Migration to <code>reledmac</code> 2.4.0	341
Appendix A.14 Migration to <code>reledmac</code> 2.5.0	341
Appendix A.15 Migration to <code>reledmac</code> 2.7.0	341
Appendix A.16 Migration to <code>reledmac</code> 2.7.2	341
Appendix A.17 Migration to <code>reledmac</code> 2.8.0	341
Appendix A.18 Migration to <code>reledmac</code> 2.13.1	341
References	343
Index	343
Change History	389

1 Introduction

1.1 Aim of the package

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

- automatic stepped line numbering, by page, section or paragraph;
- sub-lineation within the main series of line numbers;
- variant readings automatically keyed to line numbers;
- caters to both prose and verse;
- multiple series of footnotes and endnotes;
- block or columnar formatting of the footnotes;
- simple tabular material may be line numbered;
- indexing keyed to page and line numbers.

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

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

Any suggestions for new features are welcome.

This manual contains a general description of how to use `reledmac` followed by the complete source code and its extensive documentation (in sections I and following,

enumerated with Roman numerals). It ends with a list of actions to do when migrating from one version to other, a change history and an index to the source code.

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

1.2 History

1.2.1 `edmac`

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

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

The present version was developed in the summer of 1990, with the intent of adding necessary features, streamlining and documenting the code, and further generalizing it to make it easily adaptable to the needs of editors in different disciplines. John did most of the general reworking and documentation, with the financial assistance of the Division of the Humanities and Social Sciences, California Institute of Technology. Dominik adapted the code to the conventions of Frank Mittelbach’s `doc` option, and added some documentation, multiple-column footnotes, cross-references, and crop marks.¹ A description by John and Dominik of this version of `edmac` was published as ‘An overview of `edmac`: a PLAIN \TeX format for critical editions’, *TUGboat* 11 (1990), pp. 623–643.

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

As of 1994, we were pleased to be able to say that `edmac` was being used for the real-life book production of several interesting editions, such as the Latin texts of Euclid’s *Ele-*

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).

ments,² an edition of the letters of Nicolaus Copernicus,³ Simon Bredon's *Arithmetica*,⁴ a Latin translation by Plato of Tivoli of an Arabic astrolabe text,⁵ a Latin translation of part II of the Arabic *Algebra* by Abū Kāmil Shujā' b. Aslam,⁶ the Latin *Rithmachia* of Werinher von Tegernsee,⁷ a middle-Dutch romance epic on the Crusades,⁸ a seventeenth-century Hungarian politico-philosophical tract,⁹ an anonymous Latin compilation from Hungary entitled *Sermones Compilati in Studio Generali Quinqueecclesiensi in Regno Ungarie*,¹⁰ the collected letters and papers of Leibniz,¹¹ Theodosius's *Spherics*, the German *Algorismus* of Sacrobosco, the Sanskrit text of the *Kāśikāvṛtti* of Vāmana and Jayāditya,¹² and the English texts of Thomas Middleton's collected works.

1.2.2 ledmac

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

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

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

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

1.2.3 eledmac

Important changes were put in version 1.0, to make `ledmac` more easily extensible (see 7 p. 33). These changes can trigger small problems with the old customization. That is why a new name was selected: `eledmac` (extended `ledmac`).

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 *Rithmachia* 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.

To migrate from `ledmac` to `eledmac`, please read Appendix A.2 p. 334.

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. 337.

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

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

2 How the package works

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

3 Compatibility warning

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

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

Please read 7.13.5 p. 43.

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. 338).

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. 62).

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. 56).

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. 61).

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. 49). 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`. 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. 45) or `reledpar` (see 19.1.8 p. 66).

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
`\AtEveryPend` printed before every `\pstart` begins / after every `\pend` ends.

5.2.5 Numbering paragraphs (`\pstart`)

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

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

On each `\beginnumbering` the numbering restarts.

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

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

5.2.6 Languages written in Right to Left

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

5.2.7 Memory limits

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

`\pausenumbering` `reledmac` stores a lot of information about line numbers and footnotes in memory
`\resumenumbering` as it goes through a numbered section. But at the end of such a section, it empties its

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

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

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

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

```
\newcommand{\memorybreak}{\pausenumbering\resumenumbering}
```

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

5.3 Lineation commands

5.3.1 Disabling lineation

`\numberlinefalse` 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

`\linenumberlist`

line numbers. For example:

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

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

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

the standard numbering sequence is applied. The standard sequence is that specified by the combination of the `firstlinenum`, `linenumincrement`, `firstsublinenum` and `linenumincrement` counter values.

5.3.3 Setting lineation reset

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

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

5.3.4 Setting line number margin

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

```
\linenummargin{left}
```

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

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

5.3.5 Other settings

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

5.4 Changing the line numbers

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

5.4.1 Sublineation

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

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

You can change the separator between line number and subline number or using `\Xsublinesep` without any option argument (7.2.8 p. 36 or using `\Xsublinesepside`. But in the second case, it will change the separator only for line number in side, not for the footnotes.

5.4.2 Locking lineation

`\startlock` `\endlock` 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. 45.

`\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` `\advanceline` 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.

5.4.6 Execute code at each line

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

6 Apparatus commands

6.1 Terminology

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

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

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

So :

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

6.2 Critical notes

6.2.1 The lemma

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

```
\edtext{⟨lemma⟩}{⟨commands⟩}
```

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

For example:

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

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

6.2.2 Footnotes

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

`\Afootnote` Five separate series of the footnotes are maintained; each macro takes one argument like `\Afootnote{\text}`. When all of the six are used, the A notes appear in a layer just below the main text, followed by the rest in turn, down to the E notes at the bottom. 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 6.6.1 p. 32.

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

- `fulllines` to disable `\Xtwolines` and `\Xmorethantwolines` features for this note (cf. 7.2.4 p. 34).
- `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. 38).
- `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}`.

`\Eendnote` `option` can contain a comma-separated list of values. Allowed values are:

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 7.2.4 p. 34).
- `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. 39).

- `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. 39).

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

`toendnotes` **Code between endnotes** Sometimes, it is useful to insert content between endnotes of the same series: for example to separate endnotes of different sections of the same text. In this case, you could use *inside numbered text* the command:
`Xtoendnotes` `\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 case, you may want to write once-expanded¹⁴ version of the $\langle content \rangle$, that is the version where the commands are expanded on the first level. This can be, for example, to get a counter value. Use the starred version in this case. For example:

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

6.2.4 Paragraph in critical apparatus

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

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

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

6.2.5 Change lemma and line number

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

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

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

1 Smith] Jones C, D.

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

\linenum You can use `\linenum{ $\langle arg \rangle$ }` to change the line numbers passed to the notes. $\langle arg \rangle$ 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{ $\langle start page \rangle$  |  $\langle s. line \rangle$  |  $\langle s. sub-l. \rangle$  |  $\langle end p. \rangle$  |  $\langle e. l. \rangle$  |  $\langle e. sub-l. \rangle$  |  $\langle font \rangle$  | }
```

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

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.

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. 49) 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.

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`.

6.3.2 Notes about input encoding with UTF-8 processor

If you use UTF-8 processor, like $\text{Xe}\text{\LaTeX}$ or $\text{Lua}\text{\LaTeX}$, there should not be any glitches. However, pay attention to how characters are encoded. Similar-looking characters may be represented differently in unicode numbering.

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

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

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

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

- If you use $\text{Xe}\text{\LaTeX}$, add this line in your preamble: `\XeTeXinputnormalization 1`.
- If you use $\text{Lua}\text{\LaTeX}$, use the `uninormalize` package of Michal Hoftich¹⁶ with the `buffer` option set to true.

With these tools, $\text{Xe}\text{\LaTeX}$ / $\text{Lua}\text{\LaTeX}$ will dynamicaly 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`:

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

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

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

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

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

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

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

The diagram consists of three nested rectangular boxes. The innermost box contains the text "1_{inlemma}" followed by "2". The middle box contains "3₂" followed by "4". The outermost box contains "5₁". Above the middle box is the superscript "1...3", and above the outermost box is the superscript "1...5". Below the "5" in the outermost box is the subscript "1".

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

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

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

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

6.3.4 Customizing

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

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

6.4 Apparatus of Manuscripts

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

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

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

6.4.1 Marking sections of text

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

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

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

`\stopmsdata` `\stopmsdata` must be inserted at the point where the section of text previously marked by `\msdata` ends. The command must be attached right to the end of the section, with no space. As `\stopmsdata` is a `ℒTEX` 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. 33). However, the following *additional* commands are available.

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

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

6.5 Familiar notes

6.5.1 Basic use

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

6.5.2 Customizing mark

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

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

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

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

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

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

There are similar command triples for the other series.

6.5.3 Separator for multiple footnotes

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

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

6.6 Changing series

6.6.1 Create a new series

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

6.6.2 Delete series

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

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

6.6.3 Series order

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

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

6.7 Position of critical and familiar footnotes

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

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

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

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

7 Critical apparatus appearance

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

When a length, noted $\langle l \rangle$, is used, it can be stretchable: *a* plus *b* minus *c*. The final length *m* is calculated by L^AT_EX 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 the footnotes of a series as a single paragraph. If you use this arrangement, you are strongly encouraged to read 19.1.6 p. 65.
- `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.

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

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

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

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

7.2 Control line number printing

7.2.1 Print line number only at first time

`\Xnumberonlyfirstinline`

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

`\Xnumberonlyfirstintwolines`

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

`\Xsymlinenum`

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

`\Xendnumberonlyfirstinline`

`\Xendnumberonlyfirstintwolines`

`\Xendsymlinenum`

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

7.2.2 Arbitrary text before line number

`\Xbeforenumber`

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

7.2.3 Separator for line range

`\Xlinerangeseparator`

`\Xendlinerangeseparator`

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.

7.2.4 Abbreviate line range

`\Xtwolines`

`\Xmorethantwolines`

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

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

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

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

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

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

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

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

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. 24.

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

7.2.5 Disable line number

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

7.2.6 Printing pstart number

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

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

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

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

7.2.7 Printing stanza number

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

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

`\Xstanzaseparator`

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

7.2.8 Separator between line and subline numbers

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

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

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

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

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

7.2.9 Space around number

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

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

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

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

7.2.10 Space around line symbol

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

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

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

7.2.11 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.

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

7.2.12 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}
```

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

`\Xendboxsymlinenum` `\Xendboxsymlinenum[\langle s \rangle]{\langle l \rangle}` 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[\langle s \rangle]{\langle text \rangle}` where `\langle text \rangle` 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[\langle s \rangle]{\langle l \rangle}` will box the start line number in a box of length `\langle l \rangle`. The content will be put at the right of the box.
- `\Xboxendlinenum[\langle s \rangle]{\langle l \rangle}` will box the dash plus the end line number or the range symbol in a box of length `\langle l \rangle`. 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[\langle s \rangle]{\langle l \rangle}`, `\Xendboxlinenumalign[\langle s \rangle]{\langle text \rangle}`, `\Xendboxstartlinenum[\langle s \rangle]{\langle l \rangle}`, `\Xendboxendlinenum[\langle s \rangle]{\langle l \rangle}` are the same as, respectively, `\Xboxlinenum` and `\Xboxendlinenum`, `\Xboxstartlinenum`, `\Xboxendlinenum` except in endnotes.

7.3 For endnotes

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

7.4 Arbitrary code around line number

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

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

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.

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>\Xlemmadisablefontselection</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>\Xlemmadisablefontselection</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 T _E X 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 L ^A T _E X 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 L ^A T _E X 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 L ^A T _E X 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 L ^A T _E X command. For example, with the <code>bidi</code> package, to ensure having a lemma written right to left, use <code>\Xwraplemma{\RL}</code> .
<code>\Xwrapendlemma</code>	<code>\Xendwraplemma[⟨s⟩]{⟨cmd⟩}</code> is the same for endnotes.

7.7.2 Wrapping contents

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

7.8 Indent of notes content

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

7.9 Arbitrary code at the beginning of notes

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

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

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

7.10 Arbitrary code before inserting note

`\Xbeforeinserting` and `\beforeinsertingX` are very technical commands.

They allow one to add any arbitrary code just before the footnotes are added in the list of footnotes. The main use is to insert text direction code. For example, if you edit right-to-left text with bidi, 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. 40)).

Note that the changes are local to the footnote.

7.11 Options for footnotes in columns

7.11.1 Alignment

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

<code> must be one of the following command:

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

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

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

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

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

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

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

7.11.2 Size of the columns

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

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

7.12 Options for paragraphed footnotes

7.12.1 Mark separation of notes

<code>\Xafternote</code>	You can add some horizontal space after a note by using <code>\Xafternote[⟨s⟩]{⟨l⟩}</code> (for critical footnotes) or <code>\afternoteX[⟨s⟩]{⟨l⟩}</code> (for familiar footnotes). The default value is <code>1em plus.4em minus.4em</code> .
<code>\afternoteX</code>	
<code>\Xparafootsep</code>	For paragraphed footnotes (see below), you can choose the separator between each note by using <code>\Xparafootsep[⟨s⟩]{⟨text⟩}</code> for critical notes and <code>\parafootsepX</code> for familiar notes. A common separator is the double pipe (<code> </code>), which you can set by using <code>\Xparafootsep{\$\parallel\$}</code> .
<code>\parafootsepX</code>	

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

7.12.2 Ragged text

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

7.13 Options for block of notes

7.13.1 Text before notes

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

7.13.2 Code before notes

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

7.13.3 Spacing

`\Xbeforenotes` You can change the vertical space before the rule of the critical notes with `\Xbeforenotes[⟨s⟩]{⟨l⟩}`. The default value is 1.2em plus .6em minus .6em.

Be careful, the standard L^AT_EX footnote rule used by reledmac decreases by 3pt. This 3pt decrease is not changed by this command.

`\beforenotesX` You can change the vertical space printed before the rule of the familiar notes with `\beforenotesX[⟨s⟩]{⟨l⟩}`. The default value is 1.2em plus .6em minus .6em.

Be careful, the standard L^AT_EX footnote rule, which is used by reledmac, decreases 3pt. These 3pt are not changed by this command.

`\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.

7.13.4 Rule

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

Be careful, the standard L^AT_EX footnote rule, which is used by reledmac, adds 2.6pt. These 2.6pt are not changed by this command.

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

Be careful, the standard L^AT_EX footnote rule, which is used by reledmac, adds 2.6pt. These 2.6pt are not changed by this command.

7.13.5 Maximum height

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

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

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

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

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

7.13.6 Width

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

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

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

Note that changes the width of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.11.2 p. 42.

7.14 Footnotes and the reledpar columns

`\Xnoteswidthliketwocolumns`
`\noteswidthliketwocolumnsX`

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

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

7.15 Endnotes in one paragraph

`\Xendparagraph`

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

`\Xendafternote`

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

`\Xendsep`

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

8 Fonts

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

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

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

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

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

`\select@lemmfont` We will briefly discuss `\select@lemmfont` 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@lemmfont` 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@lemmfont` 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@lemmfont` selects the appropriate font for the note using that font specifier.

`reledmac` uses `\select@lemmfont` 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. 47.

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. 335.

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

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

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

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

9.4 Manual stanza indent

```
\stanzaindent
\stanzaindent*
```

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

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

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

9.5 Stanza breaking

`\setstanzapenalties`

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

The command

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

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

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

9.6 Hanging symbol

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

`\sethangingsymbol`

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

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

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

9.7 Long verse and page break

If you want to prevent page breaks inside long verses, use the option `nopbinverse` when loading package, or use `\lednopbinversetrue`. Read 18.2 p. 63 for further details.

9.8 Content before/after verses

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

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

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

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

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

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

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

9.9 Numbering stanza

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

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

You can use this feature in combination with `\Xstanza` (7.2.7 p. 36).

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

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

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

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

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

9.10 Various tools

`\ampersand` If you need to print an `&` symbol in a stanza, use the `\ampersand` macro, not `\&` which will end the stanza.

`\flagstanza` Putting `\flagstanza[⟨len⟩]{⟨text⟩}` at the start of a line in a stanza (or elsewhere) will typeset `⟨text⟩` at a distance `⟨len⟩` before the line. The default `⟨len⟩` is `\stanzaindentbase`.

9.11 Notes on empty lines

Since v2.3.0 of `reledmac`, empty lines when typesetting verses no longer produce new paragraphs, and consequently, do not insert vertical spaces. Use optional argument of `\stanza` or `\newverse` to insert vertical space (9.8 p. 47).

10 Grouping

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

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

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

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

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

`\begin{ledgroupsize}[\langle pos \rangle]{\langle width \rangle}`.

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

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

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

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

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

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

11 Cross referencing

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

11.1 Basic use

`\edlabel` First you place a label in the text using the command `\edlabel{\langle lab \rangle}`. `\langle lab \rangle` can be almost anything you like, including letters, numbers, punctuation, or a combination—anything but spaces; you might type `\edlabel{toves-3}`, for example.¹⁹

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

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

`\pstartref`

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

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

The `\edlabel` command works by writing macros to `ℳℒℒ. aux` file. You will need to process your document through `ℳℒℒ` 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. 52).

11.3 Cross-referencing which return a number in any case

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

Where #1 stands for the reference.

However, there are situations in which you will want `reledmac` to return a number without displaying any warning messages about undefined labels or the like: if you want to use the reference in a context where `ℳℒℒ` 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. 26).

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

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

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

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

The `\xxref{<lab1>}{<lab2>}` command generates a reference to a sequence of lines, for use in the second argument of `\edtext`. It takes two arguments, both of which are labels: e.g., `\xxref{mouse}{elephant}`. It calls `\linenum` (q.v., 6.2.5 p. 26 above) and sets the beginning page, line and subline numbers to those of the place where `\edlabel{mouse}` was placed, and the ending numbers to those where `\edlabel{elephant}` occurs.

11.4 Not automatic cross-referencing

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

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

11.5 Normal \LaTeX cross-referencing

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

11.6 References to start and end lines

11.6.1 Reference to main text lines

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

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

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

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

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

`\Serefwithpage` `\Serefwithpage` will make a cross-reference printed as a reference in critical 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. 26) to refer to these labels, `reledmac` will use your line settings to refer to the passage.

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

11.6.3 Settings

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

For example, you may use:

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

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

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

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

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

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

Also note that `\setSerefonlypageprefixsingle` is only a shortcut for `\Xendbeforepagenumber` (see 11.6.3 p. 52). So if you use `\Xendbeforepagenumber` without any optional argument, it will override this setting.

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

These commands are the following:

- `\Xlineflag` (for `reledpar`), enabled by default.

- `\Xlinerangeseparator`
- `\Xmorethantwolines`
- `\Xsublinesep`
- `\Xtwolines`
- `\Xtwolinesbutnotmore`
- `\Xtwolinesonlyinsamepage`

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

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

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

For one specific command When calling `\appref` and `\SEref`, you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical footnotes (6.2.2 p. 24).

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

11.7 Compatibility with xr package

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

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

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

12 Side notes

12.1 Basics

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

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

`\ledleftnote` `\ledsidenote{⟨text⟩}` will put `⟨text⟩` into the margin specified by the current setting of `\sidenotemargin{⟨location⟩}`. The permissible value for `⟨location⟩` is one out of the list `left`, `right`, `inner`, or `outer`, for example `\sidenotemargin{outer}`.
`\ledrightnote` The package's default setting is
`\ledsidenote` `\sidenotemargin{right}`
`\sidenotemargin`

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

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

12.2 Setting

12.2.1 Width

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

12.2.2 Vertical position

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

12.2.3 Distance to the main text

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

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

These can of course be changed to suit.

12.2.4 Separator between notes

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

13 Indexing

13.1 Basics

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

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

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

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

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

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

13.2 Referring to critical notes

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

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

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

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

13.3 Separator between page and line numbers

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

- is the default separator used by the MAKEINDEX program.

Consequently, if you want to use an other `\pagelinesep`, you have to configure your `.ist` index style file. For example if you use `:` as separator²⁰.

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

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

13.4 Using xindy

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

pagenumber-linenummer

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

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>.

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

13.5 Advanced setting

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

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

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

14 Glossary

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

14.1 Preamble setting

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

- Or set `glossaries`:
`\glsSetCompositor{-}`
- Or set `reledmac`:
`\renewcommand{\pagelinesep}{-}`
In this case, the above will have consequences for your use of `\edindex` and you should set your `.ist` file (13.3 p. 56).

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>]
```

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

15 Tabular material

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

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

```

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

```

1	2	3
a	bb	ccc
AAA	BB	C

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

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

For example:

```

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

```

produces the following parallel pair of verses.

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

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

`\spreadmath` `\spreadmath{<math>}` typesets `{<math>}` but the `{<math>}` has no effect on the calculation of column widths. `\spreadtext{<text>}` is the analogous command for use in edtabular environments.

```

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

```

$$\begin{array}{cccc}
1 & 2 & 3 & 4 \\
& F + G + C & & \\
a & bb & ccc & dddd
\end{array}$$

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

```

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

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	\sqcup		d
A	B	C	D

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

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

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

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

For example:

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

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

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

`\edvertdots`

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

```

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

```

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

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

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

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

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

`\showlemma` The lemma within the text is printed via `\showlemma{lemma}`. Normally, or with the ‘final’ option, the definition of `\showlemma` is:
`\newcommand*{\showlemma}[1]{#1}`
 so it just produces its argument. With the ‘draft’ option it is defined as
`\newcommand*{\showlemma}[1]{\textit{#1}}`
 so that its argument is typeset in an italic font, which may make it easier to check that all lemmas have been treated.

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

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

19.1 Known and suspected limitations

19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.13.5 p. 43 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. 62 in order to know which are the available options.

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

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

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

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

19.1.4 Marginal notes

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

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

19.1.5 Paragraph shape

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

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

19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.1 p. 33, and described in more detail on XII.6.3 p. 162, really is a nuisance if that is something you need to do. There are some possible solutions, described by Michael Downes, but this area remains unsatisfactory.

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

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

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

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

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

Finally, in many cases you should use `\Xmaxhnotes` and / or `\maxhnotesX` (7.13.5 p. 43), 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 \LaTeX whinging ‘Missing { inserted’, and then things start to fall apart. The trick in this case is to specify either:

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

or

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

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

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

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

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

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

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

19.1.8 Parallel typesetting

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

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

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

I Implementation overview

We present the `reledmac` code in roughly the order in which it is used during a run of \TeX . The order is *exactly* that in which it is read when you load The `Eledmac` package, because the same file is used to generate this manual and to generate the \LaTeX package file.

Most of what follows consists of macro definitions, but there are some commands that are executed immediately—especially at the start of the code. The documentation generally describes the code from the point of view of what happens when the macros are executed, though. As each macro is introduced, its name is printed in the margin.

After package options, we begin with the commands you use to start and stop line numbering in a section of text (Section II). Next comes the machinery for writing and reading the auxiliary file for each section that helps us count lines, and for creating list macros encoding the information from that file (Section V); this auxiliary file will be read at the start of each section, to create those list macros, and a new version of the file will be started to collect information from the body of the section.

Next are commands for marking sections of the text for footnotes (Section VI), followed by the macros that take each paragraph apart, attach the line numbers and insertions, and send the result to the vertical list (Section VII). The footnote commands (Section XII) and output routine (Section XXII) finish the main part of the processing; cross-referencing (Section XXIII) and endnotes (Section XIX) complete the story.

In what follows, macros with an `@` in their name are more internal to the workings of `reledmac` than those made up just of ordinary letters, just as in `PLAIN \TeX` (see *The TeXbook*, p. 344). You are meant to be able to make free with ordinary macros, but the ‘`@`’ ones should be treated with more respect, and changed only if you are pretty sure of what you are doing.

II Preliminaries

II.1 Links with original `edmac`

Generally, these are the modifications to the original. `edmac` code:

- Replace as many `\def`’s by `\newcommand`’s as possible to avoid overwriting \LaTeX macros.
- Replace user-level \TeX counts by \LaTeX counters.
- Use the \LaTeX font handling mechanisms.
- Use \LaTeX messaging and file facilities.

II.2 Package declaration

Announce the name and version of the package, which is targetted for `LaTeX2e`.

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledmac}[2016/06/24 v2.13.2 typeset critical editions]%
4 %

```

II.3 Package options

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

```

39 \newif\ifparapparatus@
40 \DeclareOptionX{parapparatus}{\parapparatus@true}
41
42 \newif\iflednopbinverse
43 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
44
45 \newif\ifwidthliketwocolumns%
46 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
47
48 \newif\ifcontinuousnumberingwithcolumns
49 \DeclareOptionX{continuousnumberingwithcolumns}{\
50 continuousnumberingwithcolumnstrue}%
51
52 \newif\ifxindy@
53 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
54   \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
55   \newwrite\eledmac@xindy@out%
56   \xindy@true%
57   \gdef\eledmacmarkuplocdepth{:depth 1}%
58   \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
59 }%
60
61 \newif\ifxindyhyperref@
62 \DeclareOptionX{xindy+hyperref}{%
63   \xindyhyperref@true%
64 }%
65
66 \newif\ifeledmaccompat%
67 \DeclareOptionX{eledmac-compat}{%
68   \eledmaccompat@true%
69 }%
70 \DeclareOptionX{nopenalties}{%
71   \AtBeginDocument{\let\add@penalties\relax}%
72 }
73 \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 $\text{Lua}\text{T}_{\text{E}}\text{X}$ or $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$ 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 {}%
85 {\RequirePackage{etex}%
86  \csname reserveinserts\endcsname{32}%
87 }
88 \RequirePackage{suffix}
89 \RequirePackage{xstring}
90 \RequirePackage{ifluatex}
91 \RequirePackage{ragged2e}
92 \RequirePackage{ifxetex}%
93 %

```

II.5 Compatibility with $\text{Lua}\text{T}_{\text{E}}\text{X}$

Here, we enable some primitives for $\text{Lua}\text{T}_{\text{E}}\text{X}$.

```

94 \ifx\directlua\undefined\else%
95   \directlua{tex.enableprimitives("",{"textdir","pardir","bodydir"})}
96 \fi
97 %

```

II.6 Boolean flags

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

```

98 \newif\ifl@dmemoir
99 \@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
100
101 %

```

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

```

102 \newif\if@ledgroup%
103 %

```

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

```

104 \newif\ifl@imakeidx
105 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value
106 %

```

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

```

107 \newif\ifl@indextools%
108 \@ifpackageloaded{indextools}{%
109   \l@indextoolstrue%
110   \l@imakeidxtrue%
111   \let\imki@wrindexentry\indtl@wrindexentry%
112 }{}%
113 %

```

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

`\if@RTL` The `\if@RTL` is defined by the `bidi` package, which is sometimes loaded by *polyglossia*. But we define it as well if the `bidi` package is not loaded.

```

114 \ifdef{\if@RTL}{-}{\newif\if@RTL}
115 %

```

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

```

116 \newif\if@firstlineofpage%
117 \newif\if@firstlineofpageR%
118 %

```

II.7 Messages

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

`\reledmac@warning` Write a warning message.

```

119 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
120 %

```

`\reledmac@error` Write an error message.

```

121 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
122 %

```

```

\led@err@NumberingStarted 123 \newcommand*{\led@err@NumberingStarted}{%
d@err@NumberingNotStarted 124   \reledmac@error{Numbering has already been started}{\@ehc}}
umberingShouldHaveStarted 125 \newcommand*{\led@err@NumberingNotStarted}{%
126   \reledmac@error{Numbering was not started}{\@ehc}}
127 \newcommand*{\led@err@NumberingShouldHaveStarted}{%
128   \reledmac@error{Numbering should already have been started}{\@ehc}}
129 %

```

```

\led@err@edtextoutsidestart30 \newcommand*\led@err@edtextoutsidestart{%
131   \reledmac@error{\string\edtext\space outside numbered paragraph (\...pstart
\pend)}{\@ehc}}%
132 %

```

```

\led@mess@NotesChanged33 \newcommand*\led@mess@NotesChanged{%
134   \typeout{\reledmac reminder: }%
135   \typeout{ The number of the footnotes in this section
136     has changed since the last run.}%
137   \typeout{ You will need to run LaTeX two more times
138     before the footnote placement}%
139   \typeout{ and line numbering in this section are
140     correct.}}
141 %

```

```

\led@mess@SectionContinued42 \newcommand*\led@mess@SectionContinued}[1]{%
143   \message{Section #1 (continuing the previous section)}%
144 %

```

```

\led@err@LineationInNumbered45 \newcommand*\led@err@LineationInNumbered{%
146   \reledmac@error{You can't use \string\lineation\space within
147     a numbered section}{\@ehc}}
148 %

```

```

\led@warn@BadLineation49 \newcommand*\led@warn@BadLineation{%
\led@warn@BadLinenummargin50   \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp51 \newcommand*\led@warn@BadLinenummargin{%
\led@warn@BadSublockdisp52   \reledmac@warning{Bad \string\linenummargin\space argument}}
153 \newcommand*\led@warn@BadLockdisp{%
154   \reledmac@warning{Bad \string\lockdisp\space argument}}
155 \newcommand*\led@warn@BadSublockdisp{%
156   \reledmac@warning{Bad \string\sublockdisp\space argument}}
157 %

```

```

\led@warn@NoLineFile58 \newcommand*\led@warn@NoLineFile}[1]{%
159   \reledmac@warning{Can't find line-list file #1}}
160 %

```

```

\led@warn@LineFileObsolete61 \newcommand*\led@warn@Obsolete}[1]{%
162   \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
163 %

```

```

\led@warn@BadAdvancelineSubline 64 \newcommand*{\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine 65 \reledmac@warning{\string\advanceline\space produced a sub-line
166 number less than zero.}}
167 \newcommand*{\led@warn@BadAdvancelineLine}{%
168 \reledmac@warning{\string\advanceline\space produced a line
169 number less than zero.}}
170 %

\led@warn@BadSetline 71 \newcommand*{\led@warn@BadSetline}{%
\led@warn@BadSetlinenum 72 \reledmac@warning{Bad \string\setline\space argument}}
173 \newcommand*{\led@warn@BadSetlinenum}{%
174 \reledmac@warning{Bad \string\setlinenum\space argument}}
175 %

\led@err@PstartNotNumbered 76 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart 77 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered 78 numbered section %
\led@err@PendNoPstart 79 (\string\...beginnumbering\string\endnumbering)}{\@ehc}}%
\led@err@AutoparNotNumbered 80 \newcommand*{\led@err@PstartInPstart}{%
\led@err@NumberingWithoutPstart 81 \reledmac@error{\string\pstart\space encountered while another
182 \string\pstart\space was in effect}{\@ehc}}
183 \newcommand*{\led@err@PendNotNumbered}{%
184 \reledmac@error{\string\pend\space must be used within a
185 numbered section}{\@ehc}}
186 \newcommand*{\led@err@PendNoPstart}{%
187 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
188 \newcommand*{\led@err@AutoparNotNumbered}{%
189 \reledmac@error{\string\autopar\space must be used within a
190 numbered section}{\@ehc}}
191 \newcommand*{\led@err@NumberingWithoutPstart}{%
192 \reledmac@error{\string\beginnumbering...\string\endnumbering\space
without \string\pstart}{\@ehc}}%
193 %

\led@warn@BadAction 94 \newcommand*{\led@warn@BadAction}{%
195 \reledmac@warning{Bad action code, value \next@action.}}
196 %

\led@warn@DuplicateLabel 97 \newcommand*{\led@warn@DuplicateLabel}[1]{%
\ppLabelOutSecondArgEdtext 98 \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined 99 \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined 100 }%
201 \newcommand*{\led@warn@AppLabelOutSecondArgEdtext}[1]{%
202 \reledmac@warning{\string\applabel\space outside of the second argument
of an \string\edtext\space `#1' on page \the\pageno.}}%

```

```

203 \newcommand*\led@warn@RefUndefined}[1]{%
204   \G@refundefinedtrue%
205   \reledmac@warning{Reference `#1' on page \the\pageno\space undefined.%
206     Using `000'.}%
207   \@latex@warning{Reference `#1' undefined\on@line}%
208 }%
209 \newcommand*\led@warn@pairRefUndefined}[1]{%
210   \G@refundefinedtrue%
211   \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \the\
212     pageno\space undefined.
213     Using `??'.}%
214   \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
215 }%

```

```

\led@warn@NoMarginpars16 \newcommand*\led@warn@NoMarginpars{%
217   \reledmac@warning{You can't use \string\marginpar\space in numbered text
218   }}%

```

```

\led@warn@BadSidenotemargin19 \newcommand*\led@warn@BadSidenotemargin{%
220   \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
221 %

```

```

\led@warn@NoIndexFile22 \newcommand*\led@warn@NoIndexFile}[1]{%
223   \reledmac@warning{Undefined index file #1}}
224 %

```

```

\led@warn@SeriesStillExist25 \newcommand*\led@warn@SeriesStillExist}[1]{%
226   \reledmac@warning{Series #1 is still existing !}%
227 }%
228 %

```

```

\led@err@ManySidenotes29 \newcommand*\led@err@ManySidenotes{%
\led@err@ManyLeftnotes30   \ifledRcol{%
\led@err@ManyRightnotes31     \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
space p. \the\page@numR}%
232     \else%
233     \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
space p. \the\page@num}%
234     \fi%
235   }%
236   \newcommand*\led@err@ManyLeftnotes{%
237     \ifledRcol{%

```

```

238 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@numR\
space p. \the\page@numR}%
239 \else%
240 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\
space p. \the\page@num}%
241 \fi%
242 }%
243 \newcommand{\led@err@ManyRightnotes}{%
244 \ifledRcol@%
245 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
246 \else%
247 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
space p. \the\page@num}%
248 \fi%
249 }%
250 %

```

```

\led@err@TooManyColumns51 \newcommand*\led@err@TooManyColumns}{%
\led@err@UnequalColumns52 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn53 \newcommand*\led@err@UnequalColumns}{%
\led@err@HighEndColumn54 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns55 in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
256 \newcommand*\led@err@LowStartColumn}{%
257 \reledmac@error{Start column is too low}{\@ehc}}
258 \newcommand*\led@err@HighEndColumn}{%
259 \reledmac@error{End column is too high}{\@ehc}}
260 \newcommand*\led@err@ReverseColumns}{%
261 \reledmac@error{Start column is greater than end column}{\@ehc}}
262 %

```

```

\endnotes@outsidenumbering63 \newcommand{\led@err@toendnotes@outsidenumbering}{%
264 \reledmac@error{\string\toendnotes\space and related commands must be
called inside a numbered texte (\string\...beginnumbering\string\endnumbering
)}{\@ehc}%
265 }%
266 %

```

```

\err@EdtextWithoutFootnote67 \newcommand{\led@err@EdtextWithoutFootnote}{%
268 \reledmac@error{edtext without Xfootnote. Check syntaxis}{\@ehc}%
269 }%
270 %

```

```

\led@err@FootnoteNotInSecondArgEdtext 71 \newcommand{\led@err@FootnoteNotInSecondArgEdtext}[1]{%
272   \reledmac@error{#1footnote outside of the second argument of an edtext.
    Check syntax}{\@ehc}%
273 }%
274 %

\led@error@ImakeidxAfterEledmac 75 \newcommand{\led@error@ImakeidxAfterEledmac}{%
276   \reledmac@error{Imakeidx must be loaded before reledmac.}{\@ehc}%
277 }%
278 %

\led@error@IndextoolsAfterEledmac 79 \newcommand{\led@error@IndextoolsAfterEledmac}{%
280   \reledmac@error{Indextools must be loaded before reledmac.}{\@ehc}%
281 }%
282 %

\led@error@fail@patch@@makecol 83 \newcommand{\led@error@fail@patch@@makecol}{%
284   \reledmac@error{Fail to patch \string\makecol\space command.}{\@ehc}%
285 }%
286 %

\led@error@fail@patch@@reinserts 87 \newcommand{\led@error@fail@patch@@reinserts}{%
288   \reledmac@error{Fail to patch \string\reinserts\space command.}{\@ehc}%
289 }%
290 %

\led@error@fail@patch@@docclearpage 91 \newcommand{\led@error@fail@patch@@docclearpage}{%
292   \reledmac@error{Fail to patch \string\docclearpage\space command.}{\@ehc}%
293   %
294 }%

\led@error@fail@patch@@iiiminipage 95 \newcommand{\led@error@fail@patch@@iiiminipage}{%
296   \reledmac@error{Fail to patch \string\iiiminipage\space command.}{\@ehc}%
297   %
298 }%

\led@error@fail@patch@endminipage 99 \newcommand{\led@error@fail@patch@endminipage}{%
300   \reledmac@error{Fail to patch \string@endminipage\space command.}{\@ehc}%
301   %
302 }%

```

```

warning@hsizeX@deprecated 303 \newcommand{\led@warning@hsizeX@deprecated}{%
304   \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
305 }%
306 %

```

```

warning@Xhsize@deprecated 307 \newcommand{\led@warning@Xhsize@deprecated}{%
308   \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
309 }%
310 %

```

```

warning@msdatawithoutstop 311 \newcommand{\led@warning@msdatawithoutstop}{%
312   \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
313 }%
314 %

```

```

ning@preXnotes@deprecated 315 \newcommand{\led@warning@preXnotes@deprecated}{%
316   \reledmac@warning@preXnotes@deprecated%
317 }%
318 %

```

II.8 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree 319 \providecommand*\@gobblethree}[3]{}
\@gobblefour 320 \providecommand*\@gobblefour}[4]{}
\@gobblefive 321 \providecommand*\@gobblefive}[5]{}
322 %

```

II.9 Miscellaneous commands

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

```

323 \ifledfinal
324   \newcommand*\showlemma[1]{#1}
325 \else
326   \newcommand*\showlemma[1]{\underline{#1}}
327 \fi
328
329 %

```

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

Initialize it as `\empty`.

```
330 \let\linenumberlist=\empty
331
332 %
```

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

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

```
333 \newcount\@l@tempcnta \newcount\@l@tempcntb
334 %
```

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
\ifpst@rtedL
\l@dnumpststartsl
335 \newif\ifl@dpairing
336 \newif\ifl@dpaging%
337 \newif\ifl@dprintingpages%
338 \newif\ifl@dprintingcolumns%
339 \newif\ifpst@rtedL
340 \newcount\l@dnumpststartsl
341 %
```

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

```
342 \newif\ifledRcol
343 \newif\ifledRcol@
344 %
```

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

```
345 \newif\ifnumberingR
346 %
```

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

```
347 \newif\ifXnote@%
348 %
```

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

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

```
349 \providebool{indtl@innote}%
350 \providebool{indtl@notenumber}%
351 %
```

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.

```
352 \newcount\section@num
353 \section@num=0
354 \let\extensionchars=\empty
355 %
```

`\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.

```
356 \newif\ifnumbering
357 %
```

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

The initializations here are trickier than they look. `\line@list@stuff` will use all of the counters that are zeroed here when it assembles the line-list and other lists of information about the lineation. But it will do all of this locally and within a group, and

when it is done the lists will remain but the counters will return to zero. Those same counters will then be used as we process the text of this section, but the assignments will be made globally. These initializations actually apply to both uses, though in all other respects there should be no direct interaction between the use of these counters and variables in the two processing steps. For parallel processing :

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

```

358 \newcommand*{\beginnumbering}{%
359   \ifnumbering
360     \led@err@NumberingStarted
361   \endnumbering
362 \fi
363 \global\numberingtrue
364 \global\advance\section@num \@ne
365 \initnumbering@reg
366 \message{Section \the\section@num }%
367 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
368 \l@dend@stuff
369 \setcounter{pstart}{1}
370 \ifl@dpairing
371   \global\l@dnumpstartsL \z@
372   \global\pst@rtedLfalse
373 %

```

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.

```

374 \else
375   \begingroup
376   \global\@afterindenttrue%In order to reestablish normal feature if the \
begingroup was not here
377   \initnumbering@quote
378   \ifwidthliketwocolumns%
379     \csuse{setwidthliketwocolumns@\columns@position}%
380     \csuse{setpositionliketwocolumns@\columns@position}%
381   \fi%
382 \fi
383 \gdef\eled@sections@@{}%
384 \if@noeled@sec\else%
385   \makeatletter\inputIfFileExists{\l@auxdir\jobname.eledsec\the\
section@num}{\makeatother}%
386   \immediate\openout\eled@sectioning@out=\l@auxdir\jobname.eledsec\the\
section@num\relax%

```

```

387 \fi%
388 }
389 \newcommand*{\initnumbering@reg}{%
390 \global\pst@rtedLfalse
391 \global\l@dnumstartsL \z@
392 \global\absline@num \z@
393 \gdef\normal@page@break{}
394 \gdef\l@prev@pb{}
395 \gdef\l@prev@nopb{}
396 \global\line@num \z@
397 \global\subline@num \z@
398 \global\@lock \z@
399 \global\sub@lock \z@
400 \global\sublines@false
401 \global\let\next@page@num=\relax
402 \global\let\sub@change=\relax
403 \resetprevline@
404 \resetprevpage@num
405 \global\stopmsdata@inserted@true%
406 }
407
408 %

```

`\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.

```

409 \def\endnumbering{%
410 \ifnumbering
411 \global\numberingfalse
412 \normal@pars
413 \ifnum\l@dnumstartsL=0%
414 \led@err@NumberingWithoutPstart%
415 \fi%
416 \ifl@dpairing
417 \global\pst@rtedLfalse
418 \else
419 \ifx\insertlines@list\empty\else
420 \global\noteschanged@true
421 \fi
422 \ifx\line@list\empty\else
423 \global\noteschanged@true
424 \fi
425 \fi
426 \ifnoteschanged@
427 \led@mess@NotesChanged
428 \fi
429 \else
430 \led@err@NumberingNotStarted
431 \fi

```

```

432 \autoparfalse
433 \if@noeled@sec\else%
434   \immediate\closeout\eled@sectioning@out%
435 \fi%
436 \ifl@dpairing\else
437   \global\l@dnumstartsl=\z@%
438   \endgroup
439 \fi
440 }
441 %

```

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

\resumenumbering

```

442 \newcommand{\pausenumbering}{%
443   \ifautopar\global\autopar@pausetrue\fi%
444   \endnumbering\global\numberingtrue}
445 %

```

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

```

446 \newcommand*{\resumenumbering}{%
447   \ifnumbering
448     \ifautopar@pause\autopar\fi
449     \global\pst@rtedltrue
450     \global\advance\section@num \@ne
451     \led@mess@SectionContinued{\the\section@num}%
452     \line@list@stuff{\jobname.\extensionchars\the\section@num}%
453     \l@dend@stuff
454     \ifl@dpairing\else%
455       \begingroup%
456       \initnumbering@quote%
457       \ifwidthliketwocolumns%
458         \csuse{setwidthliketwocolumns@\columns@position}%
459         \csuse{setpositionliketwocolumns@\columns@position}%
460       \fi%
461     \fi%
462     \ifcontinuousnumberingwithcolumns%
463       \ifdefined\line@numR%
464         \ifnum\line@numR>\line@num%
465           \expandafter\setlinenum\expandafter{\the\line@numR}%
466         \fi%
467       \ifnum\last@page@numR>\last@page@num%
468         \global\last@page@num=\last@page@numR%
469       \fi%
470     \fi%

```

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

```

471 \fi%
472 \else
473 \led@err@NumberingShouldHaveStarted
474 \endnumbering
475 \beginnumbering
476 \fi}
477
478
479 %

```

IV List macros

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

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

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

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

It could be nice to unify them to the \LaTeX 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.

```

480 \newcommand*{\list@create}[1]{%
481 \global\let#1=\empty%
482 }%
483 %

```

`\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 .

```

484 \newcommand*{\list@clear}[1]{%
485 \global\let#1=\empty%
486 }
487 %

```

`\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`.

```

488 \newtoks\led@toksa \newtoks\led@toksb
489 \global\led@toksa={\}
490 \long\def\xright@appenditem#1\to#2{%
491   \global\led@toksb=\expandafter{#2}%
492   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
493   \global\led@toksb={}}
494 %

```

\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`.

```

495 \long\def\xleft@appenditem#1\to#2{%
496   \global\led@toksb=\expandafter{#2}%
497   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
498   \global\led@toksb={}}
499 %

```

\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.

```

500 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
501 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
502
503 %

```

V Line counting

V.1 Choosing the system of lineation

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

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

\bypstart@true

\bypstart@false

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

\ifbypage@

\bypage@true

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

```

504 \newif\ifbypage@
505 \newif\ifbypstart@
506 %

```

The `\ifbypage@R` and `\ifbypstart@R` flag specify the current lineation for right side in case of using `reledpar`. They are now defined because they are used in some specific code. `reledpar` will use the line-of-section system unless instructed otherwise.

```

\ifbypage@R\newif\ifbypage@R
\ifbypstart@R\newif\ifbypstart@R
509 %

```

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

```

510 \newcommand*{\lineation}[1]{%
511 %

```

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

```

512 \ifnumbering
513 \led@err@LineationInNumbered
514 \else
515 %

```

If the argument is page.

```

516 \def\@tempa{#1}\def\@tempb{page}%
517 \ifx\@tempa\@tempb
518 \global\bypage@true
519 \global\bypstart@false
520 \unless\ifnocritical@%
521 \Xpstart[] [false]%
522 \fi%
523 %

```

If the argument is pstart.

```

524 \else
525 \def\@tempb{pstart}%
526 \ifx\@tempa\@tempb
527 \global\bypage@false
528 \global\bypstart@true
529 \unless\ifnocritical@%
530 \Xpstart%
531 \fi%
532 %

```

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

```

533 \else
534 \def\@tempb{section}
535 \ifx\@tempa\@tempb
536 \global\bypage@false
537 \global\bypstart@false
538 \unless\ifnocritical@%
539 \Xpstart[] [false]%
540 \fi%
541 %

```

In other case, it is an error.

```

542     \else
543         \led@warn@BadLineation
544     \fi
545 \fi
546 \fi
547 \fi}}
548 %

```

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`

```

549 \newcount\line@margin
550
551 \newcommand*{\linenummargin}[1]{%
552     \l@dgetline@margin{#1}%
553     \ifnum\l@dtempcntb>\m@ne
554         \ifledRcol
555             \global\line@marginR=\l@dtempcntb
556             \led@warn@setting@in@rightside{\linenummargin}%
557         \else
558             \global\line@margin=\l@dtempcntb
559         \fi
560     \fi}}
561
562 \newcommand*{\l@dgetline@margin}[1]{%
563     \def\@tempa{#1}\def\@tempb{left}%
564     \ifx\@tempa\@tempb
565         \l@dtempcntb \z@
566     \else
567         \def\@tempb{right}%
568         \ifx\@tempa\@tempb
569             \l@dtempcntb \@ne
570         \else
571             \def\@tempb{outer}%
572             \ifx\@tempa\@tempb
573                 \l@dtempcntb \tw@
574             \else
575                 \def\@tempb{inner}%
576                 \ifx\@tempa\@tempb
577                     \l@dtempcntb \thr@@
578                 \else
579                     \led@warn@BadLinenummargin
580                     \l@dtempcntb \m@ne
581                 \fi
582             \fi

```

```

583 \fi
584 \fi}
585
586 %

```

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.

```

587 \newcounter{firstlinenum}
588 \setcounter{firstlinenum}{5}
589 \newcounter{linenumincrement}
590 \setcounter{linenumincrement}{5}
591 %

```

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

```

592 \newcounter{firstsublinenum}
593 \setcounter{firstsublinenum}{5}
594 \newcounter{sublinenumincrement}
595 \setcounter{sublinenumincrement}{5}
596
597 %

```

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

```

598 \newcommand*{\firstlinenum}[1]{%
599 \ifledRcol%
600 \setcounter{firstlinenumR}{#1}%
601 \led@warn@setting@in@rightside{\firstlinenum}%
602 \else%
603 \setcounter{firstlinenum}{#1}%
604 \fi%
605 }
606 \newcommand*{\linenumincrement}[1]{%
607 \ifledRcol%
608 \setcounter{linenumincrementR}{#1}%
609 \led@warn@setting@in@rightside{\linenumincrement}%
610 \else%
611 \setcounter{linenumincrement}{#1}%
612 \fi%
613 }
614 \newcommand*{\firstsublinenum}[1]{%
615

```

```

616 \ifledRcol%
617   \setcounter{firstsublinenumR}{#1}%
618   \led@warn@setting@in@rightside{\firstsublinenum}%
619 \else%
620   \setcounter{firstsublinenum}{#1}%
621 \fi%
622 }
623 \newcommand*{\sublinenumincrement}[1]{%
624   \ifledRcol%
625     \setcounter{sublinenumincrementR}{#1}%
626     \led@warn@setting@in@rightside{\sublinenumincrement}%
627   \else%
628     \setcounter{sublinenumincrement}{#1}%
629   \fi%
630 }
631
632 %

```

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.

```

633 \newcount\lock@disp
634 \newcommand{\lockdisp}[1]{%
635   \l@dgetlock@disp{#1}%
636   \ifnum\l@dtempcntb>\m@ne
637     \global\lock@disp=\l@dtempcntb
638   \else
639     \led@warn@BadLockdisp
640   \fi}}
641 \newcommand*{\l@dgetlock@disp}[1]{
642   \def\@tempa{#1}\def\@tempb{first}%
643   \ifx\@tempa\@tempb
644     \l@dtempcntb \z@
645   \else
646     \def\@tempb{last}%
647     \ifx\@tempa\@tempb
648       \l@dtempcntb \@ne
649     \else
650       \def\@tempb{all}%
651       \ifx\@tempa\@tempb
652         \l@dtempcntb \tw@
653       \else
654         \l@dtempcntb \m@ne
655       \fi

```

```

656 \fi
657 \fi}
658
659 %

```

`\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.

```

660 \newcount\sublock@disp
661 \newcommand{\sublockdisp}[1]{\%
662 \l@getlock@disp{#1}%
663 \ifnum\l@dttempcntb>\m@ne
664 \global\sublock@disp=\l@dttempcntb
665 \else
666 \led@warn@BadSublockdisp
667 \fi}}
668
669 %

```

V.5 Line number style

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

`\linenumrep`

`\linenumr@p`

`\sublinenumberstyle`

`\sublinenumrep`

`\sublinenumr@p`

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

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

```

670 \newcommand*{\linenumberstyle}[1]{\%
671 \def\linenumrep##1{\@nameuse{@#1}{##1}}}
672 \newcommand*{\sublinenumberstyle}[1]{\%
673 \def\sublinenumrep##1{\@nameuse{@#1}{##1}}}
674 %

```

Initialise the number styles to arabic.

```

675 \linenumberstyle{arabic}
676 \let\linenumr@p\linenumrep
677 \sublinenumberstyle{arabic}
678 \let\sublinenumr@p\sublinenumrep
679
680 %

```

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

`\linenumsep`

`\numlabfont`

`\ledlinenum`

standard versions illustrate the general sort of thing that will be needed; they are based on the `\leftheadline` macro in *The TeXbook*, p. 416.

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

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

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

```

681 \newlength{\linenumsep}
682 \setlength{\linenumsep}{1pc}
683 \newcommand*{\numlabfont}{\normalfont\scriptsize}
684 \newcommand*{\ledlinenum}{%
685   \bgroup%
686   \ifluatex%
687     \texdir TLT%
688   \fi%
689   \numlabfont\linenumrep{\line@num}%
690   \ifsublines@
691     \ifnum\subline@num>0\relax
692       \unskip%
693       \Xsublinesep@side%
694       \sublinenumrep{\subline@num}%
695     \fi
696   \fi%
697   \egroup%
698 }%
699
700 \newcommand*{\leftlinenum}{%
701   \ledlinenum
702   \kern\linenumsep}
703 \newcommand*{\rightlinenum}{%
704   \kern\linenumsep
705   \ledlinenum}
706
707 %

```

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`.

```
708 \newcount\line@num
709 %
```

`\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.

```
710 \newcount\subline@num
711 %
```

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

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

`\sublines@false`

```
712 \newif\ifsublines@
713 %
```

`\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.

```
714 \newcount\absline@num
715 %
```

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.

```
716 \newcount\@lock
717 \newcount\sub@lock
718 %
```

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:

```
719 \list@create{\line@list}
720 \list@create{\insertlines@list}
721 \list@create{\actionlines@list}
722 \list@create{\actions@list}
723
724 %
```

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

```
\endpage@num
\endline@num
\endsubline@num
725 \newcount\page@num
726 \newcount\endpage@num
727 \newcount\endline@num
728 \newcount\endsubline@num
729 %
```

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

```
730 \newif\ifnoteschanged@
731 %
```

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

```
\resetprevline@ 732 \newcommand*{\resetprevline@}{%
733 \def\do##1{\global\csundef{prevline##1}}}%
734 \dolistloop{\@series}%
735 }
736 %
```

`\resetprevpage@num`

Inside the apparatus, at each note, the page number is stored in a macro called `\prevpageX@num`, where *X* is the letter of the current series. This macro is called when using `\Xparafootsep` or `\parafootsepX`. This macro must be reset at the beginning of each numbered section. The `\resetprevpage@` command resets this macro for every series.

```

\resetprevpage@  \newcommand*{\resetprevpage@num}{%
738   \def\do##1{%
739     \ifcsdef{prevpage##1@num}{%
740       \global\csname prevpage##1@num\endcsname=\z@%
741       \global\csname prevpage##1@numR\endcsname=\z@%
742     }%
743     {}%
744     \ifcsdef{##1prevpage@num}{%
745       \global\csname ##1prevpage@num\endcsname=\z@%
746       \global\csname ##1prevpage@numR\endcsname=\z@%
747     }%
748     {}%
749   }%
750   \dolistloop{\@series}%
751 }
752 %

```

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.

```

753 \newread\@inputcheck
754 \newcommand*{\read@linelist}[1]{%
755   \ifl@Rcol%
756   \list@clearing@regR%
757   \else%
758   \list@clearing@reg%
759   \fi%
760 %

```

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).

```

761 \list@clear{\maxlinesinpar@list}
762 %

```

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 `\resumenumbering`, 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.

```

763 \get@linelistfile{#1}%
764 \ifcontinuousnumberingwithcolumns
765     \global\page@numR=\page@numR\relax
766     \global\last@page@numR=\last@page@numR\relax
767     \global\page@num=\page@num\relax
768     \global\last@page@num=\last@page@num\relax
769 \fi
770 \@stopmsd%Security if last \endmsd{} is forgotten
771 \unless\ifledRcol%Get the last line of the last page
772     \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
773     \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
774 \else%
775     \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
776     \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
777 \fi%
778 \endgroup
779 %

```

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.

```

780 \ifledRcol
781     \global\page@numR=\m@ne
782     \ifx\actionlines@listR\empty
783         \gdef\next@actionlineR{1000000}%
784     \else
785         \glp\actionlines@listR\to\next@actionlineR
786         \glp\actions@listR\to\next@actionR
787     \fi
788 \else
789     \global\page@num=\m@ne
790     \ifx\actionlines@list\empty
791         \gdef\next@actionline{1000000}%
792     \else
793         \glp\actionlines@list\to\next@actionline
794         \glp\actions@list\to\next@action
795     \fi
796 \fi

```

```

797 }
798 %

```

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

```

799 \newcommand*{\list@clearing@reg}{%
800   \list@clear{\line@list}%
801   \list@clear{\insertlines@list}%
802   \list@clear{\actionlines@list}%
803   \list@clear{\actions@list}%
804   \list@clear{\linesinpar@listL}%
805   \list@clear{\linesonpage@listL}%
806 }%
807 %

```

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

```

808 \newcommand*{\get@linelistfile}[1]{%
809   \InputIfFileExists{\l@auxdir#1}{%
810     \global\noteschanged@false
811     \begingroup
812       \catcode`\[=1 \catcode`\]=2
813       \makeatletter \catcode`\^^M=9}{%
814     \led@warn@NoLineFile{\l@auxdir#1}%
815     \global\noteschanged@true
816     \begingroup}%
817 }
818 %
819 %

```

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

V.11 Commands within the line-list file

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

be introduced in a later section, among the cross-referencing commands it is associated with.)

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

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

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

```

820 \newcommand{\line@list@version}[1]{%
821   \IfStrEq{#1}{\this@line@list@version}%
822   {%
823     {\ifledRcol%
824       \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
825       \else%
826       \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
827       \fi%
828       \endinput%
829     }%
830   }%
831   %

```

`\@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`.

```

832
833 \newcommand*{\@nl}[2]{%
834   \fix@page{#1}%
835   \ifledRcol%
836     \@nl@regR%
837   \else%
838     \@nl@reg%
839   \fi%

```

```

840 }
841 \newcommand*{\@nl@reg}{%
842   \ifx\l@dchset@num\relax \else
843     \advance\absline@num \@ne
844     \set@line@action
845     \let\l@dchset@num=\relax
846     \advance\absline@num \m@ne
847     \advance\line@num \m@ne
848   \fi
849 }%
```

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

```

850 \advance\absline@num \@ne
851   \ifx\next@page@num\relax \else
852     \page@action
853     \let\next@page@num=\relax
854   \fi
855   \ifx\sub@change\relax \else
856     \ifnum\sub@change>\z@
857       \sublines@true
858     \else
859       \sublines@false
860     \fi
861     \sub@action
862     \let\sub@change=\relax
863   \fi
864 }%
```

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

```

865   \ifcase\@lock
866   \or
867     \@lock \tw@
868   \or \or
869     \@lock \z@
870   \fi
871   \ifcase\sub@lock
872   \or
873     \sub@lock \tw@
874   \or \or
875     \sub@lock \z@
876   \fi
877 }%
```

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

```

878   \ifsublines@
879     \ifnum\sub@lock<\tw@
880       \advance\subline@num \@ne
```

```

881         \fi
882     \else
883         \ifnum \@lock<\tw@
884             \advance\line@num \@ne \subline@num \z@
885         \fi
886     \fi}
887
888 %

```

`\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`.

```

889 \newcount\last@page@num
890 \last@page@num=-10000
891
892 \newcommand*{\fix@page}[1]{%
893     \ifledRcol
894         \ifnum #1=\last@page@numR
895         \else
896             \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
897             \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
898             \ifbypageR
899                 \line@numR \z@ \subline@numR \z@
900             \fi
901             \page@numR=#1\relax
902             \last@page@numR=#1\relax
903             \def\next@page@numR{#1}%
904         \fi
905     \else
906         \ifnum #1=\last@page@num
907         \else
908             \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
909             \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
910             \ifbypage@
911                 \line@num \z@ \subline@num \z@
912             \fi
913             \page@num=#1\relax
914             \last@page@num=#1\relax
915             \def\next@page@num{#1}%
916             \listxadd{\normal@page@break}{\the\absline@num}
917         \fi
918     \fi}
919 %

```

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

```

\@lopL
\@lopR
920 \newcommand*{\@pend}[1]{}
921 \newcommand*{\@pendR}[1]{}

```

```

922 \newcommand*{\@lopL}[1]{}
923 \newcommand*{\@lopR}[1]{}
924
925 %

```

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

```

926 \newcommand*{\sub@on}{\ifsublines@
927   \let\sub@change=\relax
928 \else
929   \def\sub@change{1}%
930 \fi}
931 \newcommand*{\sub@off}{\ifsublines@
932   \def\sub@change{-1}%
933 \else
934   \let\sub@change=\relax
935 \fi}
936
937 %

```

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

```

938
939 \newcommand*{\@adv}[1]{%
940   \ifsublines@
941     \ifledRcol
942       \advance\subline@numR by #1\relax
943       \ifnum\subline@numR<\z@
944         \led@warn@BadAdvancelineSubline
945         \subline@numR \z@
946       \fi
947     \else
948       \advance\subline@num by #1\relax
949       \ifnum\subline@num<\z@
950         \led@warn@BadAdvancelineSubline
951         \subline@num \z@
952       \fi
953     \fi
954   \else
955     \ifledRcol
956       \advance\line@numR by #1\relax
957       \ifnum\line@numR<\z@
958         \led@warn@BadAdvancelineLine
959         \line@numR \z@
960       \fi
961     \else
962       \advance\line@num by #1\relax

```

```

963     \ifnum\line@num<\z@
964         \led@warn@BadAdvancelineLine
965         \line@num \z@
966     \fi
967 \fi
968 \fi
969 \set@line@action}
970
971 %

```

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

```

972
973 \newcommand*{\@set}[1]{%
974     \ifledRcol
975         \ifsublines@
976             \subline@numR=#1\relax
977         \else
978             \line@numR=#1\relax
979         \fi
980         \set@line@action
981     \else
982         \ifsublines@
983             \subline@num=#1\relax
984         \else
985             \line@num=#1\relax
986         \fi
987         \set@line@action
988     \fi}
989
990 %

```

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

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

```

991
992 \newcommand*{\l@d@set}[1]{%
993     \ifledRcol
994         \line@numR=#1\relax
995         \advance\line@numR \@ne
996         \def\l@dchset@num{#1}
997     \else
998         \line@num=#1\relax
999         \advance\line@num \@ne
1000         \def\l@dchset@num{#1}
1001     \fi}
1002 \let\l@dchset@num\relax

```

```
1003
1004 %
```

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

```
1005
1006 \newcommand*{\page@action}{%
1007   \ifledRcol
1008     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1009     \xright@appenditem{\next@page@numR}\to\actions@listR
1010   \else
1011     \xright@appenditem{\the\absline@num}\to\actionlines@list
1012     \xright@appenditem{\next@page@num}\to\actions@list
1013   \fi}
1014 %
```

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

```
1015
1016 \newcommand*{\set@line@action}{%
1017   \ifledRcol
1018     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1019     \ifsublines@
1020       \@l@tempcnta=-\subline@numR
1021     \else
1022       \@l@tempcnta=-\line@numR
1023     \fi
1024     \advance\@l@tempcnta by -5000\relax
1025     \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1026   \else
1027     \xright@appenditem{\the\absline@num}\to\actionlines@list
1028     \ifsublines@
1029       \@l@tempcnta=-\subline@num
1030     \else
1031       \@l@tempcnta=-\line@num
1032     \fi
1033     \advance\@l@tempcnta by -5000\relax
1034     \xright@appenditem{\the\@l@tempcnta}\to\actions@list
1035   \fi}
1036 %
```

\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.

```
1037
1038 \newcommand*{\sub@action}{%
1039   \ifledRcol
1040     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1041   \ifsublines@
```

```

1042 \xright@appenditem{-1001}\to\actions@listR
1043 \else
1044 \xright@appenditem{-1002}\to\actions@listR
1045 \fi
1046 \else
1047 \xright@appenditem{\the\absline@num}\to\actionlines@list
1048 \ifsublines@
1049 \xright@appenditem{-1001}\to\actions@list
1050 \else
1051 \xright@appenditem{-1002}\to\actions@list
1052 \fi
1053 \fi}
1054 %

```

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

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

```

1055 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1056
1057 \newcommand*{\do@lockon}{%
1058 \ifx\next\lock@off
1059 \global\let\lock@off=\skip@lockoff
1060 \else
1061 \ifledRcol
1062 \do@lockonR
1063 \else
1064 \do@lockonL
1065 \fi
1066 \fi}
1067
1068
1069 \newcommand*{\do@lockonL}{%
1070 \xright@appenditem{\the\absline@num}\to\actionlines@list
1071 \ifsublines@
1072 \xright@appenditem{-1005}\to\actions@list
1073 \ifnum\sub@lock=\z@
1074 \sub@lock \@ne
1075 \else
1076 \ifnum\sub@lock=\thr@@
1077 \sub@lock \@ne
1078 \fi
1079 \fi
1080 \else
1081 \xright@appenditem{-1003}\to\actions@list

```

```

1082 \ifnum \@lock=\z@
1083 \@lock \@ne
1084 \else
1085 \ifnum \@lock=\thr@@
1086 \@lock \@ne
1087 \fi
1088 \fi
1089 \fi}
1090
1091 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1092 \newcommand*{\do@lockoffL}{%
1093 \xright@appenditem{\the\absline@num}\to\actionlines@list
1094 \ifsublines@
1095 \xright@appenditem{-1006}\to\actions@list
1096 \ifnum\sub@lock=\tw@
1097 \sub@lock \thr@@
1098 \else
1099 \sub@lock \z@
1100 \fi
1101 \else
1102 \xright@appenditem{-1004}\to\actions@list
1103 \ifnum \@lock=\tw@
1104 \@lock \thr@@
1105 \else
1106 \@lock \z@
1107 \fi
1108 \fi}
1109
1110 \newcommand*{\do@lockoff}{%
1111 \ifledRcol
1112 \do@lockoffR
1113 \else
1114 \do@lockoffL
1115 \fi}
1116 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1117 \global\let\lock@off=\do@lockoff
1118
1119 %

```

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

```

1120 \newcommand*{\n@num}{%
1121 \ifledRcol%
1122 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1123 \xright@appenditem{-1007}\to\actions@listR
1124 \else%

```

```

1125 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1126 \xright@appenditem{-1007}\to\actions@list%
1127 \fi%
1128 }%
1129
1130 %

```

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

```

1131 \newcommand*\n@num@stanza{%
1132 \ifledRcol%
1133 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1134 \xright@appenditem{-1008}\to\actions@listR%
1135 \else%
1136 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1137 \xright@appenditem{-1008}\to\actions@list%
1138 \fi%
1139 }
1140 %

```

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

\hidenumbering

\hidenumberingonleftpage

\hidenumberingonrightpage

```

1141 \newif\ifl@dhidenumber
1142 \newcommand*\hidenumbering{%
1143 \ifledRcol%
1144 \write\linenum@outR{\string\hide@num}%
1145 \else%
1146 \write\linenum@out{\string\hide@num}%
1147 \fi%
1148 }%
1149 \newcommand*\hide@num{%
1150 \ifledRcol%
1151 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1152 \xright@appenditem{-1009}\to\actions@listR%
1153 \else%
1154 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1155 \xright@appenditem{-1009}\to\actions@list%
1156 \fi%
1157 }
1158 \newcommand*\hidenumberingonleftpage{%
1159 \ifledRcol%
1160 \write\linenum@outR{\string\hide@num@left}%
1161 \else%
1162 \write\linenum@out{\string\hide@num@left}%
1163 \fi%
1164 }%
1165

```

```

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

```

\@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.

```

1203 \newcount\insert@count
1204 %

```

- #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.

```
1205 \newcommand*\dummy@ref}[2]{#2}
1206 %
```

\@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.

```
1207 \newcommand*\@ref}[2]{%
1208   \ifledRcol%
1209     \@ref@regR{#1}{#2}%
1210   \else%
1211     \@ref@reg{#1}{#2}%
1212   \fi%
1213 }%
1214 \newcommand*\@ref@reg}[2]{%
1215   \global\insert@count=#1\relax
1216   \global\advance\@edtext@level by 1%
1217   \loop\ifnum\insert@count>\z@
1218     \xright@appenditem{\the\absline@num}\to\insertlines@list
1219     \global\advance\insert@count \m@ne
1220   \repeat
1221 %
```

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.

```
1222 \begingroup
1223   \let\@ref=\dummy@ref
1224   \let\@lopL\@gobble
1225   \let\page@action=\relax
1226   \let\sub@action=\relax
1227   \let\set@line@action=\relax
1228   \let\@lab=\relax
1229   \let\@lemma=\relax%
1230   \let\@sw\@gobblethree%
1231   #2
1232   \global\endpage@num=\page@num
1233   \global\endline@num=\line@num
1234   \global\endsubline@num=\subline@num
1235 \endgroup
1236 %
```

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

```

1237 \xright@appenditem%
1238   {\the\page@num|\the\line@num|%
1239    \ifsublines@ \the\subline@num \else 0\fi}%
1240    \the\endpage@num|\the\endline@num|%
1241    \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1242 %

```

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.

```

1243 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@the\
@edtext@level\endcsname}%
1244 \providebool{lemmacommand@the\@edtext@level}%
1245 \boolfalse{lemmacommand@the\@edtext@level}%
1246 %

```

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

```

1247 #2%
1248 %

```

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.

```

1249 \ifnum\@edtext@level>0%
1250   \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtext@the\@edtext@level\endcsname}}%
1251   \ifcsundef{sw@list@edtext@the\@edtext@level}{\create@this@edtext@level
}\}%
1252   \letcs{\@tmp}{sw@list@edtext@the\@edtext@level}%
1253   \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}%
1254   \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1255   \global\cslet{sw@list@edtext@the\@edtext@level}{\@tmp}%
1256   \fi%
1257 %

```

Decrease edtext level counter.

```

1258 \global\advance\@edtext@level by -1%
1259 %
1260 }
1261
1262 %

```

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`.

```
1263 \newwrite\linenum@out
1264 %
```

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

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

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

```
1265 \newif\iffirst@linenum@out@
1266 \first@linenum@out@true
1267 %
```

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

```
1268 \newcommand{\this@line@list@version}{5}%
1269 %
```

`\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.

```
1270 \newcommand*{\line@list@stuff}[1]{%
1271 %
```

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

```
1272 \read@linelist{#1}%
1273 %
```

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.

```
1274 \iffirst@linenum@out@
1275 \immediate\closeout\linenum@out%
1276 \global\first@linenum@out@false%
```

```

1277 \immediate\openout\linenum@out=\l@auxdir#1\relax%
1278 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}}%
1279 \ifl@dpaging%
1280 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}}%
1281 \fi%
1282 \else
1283 %

```

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.

```

1284 \if@minipage%
1285 \leavevmode%
1286 \fi%
1287 \closeout\linenum@out%
1288 \openout\linenum@out=\l@auxdir#1\relax%
1289 \write\linenum@out{\string\line@list@version{\this@line@list@version}}
%
1290 \ifl@dpaging%
1291 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
%
1292 \fi%
1293 \fi}
1294
1295 %

```

\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.

```

1296 \newcommand*{\new@line}{%
1297 \IfStrEq{\led@pb@setting}{after}%
1298 {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1299 {\xifinlist{\the\absline@num}{\normal@page@break}%
1300 {\numgdef{\@next@page}{\c@page+\@ne}%
1301 \write\linenum@out{\string\@nl[\@next@page][\@next@page}}}%
1302 }%
1303 {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1304 }%
1305 {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1306 }%
1307 \IfStrEq{\led@pb@setting}{before}%
1308 {\numdef{\next@absline}{\the\absline@num+\@ne}%
1309 \xifinlist{\next@absline}{\l@prev@nopb}%
1310 {\xifinlist{\the\absline@num}{\normal@page@break}%
1311 {\numgdef{\nc@page}{\c@page+\@ne}%
1312 \write\linenum@out{\string\@nl[\nc@page][\nc@page}}}%
1313 }%
1314 {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1315 }%

```

```

1316     {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1317     }%
1318     {}%
1319     \IfStrEqCase{\led@pb@setting}{\before}{\relax}{\after}{\relax}}{\write\
linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1320 }
1321 %
1322 %

```

\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

```

1323 \newif\if@noneed@Footnote%
1324
1325 \newcommand*{\flag@start}{%
1326     \ifledRcol%
1327         \edef\next{\write\linenum@outR{%
1328             \string\@ref[\the\insert@countR] []}}%
1329         \next%
1330         \ifnum\insert@countR<1%
1331             \if@noneed@Footnote\else%
1332                 \led@err@EdtextWithoutFootnote%
1333             \fi%
1334         \fi%
1335     \else%
1336         \edef\next{\write\linenum@out{%
1337             \string\@ref[\the\insert@count] []}}%
1338         \next%
1339         \ifnum\insert@count<1%
1340             \if@noneed@Footnote\else%
1341                 \led@err@EdtextWithoutFootnote%
1342             \fi%
1343         \fi%
1344     \fi}%
1345 %
1346 %

```

\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.

```

1347
1348
1349 \newcommand*{\startsub}{\dimen0\lastskip
1350 \ifdim\dimen0>0pt \unskip \fi
1351 \ifledRcol \write\linenum@outR{\string\sub@on}%
1352 \else      \write\linenum@out{\string\sub@on}%
1353 \fi
1354 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1355 \def\endsub{\dimen0\lastskip
1356 \ifdim\dimen0>0pt \unskip \fi
1357 \ifledRcol \write\linenum@outR{\string\sub@off}%
1358 \else      \write\linenum@out{\string\sub@off}%
1359 \fi
1360 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1361
1362 %

```

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

```

1363 \newcommand*{\advanceline}[1]{\leavevmode%
1364 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1365 \else      \write\linenum@out{\string\@adv[#1]}%
1366 \fi%
1367 }
1368 %

```

\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.

```

1369
1370 \newcommand*{\setline}[1]{%
1371 \leavevmode%
1372 \ifnum#1<\z@
1373 \led@warn@BadSetline
1374 \else
1375 \ifledRcol \write\linenum@outR{\string\@set[#1]}%
1376 \else      \write\linenum@out{\string\@set[#1]}%
1377 \fi
1378 \fi}
1379
1380 %

```

\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.

```

1381
1382 \newcommand*{\setlinenum}[1]{%
1383   \ifnum#1<\z@
1384     \led@warn@BadSetlinenum
1385   \else
1386     \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1387     \else      \write\linenum@out{\string\l@d@set[#1]} \fi
1388   \fi}
1389
1390 %

```

`\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.

```

1391
1392 \newcommand*{\startlock}{%
1393   \ifledRcol \write\linenum@outR{\string\lock@on}%
1394   \else      \write\linenum@out{\string\lock@on}%
1395   \fi}
1396 \def\endlock{%
1397   \ifledRcol \write\linenum@outR{\string\lock@off}%
1398   \else      \write\linenum@out{\string\lock@off}%
1399   \fi}
1400 %

```

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

```

\ifl@dskipversenumber
\l@dskipnumbertrue
\l@dskipnumberfalse
\skipnumbering
1401 \newif\ifl@dskipnumber
1402 \newif\ifl@dskipversenumber%
1403 \newcommand*{\skipnumbering}{%
1404   \leavevmode%
1405   \ifledRcol%
1406     \ifinstanza%
1407       \write\linenum@outR{\string\n@num@stanza}%
1408     \else%
1409       \write\linenum@outR{\string\n@num}%
1410     \fi%
1411     \advanceline{-1}%
1412   \else%
1413     \ifinstanza%
1414       \write\linenum@out{\string\n@num@stanza}%
1415     \else%
1416       \write\linenum@out{\string\n@num}%
1417     \fi%
1418     \advanceline{-1}%
1419   \fi%
1420 }%
1421

```

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

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

VI.1 `\edtext` itself

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

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

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

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

```
1423 \list@create{\end@lemmas}
1424 %
```

`\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.

```
1425 \newcommand{\dummy@edtext}[2]{#1}
1426 %
```

`\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`.

```
1427 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1428 %
```

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

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

The first class is font-changing macros. We suppress expansion for them by letting them become equal to zero.²⁵ This is done because we want to pass into our notes the generic commands to change to roman or whatever, and not their expansions that will ask for a particular style at a specified size. The notes may well be in a smaller font, so the command should be expanded later, when the note’s environment is in effect.

A second sort to turn off includes a few of the accent macros. Most are not a problem: an accent that is expanded to an `\accent` command may be harder to read but it works just the same. The ones that cause problems are: those that use alignments— \TeX seems to get confused about the difference between alignment parameters and macro parameters; those that use temporary control sequences; and those that look carefully at what the current font is.

(The `\copyright` macro defined in `PLAIN \TeX` has this sort of problem as well, but is not used enough to bother with. That macro, and any other that causes trouble, will get by all right if you put a `\protect` in front of it in your file.)

We also need to eliminate all `reledmac` macros like `\edlabel` and `\setline` that write things to auxiliary files: that writing should be done only once. And we make `\edtext` itself, if it appears within its own argument, do nothing but copy its first argument.

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

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

```

1429 \newcommand*{\no@expands}{%
1430   \let\select@lemmafnt=0%
1431   \let\startsub=\relax \let\endsub=\relax
1432   \let\startlock=\relax \let\endlock=\relax
1433   \let\edlabel=\@gobble
1434   \let\setline=\@gobble \let\advanceline=\@gobble
1435   \let\sameword\sameword@inedtext%
1436   \let\edtext=\dummy@edtext
1437   \l@dtabnoexpands
1438   \morenoexpands}

```

²⁵Since ‘control sequences equivalent to characters are not expandable’—*The \TeX book*, answer to Exercise 20.14.

```

1439 \let\morenoexpands=\relax
1440
1441 %

```

\@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.

```

1442 \newcommand{\@tag}{}
1443 %

```

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

```

1444 \newcount\@edtext@level%
1445 \@edtext@level=0%
1446 %

```

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

```

1447 \newif\if@edtext@secondarg@%
1448 %

```

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

```

1449 \newcounter{edtext}
1450 \renewcommand{\theedtext}{edtxt@arabic{edtext}}%
1451 %

```

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

```

1452 \newcommand{\edtext}[2]{\leavevmode%
1453 %

```

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

```

1454 \ifnumberedpar@%
1455 %

```

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

```

1456 \global\advance\@edtext@level by 1%
1457 %

```

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

```

1458 \stepcounter{edtext}%
1459 %

```

By default, we do not use \lemma

```

1460 \global\@lemmacommand@false%
1461 %

```

```

1462 \begingroup%
1463 %

```

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`.

```

1464 \ifledRcol%
1465 \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1466 {\global\let\sw@inthisedtext\empty}%
1467 {\expandafter\gl@p\csname sw@list@edtextR@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1468 \else%
1469 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1470 {\global\let\sw@inthisedtext\empty}%
1471 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1472 \fi%
1473 %

```

\@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.

```

1474 \global\renewcommand{\@tag}{%
1475 \no@expands #1%
1476 }%
1477 %

```

\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`.

```

1478 \set@line%
1479 %

```

`\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`.

```

1480 \ifledRcol \global\insert@countR \z@%
1481 \else \global\insert@count \z@ \fi%
1482 %

```

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.

```

1483     \@edtext@secondarg@true%
1484     \ignorespaces #2\relax%
1485     \@edtext@secondarg@false%
1486 %

```

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

```

1487     \ifundefined{xpg@main@language}{%if not polyglossia
1488         \flag@start}%
1489         {\ifRTL\flag@end\else\flag@start\fi%
1490         }%
1491 %

```

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

```

1492     \if@lemmacommand@%
1493     \ifledRcol%
1494         \write\linenum@outR{\string\@lemma}%
1495     \else%
1496         \write\linenum@out{\string\@lemma}%
1497     \fi%
1498     \fi%
1499 %

```

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.

```

1500     \endgroup%
1501     \ifdef{\hypertarget}%
1502     {%
1503         \csedef{thisedtext@the\@edtext@level}{\theedtext}%We need one
macro by level, as #1 can contain new \edtext
1504         \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}{}}%
1505         \showlemma{#1}%
1506         \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}%
1507     }%
1508     {%
1509         \showlemma{#1}%

```

```

1510 }%
1511 %

```

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.

```

1512 \ifx\end@lemmas\empty \else%
1513 \gl@p\end@lemmas\to\x@lemma%
1514 \x@lemma%
1515 \global\let\x@lemma=\relax%
1516 \fi%
1517 \@ifundefined{xp@main@language}{%if not polyglossia
1518 \flag@end}%
1519 {\if@RTL\flag@start\else\flag@end\fi% With polyglossia, you must
    track whether the language reads left to right (English) or right to left
    (Arabic).
1520 }%
1521 %

```

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`.

```

1522 \global\@noneed@Footnotefalse%
1523 \global\advance\@edtext@level by -1%
1524 \global\@lemmacommand@false%
1525 %

```

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

```

1526 \else%
1527 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
    led@err@edtextoutsidestart%
1528 \fi%
1529 }%
1530
1531 \newcommand*{\flag@end}{%
1532 \ifledRcol%
1533 \write\linenum@outR{}}%
1534 \else%
1535 \write\linenum@out{}}%
1536 \fi}%
1537
1538 %

```

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

```
1539 \newif\ifnumberline
1540 \numberlinetrue
1541 %
```

`\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.

```
1542 \newcommand*\set@line{%
1543   \ifl@edRcol
1544     \ifx\line@listR\empty
1545       \global\noteschanged@true
1546       \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1547     \else
1548       \gl@p\line@listR\to\@tempb
1549       \xdef\l@d@nums{\@tempb|\edfont@info}%
1550       \global\let\@tempb=\undefined
1551     \fi
1552   \else
1553     \ifx\line@list\empty
1554       \global\noteschanged@true
1555       \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1556     \else
1557       \gl@p\line@list\to\@tempb
1558       \xdef\l@d@nums{\@tempb|\edfont@info}%
1559       \global\let\@tempb=\undefined
1560     \fi
1561   \fi}
1562 %
1563 %
```

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

```
1564 \newcommand*\edfont@info{\f@encoding/\f@family/\f@series/\f@shape}
1565
1566 %
```

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. 119).

```

1567 \newcommand*{\lemma}[1]{%
1568   \global\@lemmacommand@true%
1569   \global\renewcommand{\@tag}{%
1570     \no@expands #1%
1571   }%
1572   \ignorespaces%
1573 }%
1574 %

```

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

```

1575 \newcommand{\@lemma}{%
1576   \booltrue{lemmacommand@the\@edtext@level}%
1577 }%
1578 %

```

\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.

```

1579 \newif\if@lemmacommand@%
1580 %

```

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. 92): 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.

```

1581 \newcommand*{\linenum}[1]{%
1582   \xdef\@tempa{#1|||||\\noexpand\\l@d@nums}%
1583   \global\let\l@d@nums=\empty
1584   \expandafter\line@set\@tempa|\\ignorespaces}
1585 %

```

\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.

```

1586 \def\line@set#1|#2\\#3|#4\\{%
1587   \gdef\@tempb{#1}%
1588   \ifx\@tempb\empty
1589     \l@d@add{#3}%
1590   \else
1591     \l@d@add{#1}%
1592   \fi
1593   \gdef\@tempb{#4}%
1594   \ifx\@tempb\empty\else
1595     \l@d@add{|\}\line@set#2\\#4\\}%
1596   \fi}
1597 %

```

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

```

1598 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1599
1600 %

```

VI.4 Lemma disambiguation

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

The general mechanism is the following:

- **At the first run**, each `\sameword` command increments an `etoolbox` counter the name of which contains the argument of the `\sameword` commands.
- Then this counter, associated with the argument of `\sameword` is stored, with the `\@sw` command, in the auxiliary file of the current `eledmac` section (the `.1`, `.2...` file).
- **When this auxiliary file is read at the second run**, different operations are achieved:

1. Get the rank of each `\sameword` in a line (relative rank) from the rank of each `\sameword` in all the numbered section (absolute rank):
 - For each paired `\sameword` argument and absolute line number, a counter is defined. Its value corresponds to the number of times `\sameword{⟨argument⟩}+` is called from the beginning of the lineation to the end of the current line. We also store the same data for the preceding absolute line number, if it does not have `\sameword{⟨argument⟩}`.
 - For each `\sameword` having the same argument, we subtract from its absolute rank the number stored for the paired `\sameword` argument and previous absolute line number. Consequently, we obtain the relative rank.

- See the following example which explain how for same `\sameword` absolute ranks are transformed to relative rank.

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

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

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

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

In this example, all lemma information for `\edtext` is framed. The text in parenthesis is the content of critical notes associated to the preceding frame. As you can see, we have two level of `\edtext`.

The list for `\edtexts` of level 1 is $\{\{1, 2, 2, 3, 4, 3\}, \{5, 4\}\}$.

The list for `\edtexts` of level 2 is $\{\{1, 2, 2, 3\}, \{5\}\}$.

As you can see, the mandatory argument of `\sameword` does not matter: we store the rank informations for every word potentially ambiguous.

- At the second run, when a critical notes is called, we associate it to the next item of the list associated to is `\edtext` level. So, in the previous example:
 - Critical notes (A) and (B) are associated with $\{1, 2, 2, 3\}$.
 - Critical note (C) is associated with $\{1, 2, 2, 3, 4, 3\}$.
 - Critical note (D) is associated with $\{5\}$.
 - Critical note (E) is associated with $\{5, 4\}$.
- At the second run, when a critical note is printed:
 - The `\sameword` command is let `\sameword@inedtext`.

- At each call of this `\sameword@inedtext`, we step to the next element of the list associated to the note. Let it be r .
- For the word marked by `\sameword`, we calculate how many time it is called in its line. To do it:
 - * We get the absolute line number of the current `\sameword`. This absolute line number was stored with list of relative rank for the current `\edtext`. That means, in the previous example, that, if the absolute line number of `\edtext` was 1, that critical notes (A) and (B) were not associated with $\{1, 2, 2, 3\}$ but with $\{(1, 1), (2, 1), (2, 1), (3, 1)\}$. Such method to know the absolute line number associated to a `\sameword` is required because a `\edtext` can be overlap many lines, but `\sameword` can't get it.
 - * We get the value associated, when reading the auxiliary file, to the pair compose by the current marked word and the current absolute line number. Let this value be n .
- If $n > 1$, that mean the current word appears more than once time in its line. In this case, we call `\showwordrank` with the word as first argument and r as second argument. If the word is called only once, we just print it.

After theory, implementation.

`\get@sw@txt` As the argument of `\sameword` can contain active character if we use `inputenc` with `utf8` option instead of native UTF-8 engine, we store its detokenized content in a macro in order to allow dynamic name of macro with `\csname`.²⁶

Because there is a bug with `\detokenize` and \XeTeX when using non BMP characters²⁷, we detokenize only for not \XeTeX engines. In any case, in \XeTeX , a `\csname` construction can contain UTF-8 characters without a problem, as UTF-8 characters are not managed with category code, but instead read directly as UTF-8 characters.

```

1601 \newcommand{\get@sw@txt}[1]{%
1602   \ifxetex%
1603     \xdef\sw@txt{#1}%
1604   \else%
1605     \expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize{#1}}%
1606   \fi%
1607 }%
1608 %

```

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

```

1609 \newcommand{\sameword}[2][1,usedefault]{%
1610   \leavevmode%
1611   \get@sw@txt{#2}%
1612 %

```

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

²⁶See <http://tex.stackexchange.com/q/244538/7712>.

²⁷<http://sourceforge.net/p/xetex/bugs/108/>

```

1613 \unless\ifledRcol%
1614 \csnumgdef{sw@sw@txt}{\csuse{sw@sw@txt}+\@ne}%
1615 %

```

Then, write its value to the numbered file.

```

1616 \protected@write\linenum@out{}{\string\@sw{sw@txt}{\csuse{sw@sw@txt}
}}{#1}}%
1617 %

```

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

```

1618 \else%
1619 \csnumgdef{sw@sw@txt}{\csuse{sw@sw@txt}+\@ne}%
1620 \protected@write\linenum@outR{}{\string\@sw{sw@txt}{\csuse{sw@sw@txt}
}}{#1}}%
1621 \fi%
1622 %

```

And print the word.

```

1623 #2%
1624 }%
1625 %

```

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

```

\if@addsw26 \newif\if@addsw%
1627 %

```

\@sw The command printed in the auxiliary files.

```

1628 \newcommand{\@sw}[3]{%
1629 \get@sw@txt{#1}%
1630 \unless\ifledRcol%
1631 %

```

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

```

1632 \csxdef{sw@sw@txt @\the\absline@num @\the\section@num}{#2}%
1633 %

```

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

```

1634 \numdef{\prev@line}{\the\absline@num-1}%
1635 \ifcsundef{sw@sw@txt @\prev@line @\the\section@num}{%
1636 \csnumgdef{sw@sw@txt @\prev@line @\the\section@num}{#2-1}%
1637 }{}%
1638 %

```

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

```

1639 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1640 %

```

And do the same thing for the right side.

```

1641 \else%
1642 \csxdef{sw@sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1643 \numdef{\prev@line}{\the\absline@numR-1}%
1644 \ifcsundef{sw@sw@txt @\prev@line @\the\section@numR @R}{%
1645 \csnumgdef{sw@sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1646 }{}%
1647 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R
}}%
1648 \fi%
1649 %

```

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

```

1650 \@tempcnta=\@edtext@level
1651 \@whilenum{\@tempcnta>0}\do{%
1652 \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1653 {%
1654 \@addswfalse%
1655 \notbool{lemmacommand@\the\@tempcnta}%
1656 {\@addswtrue}%
1657 {\IfStrEq{#3}{inlemma}%
1658 {\@addswtrue}%
1659 {%
1660 \def\do##1{%
1661 \ifnumequal{##1}{\the\@tempcnta}%
1662 {\@addswtrue\listbreak}%
1663 }%
1664 }%
1665 \docsvlist{#3}%
1666 }%
1667 }%
1668 \if@addsw%
1669 \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1670 \ifledRcol%
1671 \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
1672 \else%
1673 \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
1674 \fi%
1675 \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
1676 \fi%
1677 }%
1678 }%
1679 \advance\@tempcnta by -1%
1680 }%
1681 }%
1682 %

```

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

```

1683 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1684   \get@sw@txt{#2}%
1685   \unless\ifledRcol@%
1686   %

```

Just a precaution.

```

1687   \ifx\sw@list@inedtext\empty%
1688     \def\the@sw{999}%
1689     \def\this@absline{-99}%
1690   \else%
1691   %

```

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

```

1692     \gl@p\sw@list@inedtext\to\@tmp%
1693     \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1694     \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1695     \fi%
1696   %

```

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

```

1697     \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
1698       \numdef{\prev@line}{\this@absline-1}%
1699       \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1700     }%
1701     {\numdef{\sw@atthisline}{0}}%
1702   %

```

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

```

1703     \ifnumgreater{\sw@atthisline}{1}%
1704       {\showwordrank{#2}{\the@sw}}%
1705       {#2}%
1706   %

```

And the same for right side.

```

1707   \else%
1708     \ifx\sw@list@inedtext\empty%
1709       \def\the@sw{999}%
1710       \def\this@absline{-99}%
1711     \else%
1712       \gl@p\sw@list@inedtext\to\@tmp%
1713       \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1714       \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1715       \fi%
1716     \ifcsdef{sw@\sw@txt @\this@absline @\the\section@numR @R}{%

```

```

1717     \numdef{\prev@line}{\this@absline-1}%
1718     \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
1719     }%
1720     {\numdef{\sw@atthisline}{0}}%
1721     \ifnumgreater{\sw@atthisline}{1}%
1722         {\showwordrank{#2}{\the@sw}}%
1723         {#2}%
1724     \fi%
1725 }%
1726 %

```

`\showwordrank`₂₇ % Finally, the way the rank will be printed.

```

1728 \newcommand{\showwordrank}[2]{%
1729     #1\textsuperscript{#2}%
1730 }%
1731 %

```

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.

```

1732 \newbox\raw@text
1733 \newif\ifnumberedpar@
1734 \newcount\num@lines
1735 \newbox\one@line
1736 \newcount\par@line
1737 %

```

`\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`
`\numberpstarttrue` \pstart needs to appear at the start of every paragraph that is to be numbered; the
`\numberpstartfalse` \autopar command below may be used to insert these commands automatically.
`\labelpstarttrue` Beware: everything that occurs between `\pstart` and `\pend` is happening within
`\labelpstartfalse` a group; definitions must be global if you want them to survive past the end of the
`\thepstart` paragraph.

```

1738
1739 \newcommand{\AtEveryPstart}[1]{%
1740   \ifstrempy{#1}%
1741     {\xdef\at@every@pstart{}}%
1742     {\gdef\at@every@pstart{\noindent#1}}%
1743 }%
1744 \xdef\at@every@pstart{}%
1745
1746 \newcounter{pstart}
1747 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
1748 \newif\ifnumberpstart
1749 \numberpstartfalse
1750 \newif\iflabelpstart
1751 \labelpstartfalse
1752 \newcommandx*{\pstart}[1][1]{%
1753   \normal@pars%
1754   \ifstrempy{#1}{\at@every@pstart}{\noindent#1}%
1755   \ifluatex%
1756     \edef\l@luatextextdir@L{\the\textdir}%
1757   \fi%
1758   \@nbreaktrue%
1759   \ifnumbering \else%
1760     \led@err@PstartNotNumbered%
1761     \beginnumbering%
1762   \fi%
1763   \ifnumberedpar@%
1764     \led@err@PstartInPstart%
1765   \pend%
1766   \fi%
1767   \list@clear{\inserts@list}%
1768   \global\let\next@insert=\empty%
1769   \begingroup%
1770   \global\advance \l@dnumpstartsL\@ne
1771   \global\setbox\raw@text=\vbox\bgroup%
1772   \ifautopar\else%
1773     \ifnumberpstart%
1774       \ifinstanza\else%
1775         \ifsidepstartnum\else%
1776           \thepstart%
1777         \fi%
1778       \fi%
1779     \fi%

```

```

1780 \fi%
1781 \numberedpar@true%
1782 \iflabelpstart\protected@edef\@currentlabel%
1783     {\p@pstart\thepstart}
1784 \fi%
1785 \l@dzzeropenalties%
1786 \ignorespaces%because not automatically ignored if an optional argument
    is used (classical TeX behavior for space after commands)
1787 }
1788 %

```

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

```

1789 \newcommand*{\pend}[1][1]{\ifnumbering \else%
1790     \led@err@PendNotNumbered%
1791 \fi%
1792 \global\l@dskipversenumberfalse%
1793 \ifnumberedpar@ \else%
1794     \led@err@PendNoPstart%
1795 \fi%
1796 %

```

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.

```

1797 \l@dzzeropenalties%
1798 \endgraf\global\num@lines=\prevgraf\egroup%
1799 \global\par@line=0%
1800 %

```

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`.

```

1801 \csnumdef{pstartline}{0}%
1802 \loop\ifvbox\raw@text%
1803     \csnumdef{pstartline}{\pstartline+\@ne}%
1804     \do@line%
1805     \ifbypstart@%
1806         \ifnumequal{\pstartline}{1}{%
1807             \bgroup%
1808             \let\leavevmode\relax%
1809             \setline{1}%
1810             \egroup%
1811             \resetprevline@}{}%
1812 \fi%

```

```

1813 \repeat%
1814 %

```

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

```

1815 \flush@notes%
1816 \endgroup%
1817 \ignorespaces%
1818 %

```

Increase pstart counter.

```

1819 \ifnumberpstart%
1820   \pstartnumtrue%
1821 \fi%
1822 \addtocounter{pstart}{1}%
1823 %

```

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

```

1824 \normal@pars%
1825 \ifstrempy{#1}{\at@every@pend}{\noindent#1}%
1826 %

```

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

```

1827 \@nobreakfalse%
1828 \ifautopar%
1829   \autopar%
1830 \fi%
1831 }
1832 %
1833 %

```

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.

```

\AtEveryPend34
\at@every@pend35 \newcommand{\AtEveryPend}[1]{%
1836   \ifstrempy{#1}%
1837     {\xdef\at@every@pend{}}%
1838     {\gdef\at@every@pend{\noindent#1}}%
1839 }%
1840 \xdef\at@every@pend{}%
1841 %
1842 %

```

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

```

1843 \newcommand*{\l@dzzeropenalties}{%
1844   \brokenpenalty \z@ \clubpenalty \z@
1845   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
1846   \postdisplaypenalty \z@ \widowpenalty \z@}
1847
1848 %

```

`\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.

```

1849 \newif\ifautopar
1850 \autoparfalse
1851 \newcommand*{\autopar}{
1852   \ifledRcol
1853     \ifnumberingR \else
1854       \led@err@AutoparNotNumbered
1855       \beginnumberingR
1856       \fi
1857     \else
1858       \ifnumbering \else
1859         \led@err@AutoparNotNumbered
1860         \beginnumbering
1861         \fi
1862       \fi
1863       \autopartrue
1864       \everypar{\setbox0=\lastbox
1865         \endgraf \vskip-\parskip

```

```

1866 \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\
fi\fi
1867 \let\par=\pend}%
1868 \ignorespaces}
1869 %

```

\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.

```

1870 \newcommand*{\normal@pars}{\ifautopar\everypar{}\fi\let\par\endgraf}
1871
1872 %

```

\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.

```

1873 \newif\ifautopar@pause
1874 %

```

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

```

1875 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
1876 \newcommand*{\do@line}{%
1877   {\vbadness=10000
1878     \splittopskip=\z@
1879     \do@linehook
1880   \l@demptyd@ta
1881     \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
1882   \unvbox\one@line \global\setbox\one@line=\lastbox
1883   \getline@num
1884   \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{\}
1885   \ifnum\@lock>\@one
1886     \inserthangingsymboltrue
1887   \else
1888     \inserthangingsymbolfalse
1889   \fi
1890   \check@pb@in@verse
1891   \ifl@dhidenumber%
1892     \global\l@dhidenumberfalse%
1893     \f@x@l@cks%
1894   \else%
1895     \affixline@num%
1896   \fi%
1897   %

```

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

```

1898 \xifinlist{\the\l@dumpstartsL}{\eled@sections@@}%
1899     {\print@eledsection}%
1900     {\print@line}%
1901 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{\}
1902 }%
1903 %

```

VII.2.2 Process for “normal” line

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

```

1904 \def\print@line{
1905 %

```

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

```

1906     \affixpstart@num%
1907 %

```

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

```

1908     \hb@xt@ \linewidth{%
1909 %

```

User hook.

```

1910         \do@insidelinehook%
1911 %

```

Left line number

```

1912         \l@dld@ta%
1913 %

```

Prepare text to be inserted before notes.

```

1914         \if@firstlineofpage%
1915             \set@Xtxtbeforenotes%
1916             \global\@firstlineofpagefalse%
1917         \fi%
1918 %

```

Insert footnotes made of manuscripts data.

```

1919         \insert@msdata%
1920 %

```

Restore marginal and footnotes.

```

1921         \add@inserts\affixside@note%
1922 %

```

Print left notes.

```

1923 \l@dlsn@te
1924 %

```

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

```

1925 {\ledllfill\hb@xt@ \wd\one@line{\new@line%
1926 %

```

If we use Lua_{TEX} then restore the direction.

```

1927 \ifluatex%
1928 \textdir\l@luatextextdir@L%
1929 \fi%
1930 %

```

Insert, if needed, the hanging symbol.

```

1931 \inserthangingsymbol%
1932 %

```

And so, print the line.

```

1933 \l@dunhbox@line{\one@line}}%
1934 %

```

Right line number

```

1935 \ledrlfill\l@drd@ta%
1936 %

```

Print right notes.

```

1937 \l@drsn@te
1938 }}%
1939 %

```

And reinsert penalties (for page breaking)...

```

1940 \add@penalties%
1941 }
1942 %

```

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.

```

1943 \def\print@eledsection{%
1944 \if@firstlineofpage%
1945 \set@Xtxtbeforenotes%
1946 \global\@firstlineofpagefalse%
1947 \fi%
1948 \insert@msdata%

```

```

1949 \add@inserts\affixside@note%
1950 \numdef{\temp@}{\l@dnumpstartsL-1}%
1951 \xifinlist{\temp@}{\eled@sections@@}{\@nbreaktrue}{\@nbreakfalse}%
1952 \@eled@sectioningtrue%
1953 \csuse{eled@sectioning@\the\l@dnumpstartsL}%
1954 \@eled@sectioningfalse%
1955 \global\csundef{eled@sectioning@\the\l@dnumpstartsL}%
1956 \if@RTL%
1957 \hspace{-3\paperwidth}%
1958 {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1959 \else%
1960 \hspace{3\paperwidth}%
1961 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1962 \fi%
1963 \vskip-\baselineskip%
1964 }
1965 %

```

VII.2.4 Hooks

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

```

1966 \newcommand*\do@linehook{}
1967 \newcommand*\do@insidelinehook{}
1968 %

```

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

```

1969 \newcommand*\dolinehook[1]{\gdef\do@linehook{#1}}%
1970 \newcommand*\doinsidelinehook[1]{\gdef\do@insidelinehook{#1}}%
1971
1972 %

```

VII.2.5 Sidenotes and marginal line number initialization

`\l@demptyd@ta` Nulls the `\l@demptyd@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.

```

1973 \l@dcsnotetext \newcommand*\l@demptyd@ta{}%
1974 \l@dcsnotetext@l \gdef\l@dld@ta{}%
1975 \l@dcsnotetext@r \gdef\l@drd@ta{}%
1976 \gdef\l@dcsnotetext@l{}%
1977 \gdef\l@dcsnotetext@r{}%
1978 \gdef\l@dcsnotetext{}%
1979
1980 %

```

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

```

\l@drsn@te
1981 \newcommand{\l@dlsn@te}{%
1982   \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep}}
1983 \newcommand{\l@drsn@te}{%
1984   \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
1985
1986 %

```

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

```

1987 \newcommand*{\ledllfill}{\hfil}
1988 \newcommand*{\ledrlfill}{}
1989
1990 %

```

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.

```

1991 \newcommand*{\getline@num}{%
1992   \global\advance\absline@num \@ne%
1993   \do@actions
1994   \do@ballast
1995   \ifnumberline
1996     \ifsublines@
1997       \ifnum\sub@lock<\tw@
1998         \global\advance\subline@num \@ne
1999       \fi
2000     \else
2001       \ifnum\@lock<\tw@
2002         \global\advance\line@num \@ne
2003         \global\subline@num \z@
2004       \fi
2005     \fi
2006   \fi
2007 }
2008 %

```

`\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, $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ will be given extra encouragement to break the page here (see XI.2 p. 149).

`\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.

```
2009 \newcount\ballast@count
2010 \newcounter{ballast}
2011 \setcounter{ballast}{0}
2012 %
```

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.

```
2013 \newcommand*{\do@ballast}{\global\ballast@count \z@
2014 \begingroup
2015   \advance\absline@num \@ne
2016   \ifnum\next@actionline=\absline@num
2017     \ifnum\next@action>-1001\relax
2018       \global\advance\ballast@count by -\c@ballast
2019     \fi
2020   \fi
2021 \endgroup}
2022 %
```

`\do@actions` The `\do@actions` macro looks at the list of actions to take at particular absolute line
`\do@actions@next` numbers, and does everything that is specified for the current line.

It may call itself recursively, and to do this efficiently (using TeX's optimization for tail recursion), we define a control-sequence called `\do@actions@next` that is always the last thing that `\do@actions` does. If there could be more actions to process for this line, `\do@actions@next` is set equal to `\do@actions`; otherwise it is just `\relax`.

```
2023 \newcommand*{\do@actions}{%
2024   \global\let\do@actions@next=\relax
2025   \ifnum\absline@num<\next@actionline\else
2026   %
```

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.

```
2027   \ifnum\next@action>-1001
2028     \global\page@num=\next@action
2029     \global\@firstlineofpagetrue%
2030     \ifbypage@
2031       \global\line@num=\z@ \global\subline@num=\z@
2032       \resetprevline@
2033     \fi
2034     \add@msdata@firstlineofpage%
2035   %
```

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`.)

```

2036 \else
2037 \ifnum\next@action<-4999
2038 \@l@tempcnta=-\next@action
2039 \advance\@l@tempcnta by -5001
2040 \ifsublines@
2041 \global\subline@num=\@l@tempcnta
2042 \else
2043 \global\line@num=\@l@tempcnta
2044 \fi
2045 %

```

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

```

2046 \else
2047 \@l@tempcnta=-\next@action
2048 \advance\@l@tempcnta by -1000
2049 \do@actions@fixedcode
2050 \fi
2051 \fi
2052 %

```

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.

```

2053 \ifx\actionlines@list\empty
2054 \gdef\next@actionline{1000000}%
2055 \else
2056 \gl@p\actionlines@list\to\next@actionline
2057 \gl@p\actions@list\to\next@action
2058 \global\let\do@actions@next=\do@actions
2059 \fi
2060 \fi
2061 %

```

Make the recursive call, if necessary.

```

2062 \do@actions@next}
2063
2064 %

```

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

```

2065 \newcommand*{\do@actions@fixedcode}{%
2066 \ifcase\@l@tempcnta
2067 \or% % 1001 = starting sublineation
2068 \global\sublines@true
2069 \or% % 1002 = ending sublineation
2070 \global\sublines@false
2071 \or% % 1003 = starting locking number
2072 \global\@lock=\@ne

```

```

2073 \or% % 1004 = ending locking number
2074 \ifnum\@lock=\tw@
2075 \global\@lock=\thr@@
2076 \else
2077 \global\@lock=\z@
2078 \fi
2079 \or% % 1005 = starting locking subnumber
2080 \global\sub@lock=\@ne
2081 \or% % 1006 = ending locking subnumber
2082 \ifnum\sub@lock=\tw@
2083 \global\sub@lock=\thr@@
2084 \else
2085 \global\sub@lock=\z@
2086 \fi
2087 \or% % 1007 = skipping numbering
2088 \l@dskipnumbertrue
2089 \or% % 1008 = skipping numbering in stanza
2090 \l@dskipversenumbertrue%
2091 \or% % 1009 = hiding number
2092 \l@dhiddenumbertrue
2093 \or% % 1010 = inserting msdata
2094 \add@msdata%
2095 \else
2096 \led@warn@BadAction
2097 \fi}
2098
2099
2100 %

```

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:

$$\begin{aligned}
 n &= \text{int}((\text{linenum} - \text{firstlinenum}) / \text{linenumincrement}) \\
 m &= \text{firstlinenum} + (n \times \text{linenumincrement})
 \end{aligned}$$

(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@dttempcnta`, 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@dttempcntb` for comparison.

First, the case when we are within a sub-line range.

```

2101 \newcommand*{\affixline@num}{%
2102 %

```

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).

```

2103 \ifledgroupnotesL\else
2104 \ifnumberline
2105 \ifl@dskipnumber
2106 \global\l@dskipnumberfalse
2107 \else
2108 \ifsublines@
2109 \@l@dttempcntb=\subline@num
2110 \ifnum\subline@num>\c@firstsublinenum
2111 \@l@dttempcnta=\subline@num
2112 \advance\@l@dttempcnta by-\c@firstsublinenum
2113 \divide\@l@dttempcnta by\c@sublinenumincrement
2114 \multiply\@l@dttempcnta by\c@sublinenumincrement
2115 \advance\@l@dttempcnta by\c@firstsublinenum
2116 \else
2117 \@l@dttempcnta=\c@firstsublinenum
2118 \fi
2119 %

```

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.

```

2120 \ch@cksub@l@ck
2121 %

```

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

```

2122 \else
2123 \@l@dttempcntb=\line@num
2124 %

```

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

```

2125 \ifx\linenumberlist\empty
2126 \ifnum\line@num>\c@firstlinenum
2127 \@l@dttempcnta=\line@num
2128 \advance\@l@dttempcnta by-\c@firstlinenum
2129 \divide\@l@dttempcnta by\c@linenumincrement
2130 \multiply\@l@dttempcnta by\c@linenumincrement
2131 \advance\@l@dttempcnta by\c@firstlinenum
2132 \else
2133 \@l@dttempcnta=\c@firstlinenum
2134 \fi
2135 \else
2136 %

```

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

```

2137      \l@ldtempcnta=\line@num
2138      \edef\rem@inder{,\linenumberlist,\number\line@num,}%
2139      \edef\sc@n@list{\def\noexpand\sc@n@list
2140        #####1,\number\l@ldtempcnta,#####2|{\def\noexpand\rem@inder
2141        {#####2}}}%
2142      \sc@n@list\expandafter\sc@n@list\rem@inder|}%
2143      \ifx\rem@inder\empty%
2144        \advance\l@ldtempcnta\@ne
2145      \fi
2146    \fi
%
```

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

```

2147      \ch@ck@l@ck
2148    \fi
2149  %
```

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

```

2150      \ifnum\l@ldtempcnta=\l@ldtempcntb
2151      \ifl@dskipversenumber\else
2152    %
```

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 $\mathcal{E}\mathcal{T}\mathcal{E}\mathcal{X}$ 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@ld@ta` A left line number is stored in `\l@ld@ta` and a right one in `\l@drd@ta`.

```

\l@drd@ta
2153      \if@twocolumn
2154        \if@firstcolumn
2155          \gdef\l@ld@ta{\llap{\leftlinenum}}}%
2156        \else
2157          \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2158        \fi
2159      \else
2160        \l@ldtempcntb=\line@margin
2161        \ifnum\l@ldtempcntb>\@ne
2162          \advance\l@ldtempcntb \page@num
2163        \fi
2164        \ifodd\l@ldtempcntb
```

```

2165         \gdef\l@drd@ta{\rlap{{\rightlinenum}}}%
2166         \else
2167         \gdef\l@dld@ta{\llap{{\leftlinenum}}}%
2168     \fi
2169 \fi
2170 \fi
2171 \fi
2172 %

```

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

```

2173     \f@x@l@cks
2174 \fi
2175 \fi
2176 \fi
2177 }
2178 %
2179 %

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck` checks subline locking. If it fails, then we disable the line-number display by setting the counters to arbitrary but unequal values.

```

2180 \newcommand*{\ch@cksub@l@ck}{%
2181     \ifcase\sub@lock
2182     \or
2183     \ifnum\sublock@disp=\@ne
2184     \l@tempcntb=\z@ \l@tempcnta=\@ne
2185     \fi
2186     \or
2187     \ifnum\sublock@disp=\tw@ \else
2188     \l@tempcntb=\z@ \l@tempcnta=\@ne
2189     \fi
2190     \or
2191     \ifnum\sublock@disp=\z@
2192     \l@tempcntb=\z@ \l@tempcnta=\@ne
2193     \fi
2194 \fi}
2195 %

```

Similarly for line numbers.

```

2196 \newcommand*{\ch@ck@l@ck}{%
2197     \ifcase\l@lock
2198     \or
2199     \ifnum\lock@disp=\@ne
2200     \l@tempcntb=\z@ \l@tempcnta=\@ne
2201     \fi
2202     \or
2203     \ifnum\lock@disp=\tw@ \else

```

```

2204         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2205         \fi
2206         \or
2207         \ifnum\lock@disp=\z@
2208             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2209         \fi
2210     \fi}
2211 %

```

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

```

2212 \newcommand*{\f@x@l@cks}{%
2213     \ifcase\@lock
2214     \or
2215         \global\@lock=\tw@
2216     \or \or
2217         \global\@lock=\z@
2218     \fi
2219     \ifcase\sub@lock
2220     \or
2221         \global\sub@lock=\tw@
2222     \or \or
2223         \global\sub@lock=\z@
2224     \fi}
2225 %
2226 %

```

X Pstart number printing in side

In side, the printing of pstart number is running like the printing of line number. There is only some differences:

`\affixpstart@num`
`\pstartnum`

- The pstarts counter is upgrade in the `\pend` command. Consequently, the `\affixpstart@num` command has not to upgrade it, unlike the `\affixline@num` which upgrades the lines counter.
- To print the pstart number only at the beginning of a pstart, and not in every line, a boolean test is made. The `\pstartnum` boolean is set to TRUE at every `\pend`. It is tried in the `\leftpstartnum` and `\rightpstartnum` commands. After the try, it is set to FALSE.

```

\leftpstartnum27
\rightpstartnum28 \newif\ifsidepstartnum
\ifsidepstartnum29 \newcommand*{\affixpstart@num}{%
2230     \ifsidepstartnum
2231         \if@twocolumn
2232             \if@firstcolumn
2233                 \gdef\l@dld@ta{\llap{{\leftpstartnum}}}%

```

```

2234         \else
2235             \gdef\l@drd@ta{\rlap{{\rightpstartnum}}}%
2236         \fi
2237     \else
2238         \@l@dttempcntb=\line@margin
2239         \ifnum\@l@dttempcntb>\@ne
2240             \advance\@l@dttempcntb \page@num
2241         \fi
2242         \ifodd\@l@dttempcntb
2243             \gdef\l@drd@ta{\rlap{{\rightpstartnum}}}%
2244         \else
2245             \gdef\l@dld@ta{\llap{{\leftpstartnum}}}%
2246         \fi
2247     \fi
2248 \fi
2249
2250 }
2251 %
2252
2253 \newif\ifpstartnum
2254 \pstartnumtrue
2255 \newcommand*{\leftpstartnum}{
2256     \ifpstartnum\thepstart
2257     \kern\linenumsep\fi
2258     \global\pstartnumfalse
2259 }
2260 \newcommand*{\rightpstartnum}{
2261     \ifpstartnum
2262     \kern\linenumsep
2263     \thepstart
2264     \fi
2265     \global\pstartnumfalse
2266 }
2267 %

```

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.

```
2268 \list@create{\inserts@list}
2269 %
```

`\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`.

```
2270 \newcommand*{\add@inserts}{%
2271   \global\let\add@inserts@next=\relax
2272 %
```

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

```
2273   \ifx\inserts@list\empty \else
2274 %
```

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.

```
2275   \ifx\next@insert\empty
2276     \ifx\insertlines@list\empty
2277       \global\noteschanged@true
2278       \gdef\next@insert{100000}%
2279     \else
2280       \gl@p\insertlines@list\to\next@insert
2281     \fi
2282   \fi
2283 %
```

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.

```
2284   \ifnum\next@insert=\absline@num
2285     \gl@p\inserts@list\to\@insert
2286     \@insert
2287     \global\let\@insert=\undefined
2288     \global\let\next@insert=\empty
2289     \global\let\add@inserts@next=\add@inserts
2290   \fi
2291 \fi
2292 %
```

Make the recursive call, if necessary.

```
2293 \add@inserts@next}
2294
2295 %
```

XI.2 Penalties

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

In this code, `\num@lines` is the number of lines in the whole paragraph, and `\par@line` is the line we are working on at the moment. The count `\@l@tempcnta` is used to calculate and accumulate the penalty; it is initially set to the value of `\ballast@count`, which has been worked out in `\do@ballast` above (VIII p. 139). Finally, the penalty is checked to see that it does not go below -10000 .

```
2296 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
2297   \ifnum\num@lines>\@ne
2298     \global\advance\par@line \@ne
2299     \ifnum\par@line=\@ne
2300       \advance\@l@tempcnta \clubpenalty
2301     \fi
2302     \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
2303     \ifnum\@l@tempcntb=\num@lines
2304       \advance\@l@tempcnta \widowpenalty
2305     \fi
2306     \ifnum\par@line<\num@lines
2307       \advance\@l@tempcnta \interlinepenalty
2308     \fi
2309   \fi
2310   \ifnum\@l@tempcnta=\z@
2311     \relax
2312   \else
2313     \ifnum\@l@tempcnta>-10000
2314       \penalty\@l@tempcnta
2315     \else
2316       \penalty -10000
2317     \fi
2318   \fi}
2319
2320 %
```

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.

```

2321 \newcommand*{\flush@notes}{%
2322   \@xloop
2323   \ifx\inserts@list\empty \else
2324     \gl@p\inserts@list\to\@insert
2325     \@insert
2326     \global\let\@insert=\undefined
2327   \repeat}
2328
2329 %

```

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

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

```

2330 \def\@xloop#1\repeat{%
2331   \def\body{#1\expandafter\body\fi}%
2332   \body}
2333
2334 %

```

XI.4 Text before notes

\set@Xtxtbeforenotes The $\text{\set@Xtxtbeforenotes}$ macro resets the $\text{Xtxtbeforenotes@}\langle series \rangle\text{@typeset}$ boolean to false. Just before the first note of the $\langle series \rangle$ in a page, the Xtextbeforenotes will be inserted.

```

2335 \newcommand{\set@Xtxtbeforenotes}{%
2336   \unless\ifnocritical@%
2337     \def\do##1{%
2338       \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2339     }%
2340     \dolistloop{\@series}%
2341   \fi%
2342 }%
2343 %

```

`\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.

```

2344 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2345   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2346     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2347     \ifcsvoid{Xtxtbeforenotes@#1}{}%
2348     \ifcsstring{series@display#1}{paragraph}%
2349     {\noindent\csuse{Xtxtbeforenotes@#1}}%
2350     {\expandafter\insert\csname#1footins\endcsname%
2351       \bgroup%
2352         \noindent\strut\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@
#1}%
2353       \egroup%
2354     }%
2355   }%
2356 }%
2357 {}%
2358 }%
2359 %

```

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.

```

2360 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@lemmafont#7|}
2361 \def\select@lemmafont#1/#2/#3/#4|%
2362   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2363   \selectfont}
2364
2365 %

```

XII.2 Individual note options

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

```

2366 \newcommand*{\footnoteoptions@}[3]{%
2367   \def\do##1{%
2368     \ifstrequal{#1}{L}{% In Leftside
2369       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@list%
2370       \global\advance\insert@count \@ne% Increment the left insert
counter.
2371     }%
2372     {%
2373       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@listR%
2374       \global\advance\insert@countR \@ne% Increment the right insert
counter insert.
2375     }%
2376   }%
2377   \notblank{#2}{\docsvlist{#2}}{}}% Parsing all options
2378 }
2379 %

```

XII.3 Notes language

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

```

2380 \newcommandx*{\footnotelang@lua}[1][1=L,usedefault]{%
2381   \ifstrequal{#1}{L}{%
2382     \xright@appenditem{{\csxdef{footnote@luatextextdir}{\the\textdir}}}{\to\
inserts@list%Know the dir of lemma
2383     \global\advance\insert@count \@ne%
2384     \xright@appenditem{{\csxdef{footnote@luatexpardir}{\the\pardir}}}{\to\
inserts@list%Know the dir of lemma
2385     \global\advance\insert@count \@ne%
2386   }%
2387   {%
2388     \xright@appenditem{{\csxdef{footnote@luatextextdir}{\the\textdir}}}{\to\
inserts@listR%Know the dir of lemma
2389     \global\advance\insert@countR \@ne%
2390     \xright@appenditem{{\csxdef{footnote@luatexpardir}{\the\pardir}}}{\to\
inserts@listR%Know the dir of lemma
2391     \global\advance\insert@countR \@ne%
2392   }%
2393 }
2394 %

```

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

```

2395 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2396   \ifstrequal{#1}{L}{%
2397     \if@RTL%
2398       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@list%Know the language used in the lemma
2399       \global\advance\insert@count \@ne%
2400     }%
2401   \else
2402     \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@list%Know the language of lemma
2403     \global\advance\insert@count \@ne%
2404   }%
2405   \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language\language}}{\to\
inserts@list%Know the language of lemma
2406   \global\advance\insert@count \@ne%
2407 }%
2408 \if@RTL
2409   \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@listR%Know the language of lemma
2410   \global\advance\insert@countR \@ne%
2411 }%
2412 \else
2413   \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@listR%Know the language of lemma
2414   \global\advance\insert@countR \@ne%
2415 }%
2416   \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language\language}}{\to\
inserts@listR%Know the language of lemma
2417   \global\advance\insert@countR \@ne%
2418 }%
2419 }%

```

XII.4 General survey of the way we manage notes

The processing of each note is done by four principal macros: the `\vfootnote` macro takes the text of the footnote and does the `\insert`; it calls on the `\footfmt` macro to select the right fonts, print the line number and lemma, and do any other formatting needed for that individual note. Within the output routine, the two other macros, `\footstart` and `\footgroup`, are called; the first prints extra vertical space and a footnote rule, if desired; the second does any reformatting of the whole set of the footnotes in this series for this page—such as paragraphing or division into columns—and then sends them to the page.

These four macros, and the other macros and parameters shown here, are distinguished by the ‘series letter’ that indicates which set of the footnotes we are dealing with—A, B, C, D, or E. The series letter always precedes the string `foot` in macro and

parameter names. Hence, for the A series, the four macros are called `\vAfootnote`, `\Afootfmt`, `\Afootstart`, and `\Afootgroup`.

These macros are changed depending of the footnotes arrangement: “normal”, “paragraphed”, “two columns” or “three columns”.

XII.5 General setup

`\footsplitskips` Some setup code that is common for a variety of the footnotes. The setup is for:

- `\interlinepenalty`.
- `\splittopskip` (skip before last part of notes that flow from one page to another).
- `\splitmaxdepth`.
- `\floatingpenalty`, that is penalty values being added when a long note flows from one page to another. Here, we let it to 0 when we are processing parallel pages in `eledpar`, in order to allow notes to flow from left to right pages and *vice-versa*. Otherwise, we let it to `\@MM`, which is the standard \TeX `\floatingpenalty`.

```
2420 \newcommand*{\footsplitskips}{%
2421   \interlinepenalty=\interfootnotelinepenalty
2422   \unless\ifl@dprintingpages%
2423     \floatingpenalty=\@MM%
2424   \fi%
2425   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2426   \leftskip=\z@skip \rightskip=\z@skip}
2427
2428 %
```

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

```
2429 \let\normalfootnoterule=\footnoterule
2430 %
```

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.

```
2431 \newcommandx{\Xarrangement}[2][1,usedefault]{%
2432   \def\do##1{%
2433     \csname Xarrangement@#2\endcsname{##1}%
2434   }%
```

```

2435 \ifstrempy{#1}%
2436   {%
2437   \dolistloop{\@series}%
2438   }%
2439   {
2440   \docsvlist{#1}%
2441   }%
2442 }%
2443 %

```

XII.6.2 Normal footnote

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

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

```

\skip\Afootins=12pt plus5pt minus5pt
\count\Afootins=1000
\dimen\Afootins=0.8\vsiz
\let\Afootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule

```

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

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

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

```

2444 \newcommand*{\Xarrangement@normal}[1]{%
2445   \csgdef{series@display#1}{normal}
2446   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
2447   \expandafter\let\csname #1footnote\endcsname=\normalvfootnote
2448   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
2449   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
2450   \expandafter\let\csname #1footnoterule\endcsname=%
2451   \normalfootnoterule
2452   \count\csname #1footins\endcsname=1000
2453   \dimen\csname #1footins\endcsname=\cuse{Xmaxhnotes@#1}
2454   \skip\csname #1footins\endcsname=\cuse{Xbeforenotes@#1}%
2455   \advance\skip\csname #1footins\endcsname by\cuse{Xafterrule@#1}%
2456   %

```

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 $\text{T}_{\text{E}}\text{X}$ ’s inserts.

```

2457 \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2458 %

```

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).

```

2459 \ifnoledgroup@else%
2460 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2461 \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2462 \count\csname mp#1footins\endcsname=1000
2463 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2464 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2465 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2466 \fi
2467 }
2468
2469 %

```

`\normalvfootnote` We now begin a series of commands that do ‘normal’ footnote formatting: a format much like that implemented in PLAIN T_EX, in which each footnote is a separate paragraph.

`\normalvfootnote` takes the series letter as #1, and the entire text of the footnote is #2. It does the `\insert` for this note, calling on the `\footfmt` macro for this note series to format the text of the note.

```

2470 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2471 \insert@Xtxtbeforenotes{#1}%
2472 \csuse{Xbeforeinserting@#1}%
2473 \insert\csname #1footins\endcsname\bgroup
2474 \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2475 \noindent\csuse{Xhooknote@#1}%
2476 \csuse{Xnotefontsize@#1}%
2477 \footsplitskips
2478 \ifl@dpairing\ifl@dpaging\else%
2479 \setXnoteswidthliketwocolumns@{#1}%
2480 \fi\fi%
2481 \setXnotespositionliketwocolumns@{#1}%
2482 \spaceskip=\z@skip \xspaceskip=\z@skip
2483 \csname #1footfmt\endcsname #2{#1}\egroup}
2484 %

```

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

```

2485 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
2486 \global\setbox\@nameuse{mp#1footins}\vbox{%
2487 \insert@Xtxtbeforenotes{#1}%
2488 \unvbox\@nameuse{mp#1footins}
2489 \noindent\csuse{Xhooknote@#1}%
2490 \csuse{Xnotefontsize@#1}%
2491 \hsize\columnwidth

```

```

2492 \parboxrestore
2493 \color@begingroup
2494 \csname #1footfmt\endcsname #2{#1}\color@endgroup}}
2495
2496 %

```

\normalfootfmt \normalfootfmt is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 92), 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.

```

2497
2498
2499 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmt}[4]{%
2500 \Xledsetnormalparstuff{#4}%
2501 \hangindent=\csuse{Xhangindent@#4}%
2502 \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2503 \rule{z@}{splittopskip}%
2504 {\printlinefootnote{#1}{#4}}%
2505 \print@lemma{#1}{#2}{#4}%
2506 \csuse{Xwrapcontent@#4}{#3}%
2507 \strut\par}
2508 %

```

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

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

But if the skip \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.

```

2509 \newcommand*{\normalfootstart}[1]{%
2510 %

```

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

```

2511 \ifdimequal{0pt}{\Xprenotes@}{}%
2512   {%
2513     \iftoggle{Xprenotes@}{%
2514       \togglefalse{Xprenotes@}%
2515       \skip\csname #1footins\endcsname=%
2516       \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
2517     }%
2518   }%
2519   \vskip\skip\csname #1footins\endcsname%
2520   %
2521 %

```

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. 203 for the general description of the problem.

```

2522 \leftskip0pt \rightskip0pt
2523 \ifl@dpairing\else%
2524   \hsize=\old@hsize%
2525   \fi%
2526   \setXnoteswidthliketwocolumns@{#1}%
2527   \setXnotespositionliketwocolumns@{#1}%
2528 %

```

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

```

2529 \print@Xfootnoterule{#1}%
2530 }%
2531 %

```

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

```

2532 \newcommand*{\normalfootgroup}[1]{%
2533   \csuse{Xbhookgroup@#1}%
2534   \unvbox\csname #1footins\endcsname%
2535   \hsize=\old@hsize%
2536   }%
2537 %
2538 %

```

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

```

2539 \unless\ifnoledgroup@
2540 \newcommand*{\mpnormalfootgroup}[1]{%
2541   \vskip\skip\@nameuse{mp#1footins}
2542   \ifl@dpairing\ifparledgroup%
2543     \leavevmode\marks\parledgroup@{begin}%

```

```

2544 \marks\parledgroup@series{#1}%
2545 \marks\parledgroup@type{Xfootnote}%
2546 \fi\fi\normalcolor%
2547 \ifparledgroup%
2548 \ifl@dpairing%
2549 \else%
2550 \setXnoteswidthliketwocolumns@{#1}%
2551 \setXnotespositionliketwocolumns@{#1}%
2552 \print@Xfootnoterule{#1}%%
2553 \fi%
2554 \else%
2555 \setXnoteswidthliketwocolumns@{#1}%
2556 \setXnotespositionliketwocolumns@{#1}%
2557 \print@Xfootnoterule{#1}%%
2558 \fi%
2559 \setlength{\parindent}{0pt}
2560 \csuse{Xbhookgroup@#1}%
2561 \unvbox\csname mp#1footins\endcsname}}
2562 \fi
2563 %

```

XII.6.3 Paraphed footnotes

The paraphed-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 paraphed; 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 paraphed 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 paraphed.

```

2564 \newcommand*{\Xarrangement@paragraph}[1]{%
2565 \csgdef{series@display#1}{paragraph}
2566 \expandafter\let\csname #1footstart\endcsname=\parafootstart
2567 \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
2568 \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
2569 \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
2570 \count\csname #1footins\endcsname=1000
2571 \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2572 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2573 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2574 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%

```

```
2575 \para@footsetup{#1}
2576 %
```

And the extra setup for minipages.

```
2577 \ifnoledgroup@else
2578   \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
2579   \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
2580   \count\csname mp#1footins\endcsname=1000
2581   \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2582   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2583   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2584 \fi
2585 }
2586 %
```

\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.

```
2587 \providecommand{\footfudgefiddle}{64}
2588 %
```

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

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

```
2589 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}
2590   \setXnoteswidthliketwocolumns@{#1}%
2591   \ifcsemtty{Xwidth@#1}%
2592     {}%
2593     {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
2594   \dimen0=\baselineskip
2595   \multiply\dimen0 by 1024
2596   \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
2597   \csxdef{#1footfudgefactor}{%
2598     \expandafter\strip@pt\dimen0 }}
2599
2600 %
```

`\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.

```

2601 \newcommand*{\parafootstart}[1]{%
2602   \rightskip=0pt \leftskip=0pt%
2603   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2604   \ifdimequal{0pt}{\Xprenotes@}{}%
2605   {%
2606     \iftoggle{Xprenotes@}{%
2607       \togglefalse{Xprenotes@}%
2608       \skip\csname #1footins\endcsname=%
2609       \glueexpr\csuse{Xprenotes@}+\csuse{Xaftererrule@#1}\relax%
2610     }%
2611   }%
2612 }%
2613 \vskip\skip\csname #1footins\endcsname%
2614 \setXnoteswidthliketwocolumns@{#1}%
2615 \setXnotespositionliketwocolumns@{#1}%
2616 \print@Xfootnoterule{#1}%
2617 \let\bidirTL@everypar\@empty%
2618 \noindent\leavevmode}
2619 %

```

`\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.²⁸

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.²⁹ 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.

²⁸Michael Downes, ‘Line Breaking in `\unhboxed` Text’, *TUGboat* **11** (1990), pp. 605–612.

²⁹See *The TeXbook*, p. 455 (editions after January 1990).

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.³⁰ Michael's unboxing macro is called `\Xunvxh`: unvbox, extract the last line, and unhbox it.

Doing things this way has an important consequence: as Michael pointed out, you really can't put an explicit line-break into a note built in a `\vbox` the way we are doing.³¹ 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. 157 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.

```

2620 \newcommand*{\paravfootnote}[2]{%
2621   \csuse{Xbeforeinserting@#1}%
2622   \insert\csname #1footins\endcsname
2623   \bgroup
2624     \csuse{Xnotefontsize@#1}
2625     \footplitskips
2626     \setbox0=\vbox{\hsize=\maxdimen%
2627       \let\bidir@RTL@everypar\@empty%
2628       \insert@Xtxtbeforenotes{#1}%
2629       \noindent\csuse{Xhooknote@#1}%
2630       \csname #1footfmt\endcsname #2{#1}}%
2631     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2632     \dp0=0pt
2633     \ht0=\csname #1footfudgefactor\endcsname\wd0
2634   %

```

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

```

2635   \if@RTL\noindent \leavevmode\fi\box0%
2636   \penalty0

```

³⁰Wayne supplied his own macros to do this, but since they were almost identical to Michael's, Peter Wilson have used the latter's `\Xunvxh` macro since it is publicly documented.

³¹'Line Breaking', p. 610.

```

2637 \egroup}
2638
2639 %

```

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.

```

2640 \newcommand*{\mpparavfootnote}[2]{%
2641   \global\setbox\@nameuse{mp#1footins}\vbox{%
2642     \unvbox\@nameuse{mp#1footins}%
2643     \csuse{Xnotefontsize@#1}
2644     \footsplitskips
2645     \setbox0=\vbox{\hsize=\maxdimen%
2646       \let\bidir@RTL@everypar\@empty%
2647       \insert@Xtxtbeforenotes{#1}%
2648       \noindent\color@begingroup%
2649       \csuse{Xbhooknote@#1}%
2650       \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
2651     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2652     \dp0=\z@
2653     \ht0=\csname #1footfudgefactor\endcsname\wd0
2654     \box0
2655     \penalty0
2656   }}
2657
2658 %

```

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

```

2659 \newcommand*{\Xunvxh}[2]{%
2660   \setbox0=\vbox{\unvbox#1%
2661     \global\setbox1=\lastbox}%
2662   \unhbox1
2663   \unskip           % remove \rightskip,
2664   \unskip           % remove \parfillskip,

```

```

2665 \unpenalty          % remove \penalty of 10000,
2666 \hskip\csuse{Xafternote@#2}\relax}% but add the glue to go between the
notes
2667
2668 %

```

\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).

```

2669 \newcommand*{\parafootfmt}[4]{%
2670   \Xinsertparafootsep{#4}%
2671   \ledsetnormalparstuff@common%
2672   \printlinefootnote{#1}{#4}%
2673   \print@lemma{#1}{#2}{#4}%
2674   \csuse{Xwrapcontent@#4}{#3}%
2675   \penalty-10 }
2676 %

```

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.

```

2677 \newcommand*{\parafootgroup}[1]{%
2678   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2679   \unvbox\csname #1footins\endcsname
2680   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2681   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2682   \makehboxofhboxes
2683   \setbox0=\hbox{\unhbox0 \removehboxes}%
2684   \csuse{Xhookgroup@#1}%
2685   \csuse{Xnotefontsize@#1}%
2686   \unhbox0\par%
2687   \global\hsize=\old@hsize%
2688   }%
2689
2690 %

```

\mpparafootgroup The minipage version.

```

2691 \newcommand*{\mpparafootgroup}[1]{%
2692   \setXnoteswidthliketwocolumns@{#1}%
2693   \vskip\skip\@nameuse{mp#1footins}
2694   \ifl@dpairing\ifparledgroup%
2695     \leavevmode\marks\parledgroup@{begin}%
2696     \marks\parledgroup@series{#1}%
2697     \marks\parledgroup@type{Xfootnote}%
2698   \fi\fi\normalcolor
2699   \ifparledgroup%
2700     \ifl@dpairing%
2701     \else%
2702       \setXnoteswidthliketwocolumns@{#1}%
2703       \setXnotespositionliketwocolumns@{#1}%
2704       \print@Xfootnoterule{#1}%%
2705     \fi%
2706   \else%
2707     \setXnoteswidthliketwocolumns@{#1}%
2708     \setXnotespositionliketwocolumns@{#1}%
2709     \print@Xfootnoterule{#1}%
2710   \fi%
2711   \unvbox\csname mp#1footins\endcsname
2712   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2713   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2714   \makehboxofhboxes
2715   \setbox0=\hbox{\unhbox0 \removehboxes}%
2716   \csuse{Xhookgroup@#1}%
2717   \csuse{Xnotefontsize@#1}%
2718   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2719   \unhbox0\par}}
2720
2721 %

```

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

```

\makehboxofhboxes 22 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes 23   \loop
2724     \unpenalty
2725     \setbox2=\lastbox
2726     \ifhbox2
2727       \setbox0=\hbox{\box2\unhbox0}%
2728     \repeat}
2729
2730 \newcommand*{\removehboxes}{\setbox0=\lastbox
2731   \ifhbox0{\removehboxes}\unhbox0 \fi}
2732
2733 %

```

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

```

\prevpage@num34 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep35 \ifledRcol@%
2736 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
2737 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2738 {\ifnumequal{\csuse{prevline#1}}{\line@numR}%
2739 {\ifcsempy{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
2740 {\csuse{Xparafootsep@#1}}%
2741 }%
2742 {\csuse{Xparafootsep@#1}}%
2743 }%
2744 {}%
2745 \global\csname #1prevpage@numR\endcsname=\page@numR%
2746 \else%
2747 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
2748 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2749 {\ifnumequal{\csuse{prevline#1}}{\line@num}%
2750 {\ifcsempy{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
2751 {\csuse{Xparafootsep@#1}}%
2752 }%
2753 {\csuse{Xparafootsep@#1}}%
2754 }%
2755 {}%
2756 \global\csname #1prevpage@num\endcsname=\page@num%
2757 \fi%
2758 }
2759 %

```

XII.6.4 Columnar footnotes

Common tools

`\rigidbalance` We will now define macros for three-column notes and two-column notes. Both sets of macros will use `\rigidbalance`, which splits a box (#1) into into a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\Xrigidbalance` macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they do not depend on white space. Note also the extra unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it goes into the alignment.

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

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

```

2760 \newcount\@k \newdimen\@h
2761 \newcommand*\Xrigidbalance[3]{%
2762 \hsize=\expandafter\dimexpr\csuse{Xwidth@}\@currentseries\relax%

```

```

2763 \rigidbalance{#1}{#2}{#3}%
2764 }%
2765
2766 \newcommand*{\rigidbalanceX}[3]{%
2767 \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
2768 \rigidbalance{#1}{#2}{#3}%
2769 }%
2770
2771 \newcommand*{\rigidbalance}[3]{%
2772 \setbox0=\box#1 \@k=#2 \@h=#3%
2773 @@line{\splittopskip=\@h \vbadness=\@M \hfilneg
2774 \valign{##\vfil\cr\dosplits}}}%
2775
2776 \newcommand*{\dosplits}{\ifnum\@k>0 \noalign{\hfil}\splitoff
2777 \global\advance\@k-1\cr\dosplits\fi}
2778
2779 \newcommand*{\splitoff}{\dimen0=\ht0
2780 \divide\dimen0 by\@k \advance\dimen0 by\@h
2781 \setbox2 \vsplit0 to \dimen0
2782 \unvbox2 }
2783
2784 %

```

Three columns

```

\Xarrangement@threecol 2785 \newcommand*{\Xarrangement@threecol}[1]{%
2786 \csgdef{series@display#1}{threecol}
2787 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
2788 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
2789 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
2790 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2791 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2792 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2793 \threecolfootsetup{#1}
2794 %

```

The additional setup for minipages.

```

2795 \ifnoledgroup@else
2796 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2797 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
2798 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2799 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2800 \mpthreecolfootsetup{#1}
2801 \fi
2802 }
2803
2804 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XII.6.2 p. 157 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.

```
2805 \newcommand*{\threecolfootsetup}[1]{%
2806   \count\csname #1footins\endcsname 333
2807   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
2808   \multiply\dimen\csname #1footins\endcsname \thr@@}
2809   %
```

`\mpthreecolfootsetup` The setup for minipages.

```
2810 \newcommand*{\mpthreecolfootsetup}[1]{%
2811   \count\csname mp#1footins\endcsname 333
2812   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
2813   %
2814   %
```

`\threecolvfootnote` `\threecolvfootnote` is the `\vfootnote` command for three-column notes. The call to `\Xnotefontsize@{s}` ensures that the `\splittopskip` and `\splitmaxdepth` take their values from the right `\strutbox`: the one used in a footnotes. Note especially the importance of temporarily reducing the `\hsize` to 0.3 of its normal value. This determines the widths of the individual columns. So if the normal `\hsize` is, say, 10 cm, then each column will be $0.3 \times 10 = 3$ cm wide, leaving a gap of 1 cm spread equally between columns (i.e., .5 cm between each).

The arguments are `#1` the note series letter and `#1` the full text of the note (including numbers, lemma and text).

```
2815 \notbool{parapparatus@}{\newcommand*}{\newcommand*}{\threecolvfootnote}[2]{%
2816   \csuse{Xbeforeinserting@#1}%
2817   \insert\csname #1footins\endcsname\bgroup%
2818   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2819   \noindent\csuse{Xbhooknote@#1}%
2820   \csuse{Xnotefontsize@#1}%
2821   \footplitskips%
```

```

2822 \csname #1footfmt\endcsname #2{#1}\egroup}
2823 %

```

\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.

```

2824 \notbool{parapparatus@}\newcommand*{\threecolfootfmt}[4]{%
2825 \normal@pars%
2826 \hsize \csuse{Xhsizethreecol@#4}%
2827 \nottoggle{Xparindent@#4}\parindent=\z@}%
2828 \tolerance=5000%
2829 \hangindent=\csuse{Xhangindent@#4}%
2830 \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2831 \@tempdima=\parindent%
2832 \csuse{Xcolalign@#4}%
2833 \parindent=\@tempdima%
2834 \strut{%
2835 \hspace{\parindent}%
2836 \printlinefootnote{#1}{#4}%
2837 }%
2838 \print@lemma{#1}{#2}{#4}%
2839 \csuse{Xwrapcontent@#4}{#3}%
2840 \strut\par\allowbreak}
2841 %

```

\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.

```

2842 \newcommand*{\threecolfootgroup}[1]{%
2843 \csuse{Xbhookgroup@#1}\par%
2844 \splittopskip=\ht\strutbox
2845 \expandafter
2846 \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
2847 %

```

\mpthreecolfootgroup The setup for minipages.

```

2848 \newcommand*{\mpthreecolfootgroup}[1]{%
2849 \vskip\skip\@nameuse{mp#1footins}
2850 \ifl@dpairing\ifparledgroup%
2851 \leavevmode\marks\parledgroup@{begin}%

```

```

2852 \marks\parledgroup@series{#1}%
2853 \marks\parledgroup@type{Xfootnote}%
2854 \fi\fi\normalcolor
2855 \ifparledgroup%
2856 \ifl@dpairing%
2857 \else%
2858 \setXnoteswidthliketwocolumns@{#1}%
2859 \setXnotespositionliketwocolumns@{#1}%
2860 \print@Xfootnoterule{#1}%
2861 \fi%
2862 \else%
2863 \setXnoteswidthliketwocolumns@{#1}%
2864 \setXnotespositionliketwocolumns@{#1}%
2865 \print@Xfootnoterule{#1}%
2866 \fi%
2867 \csuse{Xbhookgroup@#1}\par%
2868 \splittopskip=\ht\strutbox
2869 \expandafter
2870 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
2871
2872 %

```

Two columns

```

\Xarrangement@twocol 2873 \newcommand*{\Xarrangement@twocol}[1]{%
2874 \csgdef{series@display#1}{twocol}
2875 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
2876 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
2877 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
2878 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2879 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2880 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2881 \twocolfootsetup{#1}
2882 %

```

The additional setup for minipages.

```

2883 \ifnoledgroup@else
2884 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2885 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
2886 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2887 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2888 \mptwocolfootsetup{#1}
2889 \fi
2890 }
2891
2892 %

```

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

`\twocolvfootnote`

`\twocolfootfmt`

`\twocolfootgroup`

```

2893 \newcommand*\twocolfootsetup}[1]{%
2894   \count\csname #1footins\endcsname 500
2895   \csxdef{default@#1footins}{500}%
2896   \multiply\dimen\csname #1footins\endcsname \tw@}
2897   %

2898 \notbool{parapparatus@}\newcommand*\newcommand*\twocolvfootnote}[2]{%
2899   \csuse{Xbeforeinserting@#1}%
2900   \insert\csname #1footins\endcsname\bgroup%
2901   \hsiz=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2902   \noindent\csuse{Xhooknote@#1}%
2903   \csuse{Xnotefontsize@#1}%
2904   \footplitskips%
2905   \csname #1footfmt\endcsname #2{#1}\egroup}
2906   %

2907 \notbool{parapparatus@}\newcommand*\newcommand*\twocolfootfmt}[4]{% 4th
    arg is optional, for backward compatibility
2908   \normal@pars%
2909   \hsiz \csuse{Xhsizetwocol@#4}%
2910   \nottoggle{Xparindent@#4}\parindent=\z@}%
2911   \tolerance=5000%
2912   \hangindent=\csuse{Xhangindent@#4}%
2913   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2914   \@tempdima=\parindent%
2915   \csuse{Xcolalign@#4}%
2916   \parindent=\@tempdima%
2917   \strut{%
2918   \hspace{\parindent}%
2919   \printlinefootnote{#1}{#4}%
2920   }%
2921   \print@lemma{#1}{#2}{#4}%
2922   \csuse{Xwrapcontent@#4}{#3}%
2923   \strut\par\allowbreak}
2924   %

2925 \newcommand*\twocolfootgroup}[1]{%
2926   \csuse{Xhookgroup@#1}\par%
2927   \splittopskip=\ht\strutbox
2928   \expandafter
2929   \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
2930   %
2931   %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

2932 \newcommand*\mptwocolfootsetup}[1]{%
2933   \count\csname mp#1footins\endcsname 500
2934   \multiply\dimen\csname mp#1footins\endcsname \tw@}
2935 %

2936 \newcommand*\mptwocolfootgroup}[1]{%
2937   \vskip\skip\@nameuse{mp#1footins}
2938   \ifl@dpairing\ifparledgroup%
2939     \leavevmode\marks\parledgroup@{begin}%
2940     \marks\parledgroup@series{#1}%
2941     \marks\parledgroup@type{Xfootnote}%
2942     \fi\fi\normalcolor
2943   \ifparledgroup%
2944     \ifl@dpairing%
2945       \else%
2946         \setXnoteswidthliketwocolumns@{#1}%
2947         \setXnotespositionliketwocolumns@{#1}%
2948         \print@Xfootnoterule{#1}%
2949       \fi%
2950     \else%
2951       \setXnoteswidthliketwocolumns@{#1}%
2952       \setXnotespositionliketwocolumns@{#1}%
2953       \print@Xfootnoterule{#1}%
2954     \fi%
2955     \csuse{Xbhookgroup@#1}\par%
2956     \splittopskip=\ht\strutbox
2957     \expandafter
2958     \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
2959
2960 %

```

XII.7 Critical notes presentation

Here, we define some commons macro which are used in order to print a critical notes, that is a note with 1) line number 2) lemma 3) lemma separator 4) text associated to the lemma.

XII.7.1 Font tools

`\endashchar` The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals does not contain an en-dash or square brackets, and its period and comma are in odd locations. To allow use of the standard footnote macros with such fonts, we use the following macros for certain characters.

The `\endashchar` macro is simply an en-dash from the normal font and is immune to changes in the surrounding font. The same goes for the full stop. These two are used in `\printlines`. The right bracket macro is the same again; it crops up in

`\normalfootfmt` and the other footnote macros for controlling the format of the footnotes.

With `polyglossia`, each critical note has a `\footnote@lang` which shows the language of the lemma, and which can be used to switch the bracket from right to left.

```

2961 \def\endashchar{\textnormal{--}}
2962
2963 \newcommand*\fullstop{\textnormal{.}}
2964 \def\Xsublinesep@side{\fullstop}
2965
2966 \newcommand*\rbracket{\textnormal{%
2967   \csuse{text\csuse{footnote@lang}}}%
2968   \ifluatex%
2969     \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]{\thinspace
2970   }}%
2971   \else%
2972     \thinspace}%
2973   \fi}%
2974 }
2975
2976 %

```

XII.7.2 Pstart number in footnote

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

```

2977 \newcommand{\printpstart}[0]{%
2978   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
2979   l@dprintingcolumns}}{%
2980     \ifledRcol%
2981     \thepstartR%
2982   \else%
2983     \thepstartL%
2984   \fi%
2985   }{%
2986     \thepstart%
2987   }%
2988 }

```

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).

```

2989 %
2990 \newcommand{\print@lemma}[3]{%
2991   \bgroup%
2992   \nottoggle{Xlemmadisablefontselection@#3}%

```

```

2993     {\select@lemmafont#1|}%
2994     }%
2995     \bgroup%
2996     \csuse{Xlemmafont@#3}%Deprecated
2997     \csuse{Xwraplemma@#3}{#2}%
2998     \egroup%
2999     \egroup%
3000     \iftoggle{nosep@}{%
3001     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3002     \relax%
3003     }%
3004     {\ifcempty{Xlemmaseparator@#3}%
3005     {%
3006     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3007     \relax%
3008     }%
3009     {%
3010     \nobreak%
3011     \hskip\csuse{Xbeforelemmaseparator@#3}%
3012     \csuse{Xlemmaseparator@#3}%
3013     \hskip\csuse{Xafterlemmaseparator@#3}%
3014     \relax%
3015     }%
3016     }%
3017     }%
3018     %

```

XII.7.4 Line number printing

\printlinefootnote The `\printlinefootnote` macro is called in each `\<type>footfmt` command. It controls whether the line number is printed or not, according to the series options. Its first argument is the information about lines; its second is the series of the footnote. The printing of the line number is shared in `\printlinefootnotenumbers`.

```

3019 \newcommand{\printlinefootnote}[2]{%
3020   \l@dp@rsefootspec#1|%
3021   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3022     \edef\lineinfo{\l@dparsedstartline - \l@dparsedstartsub - \
3023     \l@dparsedendline - \l@dparsedendsub}%
3024     {%
3025     \edef\lineinfo{\l@dparsedstartline - \l@dparsedstartsub}%
3026     }%
3027     \iftoggle{nonum@}{%Try if the line number must printed for this specific
3028     not (by default, yes)
3029     \hspace{\csuse{Xinplaceofnumber@#2}}%
3030     }%
3031     {%

```

```

3032 \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
default, yes)
3033 {%
3034 \hspace{\csuse{Xinplaceofnumber@#2}}%
3035 }%
3036 {%
3037 {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
3038 {%
3039 \ifcsdef{prevline#2}%
3040 {%Be sure the \prevline exists.
3041 \ifcsequal{prevline#2}{\lineinfo@}%Try it
3042 {%
3043 \ifcsequal{Xsymlinenum@#2}%
3044 {%
3045 \hspace{\csuse{Xinplaceofnumber@#2}}%
3046 }%
3047 {\printsymlinefootnotearea{#2}}%
3048 }%
3049 {%
3050 \printlinefootnotearea{#1}{#2}%
3051 }%
3052 }%
3053 {%
3054 \printlinefootnotearea{#1}{#2}%
3055 }%
3056 }%
3057 {%
3058 \printlinefootnotearea{#1}{#2}%
3059 }%
3060 \csxdef{prevline#2}{\lineinfo@}%
3061 }%
3062 }%
3063 }%
3064 }%
3065 }
3066 %

```

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

```

3067 \newcommand{\printsymlinefootnotearea}[1]{%
3068 \hspace{\csuse{Xbeforesymlinenum@#1}}%
3069 \csuse{Xnotenumfont@#1}%
3070 \ifdimequal{\csuse{Xboxsymlinenum@#1}}{\z@}%
3071 {\csuse{Xsymlinenum@#1}}%
3072 {\hbox to \csuse{Xboxsymlinenum@#1}%
3073 {\csuse{Xsymlinenum@#1}\hfill}%
3074 }%
3075 \hspace{\csuse{Xaftersymlinenum@#1}}%

```

```

3076 }%
3077 %

```

\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.)

```

3078 \newcommand{\printlinefootnotearea}[2]{%
3079   \printXbeforenumber{#2}%
3080   \csuse{Xnotenumfont@#2}%
3081   \boxfootnotenumbers{#1}{#2}%
3082   \printXafternumber{#2}%
3083 }%
3084 %

```

\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.

```

3085 \newcommand{\boxfootnotenumbers}[2]{%
3086   \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
3087     \printlinefootnotenumbers{#1}{#2}%
3088   }%
3089   {%
3090     \hbox to \csuse{Xboxlinenum@#2}%
3091     {%
3092       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3093       \printlinefootnotenumbers{#1}{#2}%
3094       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3095     }%
3096   }%
3097 }%
3098 %

```

\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.

```

3099 \newcommand{\printlinefootnotenumbers}[2]{%
3100   \xdef\@currentseries{#2}%
3101   \ifboolexpr{%
3102     (togl{Xpstart@#2} and bool{numberpstart})%
3103     or togl{Xpstarteverytime@#2}}%
3104   {\printpstart}{}%
3105   \iftoggle{Xstanza@#2}{%
3106     \ifnumberstanza%
3107       \printstanza%
3108     \csuse{Xstanzaseparator@#2}%

```

```

3109 \fi%
3110 }{}%
3111 \iftoggle{Xonlypstart@#2}{%
3112 \csuse{Xtxtbeforenumber@#2}%
3113 \printlines#1|\ifledRcol@{\Rlineflag\fi}}%
3114 }%
3115 %

```

\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.)

```

3116 \newcommand{\printXbeforenumber}[1]{%
3117 \hspace{\csuse{Xbeforenumber@#1}}%
3118 }%
3119 %

```

\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

```

3120 \newcommand{\printXafternumber}[1]{%
3121 \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3122 \hspace{\csuse{Xafternumber@#1}}%
3123 }%
3124 %

```

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. 92: the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma.

edmac’ creator have defined six boolean in order to know which component of line number description we have to print:

- `\ifl@d@pnum` for page numbers;
- `\ifl@d@ssub` for starting sub-line;
- `\ifl@d@elin` for ending line;
- `\ifl@d@esl` for ending sub-line; and
- `\ifl@d@dash` for the dash between the starting and ending groups.

There is no boolean for the line number because it is always printed.

Maieul Rouquette has added `\ifl@d@Xtwolines` and `\ifl@d@Xmorethantwolines` to print a symbol which stands for “and subsequent” when there are two, three or more lines.

```

\ifl@d@pnum25 \newif\ifl@d@pnum
\ifl@d@ssub26 \newif\ifl@d@ssub
\ifl@d@elin27 \newif\ifl@d@elin
\ifl@d@esl28 \newif\ifl@d@esl
\ifl@d@dash29 \newif\ifl@d@dash
\ifl@d@Xtwolines30 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines31 \newif\ifl@d@Xmorethantwolines%
312 %

```

```

\l@dp@rsefootsspec \l@dp@rsefootsspec parses lines specification and defines macros which hold the nu-
\l@dparsedstartpage meric values. Just a reminder of the arguments:
\l@dparsedstartline \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dparsedstartsub \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@dparsedendpage313 \def\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
\l@dparsedendline314 \gdef\l@dparsedstartpage{#1}%
\l@dparsedendsub315 \gdef\l@dparsedstartline{#2}%
316 \gdef\l@dparsedstartsub{#3}%
317 \gdef\l@dparsedendpage{#4}%
318 \gdef\l@dparsedendline{#5}%
319 \gdef\l@dparsedendsub{#6}%
3140 }
3141 %

```

Initialise the several number value macros.

```

3142 \def\l@dparsedstartpage{0}%
3143 \def\l@dparsedstartline{0}%
3144 \def\l@dparsedstartsub{0}%
3145 \def\l@dparsedendpage{0}%
3146 \def\l@dparsedendline{0}%
3147 \def\l@dparsedendsub{0}%
3148
3149 %

```

\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`.

```

3150 \newcommand*{\setprintlines}[6]{%
3151   \l@d@pnumfalse \l@d@dashfalse
3152 %

```

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

```

3153 \ifbypage@
3154   \ifnum#4=#1 \else
3155     \l@d@pnumtrue
3156     \l@d@dashtrue
3157   \fi
3158 \fi
3159 %

```

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.

```

3160 \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
3161 \ifnum#2=#5 \else
3162     \l@d@elintrue
3163     \l@d@dashtrue
3164 \fi
3165 %

```

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

```

3166 \l@d@ssubfalse
3167 \ifnum#3=0 \else
3168     \l@d@ssubtrue
3169 \fi
3170 %

```

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.

```

3171 \l@d@eslfalse
3172 \ifnum#6=0 \else
3173     \ifnum#6=#3
3174         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3175     \else
3176         \l@d@esltrue
3177         \l@d@dashtrue
3178     \fi
3179 \fi%
3180 %

```

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

```

3181 \ifl@d@dash%
3182     \ifboolexpr{togl{fulllines@} or test{\ifcsempy{Xtwolines@}\
3183     @currentseries}}}%
3184     {%
3185     \setistwofollowinglines{#1}{#2}{#4}{#5}%
3186     \ifboolexpr{%
3187         (%
3188             togl {Xtwolinesbutnotmore@\@currentseries}%
3189             and not%
3190             (%
3191                 bool {istwofollowinglines@}%
3192             )%
3193         )%
3194         or%
3195         (%
3196             (not test{\ifnumequal{#1}{#4}})%
3197             and togl{Xtwolinesonlyinsamepage@\@currentseries}%

```

```

3198         )%
3199     }%
3200     {}%
3201     {%
3202     \l@d@dashfalse%
3203     \l@d@Xtwolinestrue%
3204     \l@d@elinfalse%
3205     \l@d@eslfalse%
3206     \ifcsempy{Xmorethantwelines@%currentseries}%
3207     {}%
3208     {\ifistwofollowinglines@%else%
3209     \l@d@Xmorethantwolinestrue%
3210     \fi%
3211     }%
3212     }%
3213 }%
3214 \fi%
3215 %

End of \setprintlines.

3216 }%
3217 %

```

`\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.

```

3218 \newif\ifistwofollowinglines@%
3219 \newcommand{\setistwofollowinglines}[4]{%
3220     \ifcsdef{lastlinenumberon@#1}%
3221     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
3222     {\numdef{\tmp}{0}}}%

```

```

3223 \istwofollowinglines@false%
3224 \ifnumequal{#4-#2}{1}%
3225 {\istwofollowinglines@true}%
3226 {\ifbypage%
3227   \ifnumequal{#3-#1}{1}%
3228   {%
3229     \ifnumequal{#2}{\tmp}%
3230     {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
3231     {}}%
3232   }%
3233   {%
3234   \fi%
3235   }%
3236 }%
3237 %

```

`\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

```

3238 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3239   \begingroup%
3240   %

```

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

```

3241 \ifluatex%
3242   \edef\@tmp{\the\textdir}%
3243   \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
    spurious space (bug #397)
3244   \fi%
3245   %

```

Decide which part of line number components we will print.

```

3246 \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3247 %

```

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.

```

3248 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3249   {\bgroup}%
3250   {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
    hfill}%
3251   \ifl@d@pnum%
3252     \wrap@edcrossref{\@this@crossref@start}{#1}%
3253     \csuse{Xsublinesep@\@currentseries}%
3254     \fi%

```

```

3255 \wrap@edcrossref{\@this@crossref@start}{%
3256 \linenumrep{#2}%
3257 \iftoggle{Xlineflag@{\@currentseries}{#8}{}}%
3258 }%
3259 \ifl@d@ssub%
3260 \csuse{Xsublinesep@{\@currentseries}}%
3261 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3262 \fi
3263 \egroup%
3264 %

```

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

```

3265 \ifdimequal{\csuse{Xboxendlinenum@{\@currentseries}}{0pt}%
3266 {\bgroup}%
3267 {\hbox to \csuse{Xboxendlinenum@{\@currentseries}}\bgroup}%
3268 \ifl@d@Xtwolines%
3269 \ifl@d@Xmorethantwolines%
3270 \csuse{Xmorethantwolines@{\@currentseries}}%
3271 \else%
3272 \csuse{Xtwolines@{\@currentseries}}%
3273 \fi%
3274 \else%
3275 \ifl@d@dash%
3276 \ifdefined\linangesep%
3277 \linangesep%
3278 \else%
3279 \csuse{Xlinangeseparator@{\@currentseries}}%
3280 \fi%
3281 \fi%
3282 \ifl@d@pnum%
3283 \wrap@edcrossref{\@this@crossref@end}{#4}%
3284 \csuse{Xsublinesep@{\@currentseries}}%
3285 \fi%
3286 \ifl@d@elin%
3287 \wrap@edcrossref{\@this@crossref@end}{%
3288 \linenumrep{#5}%
3289 \iftoggle{Xlineflag@{\@currentseries}{#8}{}}%
3290 }%
3291 \fi%
3292 \ifl@d@esl%
3293 \ifl@d@elin%
3294 \csuse{Xsublinesep@{\@currentseries}}%
3295 \fi%
3296 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3297 \fi%
3298 \fi%
3299 \ifdimequal{\csuse{Xboxendlinenum@{\@currentseries}}{0pt}%
3300 {}%
3301 {\hfill}}%Prevent underfull hbox
3302 \egroup%

```

```

3303 \endgroup%
3304 }%
3305 %

```

XIII Familiar footnotes

XIII.1 Adjacent footnotes

The original edmac provided users with five series of critical footnotes (`\Afootnote` `\Bfootnote` `\Cfootnote` `\Dfootnote` `\Efootnote`), and \LaTeX provides a single numbered footnote. The `reledmac` package uses the edmac mechanism to provide six series of numbered footnotes.

First, though, the `footmisc` package has an option whereby two or more consecutive footnotes have their marks separated by commas. This seemed to Peter Wilson such a useful ability that it was provided automatically by `eledmac`.

Maïeul Rouquette has maintained this feature in `reledmac`, despite he thought that is not directly in relationship with the aim of `reledmac`.

`\multiplefootnotemarker` These macros may have been defined by the `memoir` class, are provided by the `footmisc` package and perhaps by other footnote packages. That is why we use `\providecommand` and not `\newcommand`.

`\multfootsep`

```

3306 \providecommand*\multiplefootnotemarker}{3sp}
3307 \providecommand*\multfootsep}{\textsuperscript{\normalfont,}}
3308
3309 %

```

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

```

3310 \providecommand*\m@mmf@prepare}{%
3311 \kern-\multiplefootnotemarker
3312 \kern\multiplefootnotemarker\relax}
3313 %

```

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

```

3314 \providecommand*\m@mmf@check}{%
3315 \ifdim\lastkern=\multiplefootnotemarker\relax
3316 \edef\x@sf{\the\spacefactor}%
3317 \unkern
3318 \multfootsep
3319 \spacefactor\x@sf\relax
3320 \fi}
3321
3322 %

```

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

```
3323 \@ifclassloaded{memoir}{}{%
3324 %
```

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

```
3325 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
3326 %
```

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

```
3327
3328 \patchcmd{\@footnotemark}
3329   {\nobreak}
3330   {\m@mmf@check
3331    \nobreak
3332   }
3333   {}{}
3334 \patchcmd{\@footnotemark}
3335   {\@makefnmark}
3336   {\@makefnmark
3337    \m@mmf@prepare
3338   }
3339   {}{}
3340 %
```

Finished the modifications for the non-memoir case.

```
3341 }
3342
3343 %
```

XIII.2 Regular footnotes for numbered texts

`\l@doldold@footnotetext` In order to enable the regular footnotes in numbered text we have to play around with its `\@footnotetext`, using different forms for when in numbered or regular text.

```
3344 \pretocmd{\@footnotetext}{%
3345   \ifnumberedpar@
3346   \edtext{}{\l@dbfnote{#1}}}%
3347   \else
3348   }{}{}
3349 \apptocmd{\@footnotetext}{\fi}{}{}%
3350 %
```

```

\l@dbfnote \l@dbfnote adds the footnote to the insert list, and \vl@dbfnote calls the original
\vl@dbfnote \@footnotetext. We also patch \footnote in order to get the correct footnote
\vl@dbfnote numbers when typesetting parallel texts. This is moved into a \get@fnmark command.
\footnote
\get@fnmark
\get@thisfootnote
3351
3352 \patchcmd%
3353   {\footnote}%
3354   {\stepcounter\@mpfn}%
3355   {%
3356   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3357     \global\advance\footnote@reading by \@ne%
3358     \get@thisfootnote%
3359     \get@fnmark{\thisfootnote}%
3360     \ifcsdef{footnotereading\the\footnote@reading=typeset}%
3361       {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%
3362       {\setcounter{\@mpfn}{\footnote@reading}}%
3363     }{%
3364       \stepcounter\@mpfn%
3365     }%
3366   }%
3367   {}
3368   {}
3369
3370 \newcommand{\get@thisfootnote}{%
3371   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3372     \protected@xdef\thisfootnote{\the\footnote@reading}%
3373   }{%
3374     \protected@xdef\thisfootnote{\the\c@footnote}%
3375   }%
3376 }%
3377
3378 \newcommand{\l@dbfnote}[1]{%
3379   \get@thisfootnote%
3380   \gdef\@tag{#1\relax}%
3381   \ifledRcol%
3382     \xright@appenditem{%
3383       \ifdefined\Hy@footnote@currentHref%
3384         \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3385       \fi%
3386       \noexpand\vl@dbfnote{\expandonce\@tag}{\thisfootnote}%
3387     }%
3388     \to\inserts@listR
3389     \global\advance\insert@countR \@ne%
3390   \else%
3391     \xright@appenditem{%

```

```

3392         \ifdefined\Hy@footnote@currentHref%
3393             \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3394         \fi%
3395         \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisfootnote}%
3396     }%
3397         \to\inserts@list
3398     \global\advance\insert@count \one%
3399     \fi
3400     \ignorespaces%
3401 }%
3402
3403 \newcommand{\get@fnmark}[1]{%
3404     \ifboolexpr{bool{1@dpairing} or bool{1@dprintingpages} or bool{
1@dprintingcolumns}}{%
3405         {%
3406             \stepcounter{footnote@typeset}%
3407             \setcounter{footnote}{\c@footnote@typeset}%
3408             \immediate\write\@mainaux{%
3409                 \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
3410             }%
3411             \def\@thefnmark{\thefootnote}%
3412         }%
3413         {%
3414             \@namedef{@thefnmark}{#1}%
3415         }%
3416     }%
3417
3418 \newcommand{\vl@dbfnote}[2]{%
3419     \get@fnmark{#2}%
3420     \@footnotetext{#1}%
3421 }%
3422 %

```

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`

```

3423 \newcommand*\prebodyfootmark{%
3424     \leavevmode
3425     \ifhmode
3426         \edef\@xsf{\the\spacefactor}%
3427         \m@mmf@check
3428         \nobreak
3429     \fi}
3430 \newcommand*\postbodyfootmark{%

```

```

3431 \m@mmf@prepare
3432 \ifhmode\spacefactor\@x@sf\fi\relax}
3433
3434 %

```

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.

```

3435 \newcommandx{\arrangementX}[2][1,usedefault]{%
3436   \def\do##1{%
3437     \csname arrangementX@#2\endcsname{##1}%
3438   }%
3439   \ifstrepty{#1}%
3440     {%
3441       \dolistloop{\@series}%
3442     }%
3443     {
3444       \docsvlist{#1}%
3445     }%
3446   }%
3447   %

```

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.

```

3448 \newcommand*{\normal@footnotemarkX}[1]{%
3449   \prebodyfootmark
3450   \wrapped@bodyfootmarkX{#1}%
3451   \postbodyfootmark}
3452
3453 %

```

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

```

3454 \newcommand*{\normalbodyfootmarkX}[1]{%
3455   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}}
3456 %

```

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

```

3457 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnoteX}[2]{%
3458   \csuse{beforeinsertingX@#1}%
3459   \insert\@nameuse{footins#1}\bgroup
3460     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3461     \noindent\csuse{bhooknoteX@#1}%
3462     \csuse{notefontsizeX@#1}%
3463     \footsplitskips
3464     \ifl@dpairing\ifl@dpaging\else%
3465       \setnoteswidthliketwocolumnsX@{#1}%
3466     \fi\fi%
3467     \setnotesXpositionliketwocolumns@{#1}%
3468     \spaceskip=\z@skip \xspaceskip=\z@skip
3469     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
3470
3471 %

```

`\mpnormalvfootnoteX` The minipage version.

```

3472 \newcommand*{\mpnormalvfootnoteX}[2]{%
3473   \get@thisfootnoteX{#1}%
3474   \get@fnmarkX{#1}{\thisfootnote}%
3475   \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3476   \global\setbox\@nameuse{mpfootins#1}\vbox{%
3477     \unvbox\@nameuse{mpfootins#1}
3478     \noindent\csuse{bhooknoteX@#1}%
3479     \csuse{notefontsizeX@#1}%
3480     \hsize\columnwidth
3481     \@parboxrestore
3482     \color@beginninggroup
3483     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
3484
3485 %

```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

3486 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmtX}[2]{%
3487   \ifluatex%
3488     \textdir\footnote@luatextextdir%
3489     \pardir\footnote@luatexpardir%
3490     \par%
3491   \fi%
3492   \protected@edef\@currentlabel{%
3493     \@nameuse{@thefnmark#1}%
3494   }%
3495   \ledsetnormalparstuffX{#1}%
3496   \hangindent=\csuse{hangindentX@#1}%
3497   \everypar{\hangindent=\csuse{hangindentX@#1}}%
3498   \rule\z@{\splittopskip}%
3499   {\csuse{notenumberX@#1}\wrapped@footfootmarkX{#1}}%
3500   \csuse{wrapcontentX@#1}{#2}%

```

```

3501 \strut\par}}
3502
3503 %

```

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

```

3504 \newcommand*{\normalfootfootmarkX}[1]{%
3505   \textsuperscript{\@nameuse{@thefnmark#1}}}%
3506
3507 %

```

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

```

3508 \newcommand*{\normalfootstartX}[1]{%
3509   \ifdimequal{0pt}{\prenotesX@}{}%
3510   {%
3511     \iftoggle{prenotesX@}{%
3512       \togglefalse{prenotesX@}%
3513       \skip\csname footins#1\endcsname=%
3514       \glueexpr\csuse{prenotesX@}+\csuse{aftererruleX@#1}\relax%
3515     }%
3516     {}%
3517   }%
3518   \vskip\skip\csname footins#1\endcsname%
3519   \leftskip=\z@
3520   \rightskip=\z@
3521   \ifl@dpairing\else%
3522     \hsize=\old@hsize%
3523   \fi%
3524   \setnoteswidthliketwocolumnsX@{#1}%
3525   \setnotesXpositionliketwocolumns@{#1}%
3526   \print@footnoteXrule{#1}%
3527 }%
3528
3529 %

```

\normalfootnoteruleX The rule drawn before the footnote series group.

```

3530 \let\normalfootnoteruleX=\footnoterule
3531
3532 %

```

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

```

3533 \newcommand*{\normalfootgroupX}[1]{%
3534   \csuse{bhookgroupX@#1}%
3535   \unvbox\@nameuse{footins#1}%

```

```

3536 \hsize=\old@hsize%
3537 }%
3538
3539 %

```

`\mpnormalfootgroupX` The minipage version.

```

3540 \newcommand*{\mpnormalfootgroupX}[1]{%
3541 \vskip\skip\@nameuse{mpfootins#1}
3542 \ifl@dpairing\ifparledgroup%
3543 \leavevmode\marks\parledgroup@{begin}%
3544 \marks\parledgroup@series{#1}%
3545 \marks\parledgroup@type{footnoteX}%
3546 \fi\fi\normalcolor
3547 \ifparledgroup%
3548 \ifl@dpairing%
3549 \else%
3550 \setnoteswidthliketwocolumnsX@{#1}%
3551 \setnotesXpositionliketwocolumns@{#1}%
3552 \print@footnoteXrule{#1}%
3553 \fi%
3554 \else%
3555 \setnoteswidthliketwocolumnsX@{#1}%
3556 \setnotesXpositionliketwocolumns@{#1}%
3557 \print@footnoteXrule{#1}%
3558 \fi%
3559 \csuse{bhookgroupX@#1}%
3560 \unvbox\@nameuse{mpfootins#1}}
3561
3562 %

```

`\normalbfnoteX`₆₃

```

3564 \newcommand{\normalbfnoteX}[2]{%
3565 \get@thisfootnoteX{#1}%
3566 \ifledRcol%
3567 \ifluatex
3568 \footnotelang@lua[R]%
3569 \fi
3570 \@ifundefined{xpg@main@language}%if polyglossia
3571 {}%
3572 {\footnotelang@poly[R]}%
3573 \xright@appenditem{%
3574 \noexpand\led@set@index@fornote{#1}%
3575 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3576 \noexpand\vbfnoteX{#1}{#2}{\expandonce\thisfootnote}%
3577 \noexpand\led@reinit@index@fornote%
3578 }%
3579 \to\inserts@listR

```

```

3580 \global\advance\insert@countR \@ne%
3581 \else%
3582 \ifluatex
3583 \footnotelang@lua%
3584 \fi
3585 \@ifundefined{xpg@main@language}%if polyglossia
3586 {}%
3587 {\footnotelang@poly}%
3588 \xright@appenditem{%
3589 \noexpand\led@set@index@fornote{#1}%
3590 \unexpanded{\def\this@footnoteX@reading{\the\csname footnote#1
@reading\endcsname}%
3591 \noexpand\vbfnoteX{#1}{#2}{\expandonce\thisfootnote}%
3592 \noexpand\led@reinit@index@fornote%
3593 }%
3594 \to\inserts@list
3595 \global\advance\insert@count \@ne%
3596 \fi
3597 \ignorespaces}
3598
3599 %

```

\get@thisfootnoteX The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

3600 \newcommand{\get@thisfootnoteX}[1]{%
3601 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3602 \protected@xdef\thisfootnote{\the\csname footnote#1@reading\endcsname
}%
3603 }{%
3604 \protected@xdef\thisfootnote{\the\csname c@footnote#1\endcsname}%
3605 }%
3606 }%
3607 %

```

\vbfnoteX This command calls the correct footnote-inserting commands.

```

3608 \newcommand{\vbfnoteX}[3]{%
3609 \get@fnmarkX{#1}{#3}%
3610 \@nameuse{regvfootnote#1}{#1}{#2}%
3611 }%
3612
3613 %

```

\get@fnmarkX This command gets the correct footnote number when typesetting parallel texts.

```

3614 \newcommand{\get@fnmarkX}[2]{%
3615 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%

```

```

3616   {%
3617     \stepcounter{footnote#1@typeset}%
3618     \setcounter{footnote#1}{\value{footnote#1@typeset}}%
3619     \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3620     \immediate\write\@mainaux{%
3621       \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
@typeset\endcsname}%
3622     }%
3623   }%
3624   {%
3625     \@namedef{@thefnmark#1}{#2}%
3626   }%
3627 }
3628 %
3629 %

```

```

\newcommand{\vnumfootnoteX}[2]{%
3631   \ifnumberedpar@
3632     \edtext{}{\normalbfnoteX{#1}{#2}}%
3633   \else
3634     \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3635     \get@thisfootnoteX{#1}%
3636     \get@fnmarkX{#1}{\expandonce\thisfootnote}%
3637     \@nameuse{regvfootnote#1}{#1}{#2}%
3638   \fi}
3639
3640 %

```

`arrangementX@normal` `\arrangementX@normal{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```

3641 \newcommand*{\arrangementX@normal}[1]{%
3642   \csgdef{series@displayX#1}{normal}
3643   \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
3644   \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
3645   \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
3646   \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
3647   \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
3648   \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
3649   \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
3650   \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
3651   \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3652   \count\csname footins#1\endcsname=1000
3653   \csxdef{default@footins#1}{1000}%Use to have note only for one side
3654   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3655   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3656   \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3657 }

```

Additions for minipages.

```

3658 \ifnoledgroup@else%
3659 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3660 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
3661 \count\csname mpfootins#1\endcsname=1000
3662 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}%
3663 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3664 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3665 \fi
3666 }
3667
3668 %

```

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 3669 \newcommand*\arrangementX@twocol}[1]{%
3670 \csgdef{series@displayX#1}{twocol}
3671 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
3672 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
3673 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
3674 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3675 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3676 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3677 \twocolfootsetupX{#1}
3678 \ifnoledgroup@else%
3679 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3680 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
3681 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3682 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
3683 \mptwocolfootsetupX{#1}
3684 \fi%
3685 }
3686
3687 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
3688 \newcommand*\twocolfootsetupX}[1]{%
3689 \count\csname footins#1\endcsname 500
3690 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
3691 \multiply\dimen\csname footins#1\endcsname by \tw@}
3692 \newcommand*\mptwocolfootsetupX}[1]{%
3693 \count\csname mpfootins#1\endcsname 500
3694 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
3695
3696 %

```

`\twocolvfootnoteX` `\twocolvfootnoteX{<series>}`

```

3697 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
3698   \csuse{beforeinsertingX@#1}%
3699   \insert\csname footins#1\endcsname\bgroup%
3700     \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3701     \noindent\csuse{bhooknoteX@#1}%
3702     \csuse{notefontsizeX@#1}%
3703     \footssplitsskip%
3704     \spaceskip=\z@skip \xspaceskip=\z@skip%
3705     \@nameuse{footfmt#1}{#1}{#2}\egroup}
3706
3707 %

```

`\twocolfootfmtX` `\twocolfootfmtX{<series>}`

```

3708 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
3709   \protected@edef\@currentlabel{%
3710     \@nameuse{@thefnmark#1}%
3711   }%
3712   \normal@pars%
3713   \hangindent=\csuse{hangindentX@#1}%
3714   \everypar{\hangindent=\csuse{hangindentX@#1}}%
3715   \hspace \csuse{hsizetwocolX@#1}%
3716   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3717   \tolerance=5000\relax%
3718   \par%
3719   \@tempdima=\parindent%
3720   \csuse{colalignX@#1}%
3721   \parindent=\@tempdima%
3722   {\hspace{\parindent}%
3723     \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
3724     \csuse{wrapcontentX@#1}{#2}%
3725     \strut\par}%
3726   \allowbreak%
3727 }%
3728
3729 %

```

`\twocolfootgroupX` `\twocolfootgroupX{<series>}`
`\mptwocolfootgroupX`

```

3730 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
3731   notefontsizeX@#1}
3732   \splittopskip=\ht\strutbox
3733   \expandafter
3734   \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
3735
3736 \newcommand*{\mptwocolfootgroupX}[1]{%
3737   \vskip\skip\@nameuse{mpfootins#1}
3738   \ifl@dpairing\ifparledgroup%

```

```

3738 \leavevmode\marks\parledgroup@{begin}%
3739 \marks\parledgroup@series{#1}%
3740 \marks\parledgroup@type{footnoteX}%
3741 \fi\fi\normalcolor
3742 \ifparledgroup%
3743 \ifl@dpairing%
3744 \else%
3745 \setnoteswidthliketwocolumnsX@{#1}%
3746 \setnotesXpositionliketwocolumns@{#1}%
3747 \print@footnoteXrule{#1}%
3748 \fi%
3749 \else%
3750 \setnoteswidthliketwocolumnsX@{#1}%
3751 \setnotesXpositionliketwocolumns@{#1}%
3752 \print@footnoteXrule{#1}%
3753 \fi%
3754 \csuse{bhookgroupX@#1}%
3755 \splittopskip=\ht\strutbox
3756 \expandafter
3757 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
3758
3759 %

```

XIII.4.4 Three columns footnotes

The following macros set footnotes in three columns. It is assumed that the length of each footnote is less than the column width.

```

\arrangementX@threecol  \newcommand*{\arrangementX@threecol}[1]{%
3761 \csgdef{series@displayX#1}{threecol}
3762 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
3763 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
3764 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
3765 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3766 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3767 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3768 \threecolfootsetupX{#1}
3769 \ifnoledgroup@else%
3770 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3771 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
3772 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3773 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
3774 \mpthreecolfootsetupX{#1}
3775 \fi%
3776 }
3777
3778 %

```

```

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX

```

```

3779 \newcommand*{\threecolfootsetupX}[1]{%
3780   \count\csname footins#1\endcsname 333
3781   \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
3782   \multiply\dimen\csname footins#1\endcsname by \thr@@}
3783 \newcommand*{\mpthreecolfootsetupX}[1]{%
3784   \count\csname mpfootins#1\endcsname 333
3785   \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
3786
3787 %

```

`\threecolvfootnoteX` `\threecolvfootnoteX{<series>}{<text>}`

```

3788 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{%
%
3789   \csuse{beforeinsertingX@#1}%
3790   \insert\csname footins#1\endcsname\bgroup%
3791     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3792     \noindent\csuse{bhooknoteX@#1}%
3793     \csuse{notefontsizeX@#1}%
3794     \footsplitskips%
3795     \@nameuse{footfmt#1}{#1}{#2}\egroup}
3796
3797 %

```

`\threecolfootfmtX` `\threecolfootfmtX{<series>}`

```

3798 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
3799   \protected@edef\@currentlabel{%
3800     \@nameuse{@thefnmark#1}%
3801   }%
3802   \hangindent=\csuse{hangindentX@#1}%
3803   \everypar{\hangindent=\csuse{hangindentX@#1}}%
3804   \normal@pars%
3805   \hsize \csuse{hsizethreecolX@#1}%
3806   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3807   \tolerance=5000\relax%
3808   \@tempdima=\parindent%
3809   \csuse{colalignX@#1}%
3810   \parindent=\@tempdima%
3811   {\hspace{\parindent}%
3812     \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
3813     \csuse{wrapcontentX@#1}{#2}%
3814     \strut\par}\allowbreak}
3815
3816 %

```

`\threecolfootgroupX` `\threecolfootgroupX{<series>}`
`\mpthreecolfootgroupX`

```

3817 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3818 \splittopskip=\ht\strutbox
3819 \expandafter
3820 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
3821
3822 \newcommand*{\mpthreecolfootgroupX}[1]{\%
3823 \vskip\skip\@nameuse{mpfootins#1}
3824 \ifl@dpairing\ifparledgroup
3825 \leavevmode\marks\parledgroup@{begin}%
3826 \marks\parledgroup@series{#1}%
3827 \marks\parledgroup@type{footnoteX}%
3828 \fi\fi\normalcolor
3829 \ifparledgroup%
3830 \ifl@dpairing%
3831 \else%
3832 \setnoteswidthliketwocolumnsX@{#1}%
3833 \setnotesXpositionliketwocolumns@{#1}%
3834 \print@footnoteXrule{#1}%
3835 \fi%
3836 \else%
3837 \setnoteswidthliketwocolumnsX@{#1}%
3838 \setnotesXpositionliketwocolumns@{#1}%
3839 \print@footnoteXrule{#1}%
3840 \fi%
3841 \csuse{bhookgroupX@#1}%
3842 \splittopskip=\ht\strutbox
3843 \expandafter
3844 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
3845
3846 %

```

XIII.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@threecol` `\footparagraphX{<series>}`

```

3847 \newcommand*{\arrangementX@paragraph}[1]{\%
3848 \csgdef{series@displayX#1}{paragraph}%
3849 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
3850 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
3851 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
3852 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
3853 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3854 \count\csname footins#1\endcsname=1000
3855 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
3856 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3857 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%

```

```

3858 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3859 \para@footsetupX{#1}
3860 \ifnoledgroup@else
3861   \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
3862   \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
3863   \count\csname mpfootins#1\endcsname=1000
3864   \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
3865   \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3866   \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3867 \fi
3868 }
3869
3870 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

3871 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3872 \setnoteswidthliketwocolumnsX@{#1}%
3873 \ifcsempy{widthX@#1}%
3874   {}%
3875   {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
3876 \dimen0=\baselineskip
3877 \multiply\dimen0 by 1024
3878 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
%
3879 \expandafter
3880 \xdef\csname footfudgefactor#1\endcsname{%
3881   \expandafter\strip@pt\dimen0 }}
3882
3883 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

3884 \newcommand*{\parafootstartX}[1]{%
3885   \ifdimequal{Opt}{\prenotesX@}{}%
3886   {%
3887     \iftoggle{prenotesX@}{%
3888       \togglefalse{prenotesX@}%
3889       \skip\csname footins#1\endcsname=%
3890       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3891     }%
3892   }%
3893 }%
3894 \leftskip=\z@
3895 \rightskip=\z@
3896 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3897 \vskip\skip@nameuse{footins#1}%
3898 \setnoteswidthliketwocolumnsX@{#1}%
3899 \setnotesXpositionliketwocolumns@{#1}%

```

```

3900 \print@footnoteXrule{#1}%
3901 }
3902
3903 %

```

```

\para@vfootnoteX \para@vfootnoteX{<series>}{<text>}
\mppara@vfootnoteX

```

```

3904 \newcommand*{\para@vfootnoteX}[2]{%
3905   \csuse{beforeinsertingX@#1}%
3906   \insert\csname footins#1\endcsname%
3907   \bgroup
3908     \csuse{notefontsizeX@#1}
3909     \footsplitskips
3910     \setbox0=\vbox{\hsize=\maxdimen%
3911       \let\bidir@RTL@everypar\@empty%
3912       \noindent\csuse{bhooknoteX@#1}%
3913       \@nameuse{footfmt#1}{#1}{#2}}%
3914     \setbox0=\hbox{\unvvhX{0}{#1}}%
3915     \dp0=\z@
3916     \ht0=\csname footfudgefactor#1\endcsname\wd0
3917     \box0
3918     \penalty0
3919   \egroup}
3920 \newcommand*{\mppara@vfootnoteX}[2]{%
3921   \get@thisfootnoteX{#1}%
3922   \get@fnmarkX{#1}{\thisfootnote}%
3923   \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3924   \global\setbox\@nameuse{mpfootins#1}\vbox{%
3925     \unvvhbox\@nameuse{mpfootins#1}
3926     \csuse{notefontsizeX@#1}
3927     \footsplitskips
3928     \setbox0=\vbox{\hsize=\maxdimen%
3929       \let\bidir@RTL@everypar\@empty%
3930       \noindent\color@begingroup%
3931       \csuse{bhooknoteX@#1}%
3932       \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
3933     \setbox0=\hbox{\unvvhX{0}{#1}}%
3934     \dp0=\z@
3935     \ht0=\csname footfudgefactor#1\endcsname\wd0
3936     \box0
3937     \penalty0}}
3938
3939 %

```

```

\unvvhX\newcommand*{\unvvhX}[2]{% 2th is optional for retro-compatibility
3941   \setbox0=\vbox{\unvvhbox#1%
3942     \global\setbox1=\lastbox}%
3943   \unhbox1
3944   \unskip           % remove \rightskip,

```

```

3945 \unskip          % remove \parfillskip,
3946 \unpenalty        % remove \penalty of 10000,
3947 \hskip\csuse{afternoteX@#2}%
3948 \relax}% but add the glue to go between the notes
3949
3950 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

3951 \newcommand*{\parafootfmtX}[2]{%
3952   \protected@edef\@currentlabel{%
3953     \@nameuse{@thefnmark#1}%
3954   }%
3955   \insertparafootsepX{#1}%
3956   \ledsetnormalparstuff@common%
3957   {\csuse{notenumfontX@#1}%
3958    \csuse{notenumfontX@#1}%
3959    \wrapped@footfootmarkX{#1}%
3960    \strut%
3961    \csuse{wrapcontentX@#1}{#2}%
3962    \penalty-10}}
3963
3964 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`
`\mppara@footgroupX`

```

3965 \newcommand*{\para@footgroupX}[1]{%
3966   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3967   \unvbox\csname footins#1\endcsname
3968   \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
3969   \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
3970   \makeboxofhboxes
3971   \setbox0=\hbox{\unhbox0 \removehboxes}%
3972   \csuse{bhookgroupX@#1}
3973   \csuse{notefontsizeX@#1}
3974   \unhbox0\par}
3975
3976 \newcommand*{\mppara@footgroupX}[1]{%
3977   \setnoteswidthliketwocolumnsX@{#1}%
3978   \vskip\skip\@nameuse{mpfootins#1}
3979   \ifl@dpairing\ifparledgroup
3980     \leavevmode%
3981     \leavevmode\marks\parledgroup@{begin}%
3982     \marks\parledgroup@series{#1}%
3983     \marks\parledgroup@type{footnoteX}%
3984     \fi\fi\normalcolor
3985     \ifparledgroup%
3986       \ifl@dpairing%
3987       \else%
3988         \setnoteswidthliketwocolumnsX@{#1}%

```

```

3989 \setnotesXpositionliketwocolumns@{#1}%
3990 \print@footnoteXrule{#1}%
3991 \fi%
3992 \else%
3993 \setnoteswidthliketwocolumnsX@{#1}%
3994 \setnotesXpositionliketwocolumns@{#1}%
3995 \print@footnoteXrule{#1}%
3996 \fi%
3997 \unvbox\csname mpfootins#1\endcsname
3998 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
3999 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4000 \makehboxofhboxes
4001 \setbox0=\hbox{\unhbox0 \removehboxes}%
4002 \csuse{bhookgroupX@#1}%
4003 \csuse{notefontsizeX@#1}%
4004 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4005 \unhbox0\par}}
4006
4007 %

```

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

```

\insertparafootsepX%
4008 \newcommand{\insertparafootsepX}[1]{%
4009 \ifledRcol@%
4010 \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
4011 {\csuse{Xparafootsep@#1}}%
4012 {}%
4013 \global\csname prevpage#1@numR\endcsname=\page@numR%
4014 \else%
4015 \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
4016 {\csuse{Xparafootsep@#1}}%
4017 {}%
4018 \global\csname prevpage#1@num\endcsname=\page@num%
4019 \fi%
4020 }
4021 %

```

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.

```

4022 \newcommand{\wrapped@footfootmarkX}[1]{%
4023 \ifdefined\hypertarget%
4024 \hyperlink%
4025 {@bodyfootmark#1@\this@footnoteX@reading}%
4026 {\@nameuse{footfootmark#1}}%

```

```

4027 \Hy@raisedlink{%
4028   \hypertarget%
4029   {@footnotemark#1@\this@footnoteX@reading}%
4030   {}%
4031 }%
4032 \else%
4033   \@nameuse{footfootmark#1}%
4034 \fi%
4035 }%
4036 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package’s commands, if needed.

```

4037 \newcommand{\wrapped@bodyfootmarkX}[1]{%
4038   \ifdefined\hypertarget%
4039     \hyperlink%
4040     {@footnotemark#1@\expandafter\the\csname footnote#1@reading\
endcsname}%
4041     {@@nameuse{bodyfootmark#1}}}%
4042   \Hy@raisedlink{%
4043     \hypertarget%
4044     {@bodyfootmark#1@\expandafter\the\csname footnote#1@reading\
endcsname}%
4045     {}%
4046   }%
4047 \else%
4048   \@nameuse{bodyfootmark#1}%
4049 \fi%
4050 }%
4051 %

```

XIV Code common to both critical and familiar footnote in normal arrangement

`\par` should always be redefined to `\endgraf` within the format macro (this is what `\normal@pars` does), to override tricky material in the main text to get the lines numbered automatically (as set up by `\autopar`, for example).

In the case of footnote arranged in a “normal” way, we also must set some setting for paragraph indent and text direction when using `LuaLaTeX`.

That why we have defined `\ledsetnormalparstuff@common` in order to make this setting for both familiar and critical notes. This command is called by command to make specific setting to critical or familiar footnote.

```

\ledsetnormalparstuff@common%52 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff%53   \ifluatex%
\ledsetnormalparstuffX%54   \textdir\footnote@luatextextdir%

```

```

4055 \pardir\footnote@luatexpardir%
4056 \fi%
4057 \csuse{\csuse{footnote@dir}}%
4058 \normal@pars%
4059 \parfillskip \z@ \@plus 1fil}%
4060
4061 \newcommand*{\Xledsetnormalparstuff}[1]{%
4062 \ledsetnormalparstuff@common%
4063 \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}}%
4064 }%
4065
4066 \newcommand*{\ledsetnormalparstuffX}[1]{%
4067 \ledsetnormalparstuff@common%
4068 \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}}%
4069 }%
4070 %

```

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.

`\noteswidthliketwocolumns@`

`\noteswidthliketwocolumnsX@`

```

4071
4072 \newdimen\old@hsize%
4073 \AtBeginDocument{\old@hsize=\hsize}%
4074
4075 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
4076 \global\let\hsize@fornote=\hsize%
4077 \global\old@hsize=\hsize%
4078 \let\old@columnwidth=\columnwidth%
4079 \iftoggle{Xnoteswidthliketwocolumns@#1}%
4080 {%
4081 \csuse{setwidthliketwocolumns@\columns@position}%
4082 \global\let\hsize@fornote=\hsize%
4083 }%
4084 }%
4085 \let\hsize=\hsize@fornote%
4086 \let\columnwidth=\old@columnwidth%
4087 }%
4088
4089 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4090 \global\let\hsize@fornote=\hsize%
4091 \global\old@hsize=\hsize%
4092 \let\old@columnwidth=\columnwidth%

```

```

4093 \iftoggle{noteswidthliketwocolumnsX@#1}{%
4094   {%
4095     \csuse{setwidthliketwocolumns@\columns@position}%
4096     \global\let\hsize@fornote=\hsize%
4097   }%
4098   {%
4099     \let\hsize=\hsize@fornote%
4100     \let\columnwidth=\old@columnwidth%
4101   }%
4102
4103 %

```

`\setXnotespositionliketwocolumns@` These two commands set the position of the critical / familiar footnotes, depending on the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They call commands which are defined only in `reledpar`, because this feature has no sens without `reledpar`.

`\setnotesXpositionliketwocolumns@`

```

4104 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
4105   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4106     \csuse{setnotespositionliketwocolumns@\columns@position}%
4107   }{%
4108   }%
4109
4110 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
4111   \iftoggle{noteswidthliketwocolumnsX@#1}{%
4112     \csuse{setnotespositionliketwocolumns@\columns@position}%
4113   }{%
4114   }%
4115
4116 %

```

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.

```

4117 \def\@fnpos{familiar-critical}
4118 \def\@mpfnpos{critical-familiar}
4119 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
4120 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
4121 %

```

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@Xfootnoterule22 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoteXrule23 \vskip-\csuse{Xafterterrule@#1}%Because count in \dimen\csuse{#1footins}
4124 \nointerlineskip%
4125 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
4126 \nointerlineskip%
4127 \vskip\csuse{Xafterterrule@#1}%
4128 }%
4129
4130 \newcommand{\print@footnoteXrule}[1]{%
4131 \vskip-\csuse{afterterruleX@#1}%Because count in \dimen\csuse{footins#1}
4132 \nointerlineskip%
4133 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
4134 \nointerlineskip%
4135 \vskip\csuse{afterterruleX@#1}%
4136 }%
4137
4138 %

```

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.

```

4139 \newtoggle{Xprenotes@}%
4140 \toggletrue{Xprenotes@}%
4141 \newcommand{\Xprenotes@}{Opt}%
4142 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
4143 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
4144 %

```

The same, but for familiar footnotes.

```

\Xprenotes@ \newtoggle{prenotesX@}
\Xprenotes@ \toggletrue{prenotesX@}
4147 \newcommand{\prenotesX@}{Opt}
4148 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
4149 %

```

XVIII.0.3 Internal commands

```

firstXseries@ \gdef\firstXseries@{}
prepare@Xprenotes@ \newcommand{\prepare@Xprenotes}[1]{%
4152   \ifdimequal{Opt}{\Xprenotes@}%
4153   {%
4154   {%
4155     \IfStrEq{\firstXseries@}{}{%
4156       \global\skip\csuse{#1footins}=\Xprenotes@%
4157       \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4158       \gdef\firstXseries@{#1}%
4159     }%
4160   }%
4161   \ifseriesbefore{#1}{\firstXseries@}%
4162   {%
4163     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
4164     \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4165     \gdef\firstXseries@{#1}%
4166   }%
4167   }%

```

```

4168 }%
4169 }%
4170 }
4171 %

```

The same thing is required for familiar notes and `\prenotesX`.

```

firstseriesX@ \gdef\firstseriesX@{}
prepare@prenotesX@ \newcommand{\prepare@prenotesX}[1]{%
4174 \ifdimequal{0pt}{\prenotesX@}%
4175 {}%
4176 {%
4177 \IfStrEq{\firstseriesX@}{%
4178 \global\skip\csuse{footins#1}=\prenotesX@%
4179 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4180 \gdef\firstseriesX@{#1}%
4181 }%
4182 {%
4183 \ifseriesbefore{#1}{\firstseriesX@}%
4184 {%
4185 \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
4186 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4187 \gdef\firstXseries@{#1}%
4188 }%
4189 {}%
4190 }%
4191 }%
4192 }
4193 %

```

XIX Endnotes

First, check the `noend` option.

```

4194 \ifbool{noend@}{%Used instead of \ifnoend@ to prevent expansion problem
4195 %

```

XIX.1 Internal commands

`\l@dend@open` and `\l@dend@close` are the macros that are used to open and close the endnote file. Note that all our writing to this file is `\immediate`: all page and line numbers for the endnotes are generated by the same mechanism we use for the footnotes, so that there is no need to defer any writing to catch information from the output routine. The argument of these two command is the series letter.

```

4196 \newcommand{\l@dend@open}[1]{%
4197   \global\booltrue{l@dend@#1}%
4198   \expandafter\immediate%
4199   \expandafter\openout%
4200   \csname l@d@#1end\endcsname%
4201   =\l@auxdir\jobname.#1end\relax%
4202 }%
4203 \newcommand{\l@dend@close}[1]{%
4204   \global\boolfalse{l@dend@#1}%
4205   \expandafter\immediate%
4206   \expandafter\closeout\csname l@d@#1end\endcsname%
4207 }%
4208
4209 %

```

\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.

```

4210 \newcommand{\l@dend@stuff}{%
4211   \def\do##1{%
4212     \ifbool{l@dend@##1}{}%
4213     {\l@dend@open{##1}}%
4214     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}}%
4215   }%
4216   \dolistloop{@series}%
4217 }%
4218
4219 %

```

\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.

```

4220 \global\newbool{parapparatus@}{\long}\def\endprint#1#2#3#4#5#6#7{
4221 \csuse{Xendbhooknote@#4}%
4222 \csuse{Xendnotefontsize@#4}%
4223 \hangindent=\csuse{Xendhangindent@#4}%
4224 \ifXendinsertsep%
4225 \hskip\csuse{Xendafternote@#4}\relax%
4226 \csuse{Xendsep@#4}%
4227 \else%
4228 \iftoggle{Xendparagraph@#4}%
4229 {\global\Xendinsertsep@true}%
4230 {}%
4231 \fi%
4232 \xdef\@currentseries{#4}%
4233 \def\do##1{%
4234 \setkeys[mac]{truefootnoteoption}{##1}%
4235 }%
4236 \notblank{#5}{\docsvlist{#5}}{}%
4237 \IfStrEq{#6}{R}{\ledRcol@true}{}%
4238 \def\@this@crossref@start{#7:start}%
4239 \def\@this@crossref@end{#7:end}%
4240 \printlineendnote{#1}{#4}%
4241 \IfStrEq{#6}{R}{\ledRcol@false}{}%
4242 \undef\@this@crossref@start%
4243 \undef\@this@crossref@end%
4244 \nottoggle{Xendlemmadisablefontselection@#4}%
4245 {\select@lemmafont#1}%
4246 {}%
4247 \bgroup%
4248 \csuse{Xendlemmafont@#4}%
4249 \csuse{Xendwraplemma@#4}{#2}%
4250 \egroup%
4251 \ifbool{expr}%
4252 togl {nosep@}%
4253 or test{\ifcempty{Xendlemmaseparator@#4}}%
4254 }%
4255 {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
4256 {\nobreak%
4257 \hskip\csuse{Xendbeforelemmaseparator@#4}%
4258 \csuse{Xendlemmaseparator@#4}%
4259 \hskip\csuse{Xendafterlemmaseparator@#4}%
4260 \relax%
4261 }%
4262 \csuse{Xendwrapcontent@#4}{#3}%
4263 \nottoggle{Xendparagraph@#4}{\par}{}%
4264 \def\do##1{%
4265 \setkeys[mac]{falsefootnoteoption}{##1}%
4266 }%
4267 \notblank{#5}{\docsvlist{#5}}{}%
4268 }}%
4269

```



```

4310     {\hbox to \csuse{Xendboxsymlinenum@#1}%
4311       {\csuse{Xendsymlinenum@#1}\hfill}%
4312     }%
4313     \hspace{\csuse{Xendaftersymlinenum@#1}}%
4314   }%
4315   %

```

\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.)

```

4316 \newcommand{\printlineendnotearea}[2]{%
4317   \csuse{Xendbhooklinenumber@#2}%
4318   \hspace{\csuse{Xendbeforenumber@#2}}%
4319   \bgroup%
4320     \csuse{Xendnotenumfont@#2}%
4321     \ifdimequal{\csuse{Xendboxlinenum@#2}}{0pt}%
4322       {\printendlines#1||\ifledRcol@\@Rlineflag\fi}%
4323       {\leavevmode%
4324         \hbox to \csuse{Xendboxlinenum@#2}%
4325         {%
4326           \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4327           \printendlines#1||\ifledRcol@\@Rlineflag\fi%
4328           \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4329         }}%
4330   \egroup%
4331   \hspace{\csuse{Xendafternumber@#2}}%
4332   \csuse{Xendahooklinenumber@#2}%
4333 }%
4334 %

```

XIX.2 User level commands

XIX.2.1 Inserting contents to endnotes

The `\Xendnotes` commands are defined upper, 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*35 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes*36   \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4337     \def\do##1{%
4338       \reledmac@warning{toto}\expandafter\immediate\expandafter\write\
csname l@d@##1end\endcsname%
4339       {\unexpanded{#2}\@percentchar}%
4340     }%
4341     \ifstrempy{#1}%

```

```

4342     {\dolistloop{\@series}}}%
4343     {\docsvlist{#1}}}%
4344   }{\led@err@toendnotes@outsidenumbering}%
4345 }%
4346 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
4347   \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4348     \def\do##1{%
4349       \reledmac@warning{toto}\expandafter\immediate\expandafter\write\
csname l@d@##1end\endcsname%
4350       {#2\@percentchar}%
4351     }%
4352     \ifstrempy{#1}%
4353       {\dolistloop{\@series}}}%
4354       {\docsvlist{#1}}}%
4355   }{\led@err@toendnotes@outsidenumbering}%
4356 }%
4357 %

```

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@** is set to true at the first note of the series, and to false at the last one.

```

4358 \newif\ifXendinsertsep@
4359 \newcommand*{\doendnotes}[1]{%
4360   \l@dend@close{#1}%
4361   \begingroup
4362     \makeatletter
4363     \expandafter\let\csname #1end\endcsname=\endprint
4364     \input\l@auxdir\jobname.#1end%
4365     \global\Xendinsertsep@false%
4366   \endgroup}
4367 %

```

\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.

```

4368 \newcommand*{\doendnotesbysection}[1]{%
4369   \l@dend@close{#1}%
4370   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
4371   \begingroup%
4372     \makeatletter%
4373     \def\l@d@section##1{%
4374       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
4375       {\cslet{#1end}{\endprint}}}%

```

```

4376     {\cslet{#1end}{\@gobblefive}}}%
4377   }%
4378   \input\l@auxdir\jobname.#1end%
4379   \global\Xendinsertsep@false%
4380 \endgroup%
4381 }%
4382 %

```

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.

```

4383 }%
4384 %

```

\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.

```

4385 \newcommand*{\setprintendlines}[6]{%
4386   \l@d@pnumfalse \l@d@dashfalse
4387   \ifnum#4=#1 \else
4388     \l@d@pnumtrue
4389     \l@d@dashtrue
4390   \fi
4391 %

```

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.

```

4392   \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
4393   \ifnum#2=#5 \else
4394     \l@d@elintrue
4395     \l@d@dashtrue
4396   \fi
4397 %

```

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

```

4398   \l@d@ssubfalse
4399   \ifnum#3=0 \else
4400     \l@d@ssubtrue
4401   \fi
4402 %

```

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.

```

4403 \l@d@eslfalse
4404 \ifnum#6=0 \else
4405     \ifnum#6=#3
4406         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4407     \else
4408         \l@d@esltrue
4409         \l@d@dashtrue
4410     \fi
4411 \fi%
4412 %

4413 \ifl@d@dash%
4414     \ifboolexpr{togl{fulllines@} or test{\ifcsempy{Xendtwolines@
4415         \currentseries}}}%
4416     {%
4417     {%
4418         \setistwofollowinglines{#1}{#2}{#4}{#5}%
4419         \ifboolexpr{%
4420             (%
4421                 togl {Xendtwolinesbutnotmore@\currentseries}%
4422                 and not%
4423                 (%
4424                     bool {istwofollowinglines@}%
4425                 )%
4426             or%
4427             (%
4428                 (not test{\ifnumequal{#1}{#4}})%
4429                 and togl{Xendtwolinesonlyinsamepage@\currentseries}%
4430             )%
4431         }%
4432     }%
4433     {%
4434         \l@d@dashfalse%
4435         \l@d@Xtwolinestrue%
4436         \l@d@elinfalse%
4437         \l@d@eslfalse%
4438         \ifcsempy{Xendmorethantwolines@\currentseries}%
4439         {%
4440             {\ifistwofollowinglines@\else%
4441                 \l@d@Xmorethantwolinestrue%
4442             }%
4443         }%
4444     }%
4445 }%
4446 \fi%
4447 %

End of \setprintendlines.

4448 }%
```

```
4449 %
```

`\printendlines` Now we are ready to print it all.

```
4450 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{\begingroup
4451 \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
4452 %
```

The only subtlety left here is when to print a period between numbers. But the only instance in which this is tricky is for the ending sub-line number: it could be coming after the starting sub-line number (in which case we want only the dash) or after an ending line number (in which case we need to insert a period).

So, first, print the start lines.

```
4453 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
4454 {\bgroup}%
4455 {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}%
4456 \wrap@edcrossref{\@this@crossref@start}{\printnnum{#1}}%
4457 \ifl@d@dash%
4458 \ifl@d@pnum%
4459 \csuse{Xendlineprefixsingle@\@currentseries}%
4460 \else%
4461 \ifcempty{Xendlineprefixmore@\@currentseries}%
4462 {\csuse{Xendlineprefixsingle@\@currentseries}}
4463 {\csuse{Xendlineprefixmore@\@currentseries}}%
4464 \fi%
4465 \else%
4466 \csuse{Xendlineprefixsingle@\@currentseries}%
4467 \fi%
4468 \wrap@edcrossref{\@this@crossref@start}{\linenumrep{#2}}%
4469 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
4470 \ifl@d@ssub%
4471 \csuse{Xendsublinesep@\@currentseries}%
4472 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
4473 \fi%
4474 \egroup%
4475 %
```

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

```
4476 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
4477 {\bgroup}%
4478 {\hbox to \csuse{Xendboxendlinenum@\@currentseries}\bgroup}%
4479 \ifl@d@Xtwolines%
4480 \ifl@d@Xmorethantwolines%
4481 \csuse{Xendmorethantwolines@\@currentseries}%
4482 \else%
4483 \csuse{Xendtwolines@\@currentseries}%
4484 \fi%
4485 \else%
```

```

4486 \ifl@d@dash%
4487 \ifdefined\linangesep@%
4488 \linangesep@%
4489 \else%
4490 \csuse{Xendlinangeseparator@ \@currentseries}%
4491 \fi%
4492 \fi%
4493 \ifl@d@pnum%
4494 \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
4495 \fi%
4496 \ifl@d@elin%
4497 \ifl@d@pnum\csuse{Xendlineprefixsingle@ \@currentseries}\fi%
4498 \wrap@edcrossref{\@this@crossref@end}{\linenumrep{#5}}%
4499 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi}{}%
4500 \fi%
4501 \ifl@d@esl%
4502 \ifl@d@elin%
4503 \csuse{Xendsublinesep@ \@currentseries}%
4504 \fi%
4505 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
4506 \fi%
4507 \fi%
4508 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
4509 {}%
4510 {\hfill}%Prevent underfull hbox
4511 \egroup%
4512 \endgroup%
4513 }%
4514
4515 %

```

\printnpnum A macro to print a page number in an endnote. Should not be override anymore

```

4516 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@ \@currentseries}
4517 #1\csuse{Xendafterpagenumber@ \@currentseries}}
4518 %

```

XX Generate series of notes

In this section, X means the name of the series (A, B etc.)

\series \series\series creates one more new series. It is a public command, which just loops on the private command \newseries@.

```

4519 \newcommand{\newseries}[1]{%
4520 \def\do##1{\newseries@{##1}}%
4521 \docsvlist{#1}
4522 }
4523 %

```

\@series The \@series@ macro is an etoolbox list, which contains the name of all series.

```
4524 \newcommand{\@series}{}
4525 %
```

The command \newseries@series creates a new series of the footnote.

```
\newseries@26 \newcommand{\newseries@}[1]{
4527 %
```

XX.1 Test if series is still existing

```
4528 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
4529 {%
4530 %
```

XX.2 Init specific to reledpar

When calling \newseries@ after having loaded reledpar, we need to load specific setting.

```
4531 \ifdefined\newseries@par%
4532 \newseries@par{#1}%
4533 \fi%
4534 %
```

XX.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the \nocritical option of reledmac.

```
4535 \unless\ifnocritical@
4536 %
```

XX.3.1 Options

```
4537 \newtoggle{Xlineflag@#1}
4538 \newtoggle{Xparindent@#1}
4539 \newtoggle{Xlemmadisablefontselection@#1}
4540 \csgdef{Xwrapcontent@#1}{}%
4541 \csgdef{Xbeforeinserting@#1}{}%
4542 \csgdef{Xhangindent@#1}{Opt}%
4543 \csgdef{Xragged@#1}{}%
4544 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
4545 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
4546 \csgdef{Xcolalign@#1}{\raggedright}%
4547 \csgdef{Xnotenumberfont@#1}{\normalfont}%
4548 \csgdef{Xnotefontsize@#1}{\footnotesize}%
```

```

4549 \csgdef{Xbhooknote@#1}{}%
4550 \csgdef{Xbhookgroup@#1}{}%
4551
4552 \csgdef{Xboxlinenum@#1}{Opt}%
4553 \csgdef{Xboxlinenumalign@#1}{L}%
4554
4555 \csgdef{Xboxstartlinenum@#1}{Opt}%
4556 \csgdef{Xboxendlinenum@#1}{Opt}%
4557
4558 \csgdef{Xboxsymlinenum@#1}{Opt}%
4559 \newtoggle{Xnumberonlyfirstinline@#1}%
4560 \newtoggle{Xnumberonlyfirstintwolines@#1}%
4561 \csgdef{Xtwolines@#1}{}%
4562 \csgdef{Xmorethantwolines@#1}{}%
4563 \csgdef{Xsublinesep@#1}{\fullstop}%
4564 \newtoggle{Xtwolinesbutnotmore@#1}%
4565 \newtoggle{Xtwolinesonlyinsamepage@#1}%
4566 \newtoggle{Xonlypstart@#1}%
4567 \newtoggle{Xpstarteverytime@#1}%
4568 \newtoggle{Xpstart@#1}%
4569 \newtoggle{Xstanza@#1}%
4570 \csgdef{Xstanzaseparator@#1}{}%
4571 \csgdef{Xsymlinenum@#1}{}%
4572 \newtoggle{Xnonumber@#1}%
4573 \csgdef{Xbeforenumber@#1}{Opt}%
4574 \csgdef{Xtxtbeforenumber@#1}{}%
4575 \csgdef{Xafternumber@#1}{0.5em}%
4576 \newtoggle{Xnonbreakableafternumber@#1}%
4577 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
4578 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
4579 \csgdef{Xinplaceofnumber@#1}{1em}%
4580 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
4581 \csgdef{Xbeforelemmaseparator@#1}{0em}%
4582 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
4583 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
4584 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
4585 \csgdef{Xafterrule@#1}{Opt}%
4586
4587 \csgdef{Xtxtbeforenotes@#1}{%
4588 \newtoggle{Xtxtbeforenotes@#1@typeset}}%Not directly used by user,
but internal
4589
4590 \csgdef{Xmaxhnotes@#1}{0.8\vsizel}%
4591 \newtoggle{Xnoteswidthliketwocolumns@#1}%
4592 \csgdef{Xparafootsep@#1}{}%
4593 \csgdef{Xafternote@#1}{1em plus .4em minus .4em}%
4594 \csgdef{Xlinerrangeseparator@#1}{\endashchar}%
4595
4596 \csgdef{Xlemmafont@#1}{}%
4597 \csgdef{Xwraplemma@#1}{%

```

```

4598 \csgdef{Xwidth@#1}{\hsize}%
4599 %

```

XX.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

4600 \expandafter\newinsert\csname #1footins\endcsname%
4601 \unless\ifnoledgroup%
4602 \expandafter\newinsert\csname mp#1footins\endcsname%
4603 \fi%
4604 %

```

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.

```

4605 \global\newbool{parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname[2][ ]{%
4606 \if@edtext@secondarg%
4607 \ifledRcol%
4608 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}}%
4609 \else%
4610 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}}%
4611 \fi%
4612 \begingroup%
4613 \newcommand{\content}{##2}%
4614 \ifnumberedpar%
4615 \ifledRcol%
4616 \ifluatex%
4617 \footnotelang@lua[R]%
4618 \fi%
4619 \@ifundefined{xpg@main@language}%if polyglossia
4620 {}%
4621 {\footnotelang@poly[R]}%
4622 \footnoteoptions@{R}{##1}{true}%
4623 \xright@appenditem{%
4624 \ifbool{indtl@innote}%
4625 {\unexpanded{\let\index\nindex}}%
4626 {}%
4627 \ifbool{indtl@notenumber}%
4628 {\unexpanded{\let\index\nindex}}%There is no note
...number so
4629 {}%
4630 \noexpand\Xnote@true%
4631 \noexpand\prepare@Xprenotes{#1}%
4632 \noexpand\prepare@edindex@fornote{\l@d@nums}%

```

```

4633 \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.
4634 \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
4635 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
4636 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
4637 \noexpand\csuse{v#1footnote}{#1}%
4638 {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
4639 \noexpand\Xnote@false%
4640 \unexpanded{\undef\@this@crossref@start}%
4641 \unexpanded{\undef\@this@crossref@end}%
4642 \ifbool{indtl@innote}%
4643 {\unexpanded{\let\index\orig@@index}}%
4644 {}%
4645 \ifbool{indtl@notenumber}%
4646 {\unexpanded{\let\index\orig@@index}}%
4647 {}%
4648 }\to\inserts@listR
4649 \footnoteoptions@{R}{#1}{false}%
4650 \global\advance\insert@countR \@ne%
4651 \else%
4652 \ifluatex%
4653 \footnotelang@lua%
4654 \fi%
4655 \@ifundefined{xpg@main@language}%if polyglossia
4656 {}%
4657 {\footnotelang@poly}%
4658 \footnoteoptions@{L}{#1}{true}%
4659 \xright@appenditem{%
4660 \ifbool{indtl@innote}%
4661 {\unexpanded{\let\index\nindex}}%
4662 {}%
4663 \ifbool{indtl@notenumber}%
4664 {\unexpanded{\let\index\nindex}}%There is no note
...number so
4665 {}%
4666 \noexpand\Xnote@true%
4667 \noexpand\prepare@Xprenotes{#1}%
4668 \noexpand\prepare@edindex@fornote{\l@d@nums}%
4669 \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.
4670 \ifl@dpairing%
4671 \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote

```

Create counter used to determiner in which page was called the previous note.

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.

 $\backslash fi$

XX.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```
4710 \unless\ifnofamiliar@
4711 %
```

XX.4.1 Options

```
4712 \newtoggle{parindentX@#1}
4713 \csgdef{wrapcontentX@#1}{}%
4714 \csgdef{hangindentX@#1}{Opt}%
4715 \csgdef{beforeinsertingX@#1}{}%
4716 \csgdef{raggedX@#1}{}%
4717 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
4718 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
4719 \csgdef{colalignX@#1}{\raggedright}%
4720 \csgdef{notenumfontX@#1}{\normalfont}%
4721 \csgdef{notefontsizeX@#1}{\footnotesize}%
4722 \csgdef{bhooknoteX@#1}{}%
4723 \csgdef{bhookgroupX@#1}{}%
4724 \csgdef{afterruleX@#1}{Opt}
4725 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
4726 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
4727 \newtoggle{noteswidthliketwocolumnsX@#1}%
4728 \csgdef{parafootsepX@#1}{}%
4729 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
4730 \csgdef{widthX@#1}{\hsizex}%
4731 % End of for familiar footnotes.
4732 % \subsubsection{Create inserts, needed to add notes in foot}
4733 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
4734 % \begin{macrocode}
4735 \expandafter\newinsert\csname footins#1\endcsname%
4736 \unless\ifnoledgroup%
4737 \expandafter\newinsert\csname mpfootins#1\endcsname%
4738 \fi%
4739 %
```

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.

```
4740 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
4741 \begingroup%
4742 \prepare@prenotesX{#1}%
4743 \newcommand{\content}{##1}%
4744 %
4745 %
```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.2.1 p. 46).

```

4746         \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
4747         \ifboolexpr{bool{!l@dpairing} or bool{!l@dprintingpages} or
bool{!l@dprintingcolumns}}{%
4748         \ifcsdef{footnote#1@reading\the\csname footnote#1@reading\
endcsname=typeset}%
4749         {\setcounter{footnote#1}{\csuse{footnote#1@reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
4750         {\setcounter{footnote#1}{\the\csname footnote#1@reading
\endcsname}}}%
4751     }{%
4752     \stepcounter{footnote#1}%
4753     }%
4754 %

```

We also have to check consistency with `\onlysideX` setting.

```

4755         \ifledRcol%
4756         \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}{%
4757         \else%
4758         \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}{%
4759         \fi%
4760 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

4761         \protected\csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
4762         \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
4763         {\csuse{@footnotemark#1}}%
4764         {}%
4765     \ifluatex%
4766     \xdef\footnote@luatextextdir{\the\textdir}%
4767     \xdef\footnote@luatexpardir{\the\pardir}%
4768     \fi%
4769     \if@ledgroup%
4770     \led@set@index@fornote{#1}%
4771     \fi%
4772     \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
4773     \ifbool{indtl@innote}%
4774     {\let\index\orig@@index}%
4775     {}%
4776     \ifbool{indtl@notenumber}%
4777     {\let\index\orig@@index}%
4778     {}%
4779     \endgroup%
4780 }

```

```
4781 %
```

Then define the counters. The \LaTeX counter `footnoteX` is the only one manipulated by the user. This is this the one which is printed. The \TeX counter `\footnoteX@reading` is increased at each footnote. It is used for hyperlinks, for using `hyperlink` package, and for getting the correct footnote number when using parallel typesetting (V.2.1 p. 46).

```
4782 \newcounter{footnote#1}
4783 \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
4784 \expandafter\newcount\csname footnote#1@reading\endcsname%
4785 %
```

Create counter used to determiner in which page was called the previous note.

```
4786 \expandafter\newcount\csname prevpage#1@num\endcsname%
4787 \expandafter\newcount\csname prevpage#1@numR\endcsname%
4788 %
```

Do not forget to initialize series

```
4789 \arrangementX@normal{#1}%
4790 \fi
4791 %
```

XX.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```
4792 \unless\ifnoend@
4793 %
```

XX.5.1 The auxiliary file

`\l@d@Xend` Endnotes of all varieties are saved up in a file, one by series, typically named `\jobname.Xend`.
`\ifl@dend@X` `\l@d@end` is the output stream number for this file, and `\ifl@dend@X` is a flag that is
`\l@dend@Xtrue` true when the file is open.
`\l@dend@Xfalse`

```
4794 \expandafter\newwrite\csname l@d@#1end\endcsname%
4795 \expandafter\newif\csname ifl@dend@#1\endcsname%
4796 %
```

XX.5.2 The main macro

The `\Xendnote` macro functions to write one endnote to the `.Xend` file. We change `\newlinechar` so that in the file every space becomes the start of a new line; this generally ensures that a long note does not exceed restrictions on the length of lines in files.

```

4797 \global\expandafter\newcommand\csname #1endnote\endcsname[2][1,
4798 usedefault]{%
4799     \bgroup%
4800     \newlinechar='40%
4801     \global\@noneed@Footnotetrue%
4802     \newcommand{\content}{##2}%
4803     \stepcounter{labidx}%
4804     \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
4805     \unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\
expandafter{\sw@inthisedtext}} %Voluntary space, to add linebreak in the
output file
4806     \expandafter\string\csname #1end\endcsname%
4807     {\ifnumberedpar@l@d@nums\fi}%
4808     {\ifnumberedpar@\expandonce\@tag\fi}%
4809     {\expandonce\content}%
4810     {\#1}%
4811     {\unexpanded{##1}}%
4812     {\ifledRcol R\else L\fi}%
4813     {\theedtext}%
4814     \@percentchar%
4815     }%
4816     \egroup%
4817     \ignorespaces%
4818 }%
4819 %

```

XX.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoennotes`

```

4820 \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
4821 \toendnotes[#1]{##1}%
4822 }%
4823
4824 \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
endcsname*[1]{%
4825 \toendnotes*[#1]{##1}%
4826 }%
4827
4828 %
4829 %

```

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 `\@gobblefive`, which just skips over its five arguments.

```
4830
4831 \global\cslet{#1end}{\@gobblefive}
4832 %
```

We need to store the number of times `\doendnotesbysection` is called for one series.

```
4833 \global\expandafter\newcount\csname #1end@bysection\endcsname%
4834 %
```

XX.5.5 The options

```
4835 \csgdef{Xendwraplemma@#1}{%
4836 \csgdef{Xendwrapcontent@#1}{}%
4837 \csgdef{Xendtwolines@#1}{}%
4838 \csgdef{Xendmorethantwolines@#1}{}%
4839 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
4840 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
4841 \newtoggle{Xendlemmadisablefontselection@#1}{}%
4842 \csgdef{Xendnotenumfont@#1}{\normalfont}%
4843 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
4844 \csgdef{Xendbhooknote@#1}{}%
4845
4846 \csgdef{Xendsublinesep@#1}{\fullstop}%
4847
4848 \csgdef{Xendbeforenumber@#1}{Opt}
4849 \csgdef{Xendafternumber@#1}{0.5em}
4850
4851 \csgdef{Xendboxlinenum@#1}{Opt}%
4852 \csgdef{Xendboxlinenumalign@#1}{L}%
4853
4854 \csgdef{Xendboxstartlinenum@#1}{Opt}%
4855 \csgdef{Xendboxendlinenum@#1}{Opt}%
4856
4857 \csgdef{Xendlemmaseparator@#1}{}%
4858 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
4859 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
4860 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
4861
4862 \newtoggle{Xendparagraph@#1}%
4863 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
4864 \csgdef{Xendsep@#1}{}%
4865
4866 \csgdef{Xendinplaceofnumber@#1}{Opt}%
4867 \newtoggle{Xendnonumber@#1}%
4868
4869 \csgdef{Xendhangindent@#1}{Opt}%
```

```

4870 \newtoggle{Xendnumberonlyfirstinline@#1}%
4871 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
4872
4873 \csgdef{Xendbeforesymmlinenumber@#1}{\csuse{Xendbeforenumber@#1}}%
4874 \csgdef{Xendaftersymmlinenumber@#1}{\csuse{Xendafternumber@#1}}%
4875 \csgdef{Xendsymmlinenumber@#1}{}%
4876 \csgdef{Xendboxsymmlinenumber@#1}{0pt}%
4877
4878 \csgdef{Xendbhooklinenumber@#1}{}%
4879 \csgdef{Xendehooklinenumber@#1}{}%
4880 \csgdef{Xendbhookinplaceofnumber@#1}{}%
4881 \csgdef{Xendehookinplaceofnumber@#1}{}%
4882
4883 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%
4884
4885 \csgdef{Xendbeforepagenumber@#1}{p.}%
4886 \csgdef{Xendafterpagenumber@#1}{ }%
4887 \csgdef{Xendlineprefixsingle@#1}{}%
4888 \csgdef{Xendlineprefixmore@#1}{}%
4889
4890 \newtoggle{Xendlineflag@#1}
4891
4892 \csgdef{Xendlemmafont@#1}{}%
4893 %

```

End of endnotes declaration

```

4894 \fi%
4895 %

```

Dump series in \@series

```

4896 \listxadd{\@series}{#1}
4897 }
4898 }% End of \newseries
4899 %

```

XX.6 Init standards series (A,B,C,D,E)

```

4900 \expandafter\newseries\expandafter{\default@series}
4901 %

```

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.

```

4902 \newcommand{\seriesatbegin}[1]{%
4903   \StrDel{\@series}{#1}[\@series]%

```

```

4904 \edef\@new{}%
4905 \listadd{\@new}{#1}%
4906 \listadd{\@new}{\@series}%
4907 \xdef\@series{\@new}%
4908 }
4909 %

```

\seriesatend And \seriesatend moves the series to the end of the list.

```

4910 \newcommand{\seriesatend}[1]{%
4911   \StrDel{\@series}{#1}[\@series]%
4912   \edef\@new{}%
4913   \listadd{\@new}{\@series}%
4914   \listadd{\@new}{#1}%
4915   \xdef\@series{\@new}%
4916 }
4917 %

```

XXI.2 Test series order

\ifseriesbefore \ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>} expands <true> if <seriesA> is printed before <seriesB>, expands <false> otherwise.

```

4918 \newcommand{\ifseriesbefore}[4]{%
4919   \StrPosition{\@series}{#1}[\@first]%
4920   \StrPosition{\@series}{#2}[\@second]%
4921   \ifnumgreater{\@second}{\@first}{#3}{#4}%
4922 }
4923 %

```

XXI.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries24 \newcommand{\@getfirstseries}{%
4925   \ifdefempty{\@series}%
4926   {\xdef\@firstseries{}}%
4927   {\StrChar{\@series}{1}[\@firstseries]}%
4928 }%
4929 %

```

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`.

```

4930 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
4931   \def\do##1{%
4932     \global\settoggle{#2@##1}{#3}%
4933     \ifstrequal{#4}{critical}{
4934       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
4935     }{}
4936     \ifstrequal{#4}{familiar}{
4937       \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
4938     }{}
4939   }%
4940   \ifstreempty{#1}{%
4941     \dolistloop{\@series}%
4942     \ifstreempty{#5}{}%
4943       \docsvlist{#5}%
4944   }
4945 }%
```

```

4946         {%
4947         \docsvlist{#1}%
4948         }%
4949     }
4950 %

```

`\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.

```

4951 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
4952     \def\do##1{
4953         \csgdef{#2@##1}{#3}
4954         \ifstrequal{#4}{critical}{%
4955             \csuse{Xarrangement@\csuse{series@display##1}}{##1}%
4956         }{}
4957         \ifstrequal{#4}{familiar}{%
4958             \csuse{arrangementX@\csuse{series@displayX##1}}{##1}%
4959         }{}%
4960     }%
4961     \ifstreempty{#1}{%
4962         \dolistloop{\@series}%
4963         \ifstreempty{#5}{}{%
4964             \docsvlist{#5}
4965         }
4966     }%
4967     {%
4968         \docsvlist{#1}%
4969     }%
4970 }%
4971 %

```

XXI.3.3 Tools to generate options commands

`\newhookcommand@series` `\newhookcommand@series\command` names is a generic command to add new commands for hooks, like `\Xhsizetwocol`. The first argument is the name of the hook, the second a comma-separated list of pseudo-series where the hook can be used, like

appref in the case of `\Xtwolines`. The second argument is also used to create commands named `\<hookname><pseudoserries>`, like `\Xtwolinesappref`.

```

4972 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
4973   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
4974     \setcommand@series{##1}{#1}{##2}[][#2]%
4975   }%
4976   \ifstrempy{#2}{}{%
4977     \def\do##1{%
4978       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
4979       [1]{%
4980         \csuse{#1}[##1]{####1}%
4981       }%
4982     }%
4983   \docsvlist{#2}%
4984 }%
4985 %

```

`\newhooktoggle@series` `\newhooktoggle@series\command names` is a generic command to add new commands for a new toggle hook, like `\Xnumberonlyfirstinline`. The second argument is also used to create commands named `\<hookname><pseudoserries>`, like `\Xtwolinesbutnotmoreappref`.

```

4986 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
4987   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
4988     true},usedefault]{%
4989     \settoggle@series{##1}{#1}{##2}[][#2]%
4990   }%
4991   \ifstrempy{#2}{}{%
4992     \def\do##1{%
4993       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
4994         \csuse{#1}[##1]%
4995       }%
4996     }%
4997   \docsvlist{#2}%
4998 }%
4999 %

```

`\newhookcommand@toggle@reload` `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement, depending of type or notes

```

5000 \newcommand{\newhooktoggle@series@reload}[2]{%
5001   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
5002     true},usedefault]{%
5003     \settoggle@series{##1}{#1}{##2}[#2]%
5004   }%
5005 %

```

`\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.

```

5006 \newcommand{\newhookcommand@series@reload}[2]{%
5007   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][\{%
5008     \setcommand@series{##1}{#1}{##2}[#2]%
5009   }%
5010 }
5011 %

```

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.

```

5012 \unless\ifnocritical@
5013   \newhookcommand@series{Xwrapcontent}%
5014   \newhookcommand@series{Xbeforeinserting}%
5015   \newhookcommand@series{Xlemmafont}%
5016   \newhookcommand@series{Xwraplemma}%
5017   \newhooktoggle@series{Xparindent}
5018   \newhookcommand@series{Xhangindent}
5019   \newhookcommand@series{Xragged}
5020   \newhookcommand@series{Xhsizetwocol}
5021   \newhookcommand@series{Xhsizethreecol}
5022   \newhookcommand@series{Xcolalign}%
5023   \newhookcommand@series{Xnotenumfont}
5024   \newhookcommand@series{Xbhooknote}
5025   \newhookcommand@series@reload{Xbhookgroup}{critical}
5026   \newhookcommand@series{Xboxsymlinenum}%
5027   \newhookcommand@series{Xsymlinenum}
5028   \newhookcommand@series{Xbeforenumber}
5029   \newhookcommand@series{Xtxtbeforenumber}
5030   \newhookcommand@series{Xafternumber}
5031   \newhookcommand@series{Xbeforesymlinenum}
5032   \newhookcommand@series{Xaftersymlinenum}
5033   \newhookcommand@series{Xinplaceofnumber}
5034   \newhookcommand@series{Xlemmaseparator}
5035   \newhookcommand@series{Xbeforelemmaseparator}
5036   \newhookcommand@series{Xafterlemmaseparator}
5037   \newhookcommand@series{Xinplaceoflemmaseparator}
5038   \newhookcommand@series{Xtxtbeforenotes}
5039   \newhookcommand@series@reload{Xafterrule}{critical}
5040   \newhooktoggle@series{Xnumberonlyfirstinline}
5041   \newhooktoggle@series{Xnumberonlyfirstintwolines}
5042   \newhooktoggle@series{Xnonumber}
5043   \newhooktoggle@series{Xpstart}
5044   \newhooktoggle@series{Xpstarteverytime}%
5045

```

```

5046 \newhooktoggle@series{Xstanza}%
5047 \newhookcommand@series{Xstanzaseparator}%
5048
5049 \newhooktoggle@series{Xonlypstart}
5050 \newhooktoggle@series{Xnonbreakableafternumber}
5051 \newhooktoggle@series{Xlemmadisablefontselection}
5052 \newhookcommand@series@reload{Xmaxhnotes}{critical}
5053 \newhookcommand@series@reload{Xbeforenotes}{critical}
5054 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
5055 \newhookcommand@series@reload{Xnotefontsize}{critical}
5056
5057 \newhookcommand@series{Xboxlinenum}%
5058 \newhookcommand@series{Xboxlinenumalign}%
5059
5060 \newhookcommand@series{Xboxstartlinenum}%
5061 \newhookcommand@series{Xboxendlinenum}%
5062
5063 \newhookcommand@series{Xafternote}%
5064 \newhookcommand@series{Xparafootsep}
5065
5066 \newhookcommand@series@reload{Xwidth}{critical}%
5067
5068 \ifundef{\Xhsize}%
5069   {%
5070     \newcommandx{\Xhsize}[2][1,usedefault]{%
5071       \led@warning@Xhsize@deprecated%
5072       \Xwidth[#1]{#2}%
5073     }%
5074   }%
5075   {}%
5076 \fi
5077 \newhooktoggle@series{Xlineflag}[appref,SEref]
5078 \newhookcommand@series{Xtwolines}[appref,SEref]
5079 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
5080 \newhookcommand@series{Xsublinesep}[appref,SEref,side]
5081 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
5082 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
5083 \newhookcommand@series{Xlinrangeseparator}[appref,SEref]
5084 %

```

XXI.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

5085 \unless\ifnofamiliar@
5086   \newhookcommand@series{wrapcontentX}%
5087   \newhookcommand@series{beforeinsertingX}%
5088   \newhooktoggle@series{parindentX}
5089   \newhookcommand@series{hangindentX}

```

```

5090 \newhookcommand@series{raggedX}
5091 \newhookcommand@series{hsizewocolX}
5092 \newhookcommand@series{hsizethreecolX}
5093 \newhookcommand@series{colalignX}%
5094 \newhookcommand@series{notenumfontX}
5095 \newhookcommand@series{bhooknoteX}
5096 \newhookcommand@series@reload{bhookgroupX}{familiar}
5097 \newhookcommand@series@reload{beforenotesX}{familiar}
5098 \newhookcommand@series@reload{maxhnotesX}{familiar}
5099 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
5100 \newhookcommand@series@reload{afterruleX}{familiar}
5101 \newhookcommand@series@reload{notefontsizeX}{familiar}
5102 \newhookcommand@series{afternoteX}
5103 \newhookcommand@series{parafootsepX}
5104 \newhookcommand@series@reload{widthX}{familiar}%
5105 \ifundef{\hsizeX}%
5106   {%
5107     \newcommandx{\hsizeX}[2][1,usedefault]{%
5108       \led@warning@hsizeX@deprecated%
5109       \widthX[#1]{#2}%
5110     }%
5111   }%
5112   {}%
5113 \fi
5114 %

```

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.

```

5115 \unless\ifnoend@
5116 \newhookcommand@series{Xendwraplemma}
5117 \newhookcommand@series{Xendwrapcontent}
5118 \newhookcommand@series{Xendnotenumfont}
5119 \newhookcommand@series{Xendlemmafont}%
5120 \newhookcommand@series{Xendbhooknote}
5121
5122 \newhookcommand@series{Xendboxlinenum}%
5123 \newhookcommand@series{Xendboxlinenumalign}%
5124
5125 \newhookcommand@series{Xendboxstartlinenum}%
5126 \newhookcommand@series{Xendboxendlinenum}%
5127
5128 \newhookcommand@series{Xendnotefontsize}
5129 \newhooktoggle@series{Xendlemmadisablefontselection}
5130 \newhookcommand@series{Xendlemmaseparator}
5131 \newhookcommand@series{Xendbeforelemmaseparator}
5132 \newhookcommand@series{Xendafterlemmaseparator}
5133 \newhookcommand@series{Xendinplaceoflemmaseparator}

```

```

5134 \newhookcommand@series{Xendbeforenumber}%
5135 \newhookcommand@series{Xendafternumber}%
5136
5137
5138 \newhooktoggle@series{Xendparagraph}
5139 \newhookcommand@series{Xendafternote}
5140 \newhookcommand@series{Xendsep}
5141
5142 \newhookcommand@series{Xendinplaceofnumber}%
5143 \newhooktoggle@series{Xendnonumber}%
5144
5145 \newhooktoggle@series{Xendnumberonlyfirstinline}%
5146 \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
5147
5148 \newhookcommand@series{Xendsymmlinenum}%
5149 \newhookcommand@series{Xendbeforesymmlinenum}%
5150 \newhookcommand@series{Xendaftersymmlinenum}%
5151 \newhookcommand@series{Xendboxsymmlinenum}%
5152
5153 \newhookcommand@series{Xendbhooklinenumber}%
5154 \newhookcommand@series{Xendahooklinenumber}%
5155 \newhookcommand@series{Xendbhookinplaceofnumber}%
5156 \newhookcommand@series{Xendahookinplaceofnumber}%
5157
5158 \newhookcommand@series{Xendhangindent}%
5159
5160
5161 \fi
5162 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
5163 \newhookcommand@series{Xendtwolines}[apprefwithpage,SErefwithpage]
5164 \newhookcommand@series{Xendmorethantwolines}[apprefwithpage,SErefwithpage]
5165 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,SErefwithpage]
5166 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
SErefwithpage]
5167 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage
]
5168 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
SErefonlypage]
5169 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
5170 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
5171 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
5172 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
5173
5174 %

```

XXI.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

5175 \newcommand{\newhooktoggle@specific}[1]{%
5176   \newtoggle{#1}%
5177   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
    When enabling footnote option
5178   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false}
    }}
5179 }
5180 %

```

\newhookarg@specific \newhookarg@specific is a generic command to create argumen hook specific to a note.

```

5181 \newcommand{\newhookarg@specific}[1]{%
5182   \define@key[mac]{truefootnoteoption}{#1}{\global\def\linrangesep@{##1}}%
    When enabling footnote option
5183   \define@key[mac]{falsefootnoteoption}{#1}{\global\undef\linrangesep@}%
    When
5184 }
5185 %

```

And now, we define some hooks specific to a note.

```

5186 \newhooktoggle@specific{fulllines}%
5187 \newhooktoggle@specific{nonum}
5188 \newhooktoggle@specific{nosep}
5189 \newhookarg@specific{linrangesep}
5190 %

```

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.

```

5191 %

```

\nomk@ \nomk@ toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```

5192 \newtoggle{nomk@}%
5193 %

```

XXI.5 Alias

\Xnolemmaseparator \Xnolemmaseparator[*<series>*] is just an alias for `\Xlemmaseparator[<series>]{}`.

```

5194 \newcommandx*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]{}}
5195 %

```

XXII Output routine

Now we begin the output routine and associated things.

XXII.1 Page number management

`\pageno` `\advancepageno` `\pageno` is a page number, starting at 1, and `\advancepageno` increments the number.

```

5196 \countdef\pageno=0 \pageno=1
5197 \newcommand*\advancepageno{\ifnum\pageno<\z@ \global\advance\pageno\m@ne
5198   \else\global\advance\pageno\@ne\fi}
5199
5200 %

```

XXII.2 Extra footnotes output

With luck we might only have to change `\@makecol` and `\@reinserts` of the \TeX 's kernel. Since `reledmac`, we use `etoolbox`'s patching commands instead of overriding. It should provides better compatibility with other package which modify these commands

`\doextrafeet` `\doextrafeet` is the code extending `\@makecol` to cater for the extra `reledmac` feet. We have two categories of extra footnotes. By default, we order the footnote inserts so that the regular footnotes of \TeX are first, then familiar familiar footnotes and finally the critical footnotes.

```

5201 \newcommand*\l@ddoextrafeet{%
5202   \IfStrEq{familiar-critical}{\@fnpos}
5203     {\do@feetX\Xdo@feet}%
5204   {%
5205     \IfStrEq{critical-familiar}{\@fnpos}%
5206       {\Xdo@feet\do@feetX}%
5207       {\do@feetX\Xdo@feet}%
5208     }%
5209   }%
5210
5211 %

```

`\Xdo@feet` `\Xdo@feet` is the code extending `\@makecol` to cater for the extra critical feet.

```

5212 \newcommand*\Xdo@feet{%
5213   \setbox\@outputbox \vbox{%
5214     \unvbox\@outputbox
5215     \@opXfeet}}
5216 %

```

`\@opXfeet` `\print@Xnotes` The extra critical feet to be added to the output. The normal way to add one series, `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

5217 \newcommand\print@Xnotes[1]{%
5218   \xdef\@currentseries{#1}%
5219   \csuse{#1footstart}{#1}%
5220   \csuse{#1footgroup}{#1}%%
5221 }%
5222 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

5223 \newcommand*\@opXfeet{%
5224   \unless\ifnocritical@%
5225     \gdef\firstXseries@{}%
5226     \def\do##1{%
5227       \ifvoid\csuse{##1footins}\else%
5228         \global\skip\csuse{##1footins}=\csuse{Xbeforenotes@##1}%
5229         \global\advance\skip\csuse{##1footins} by\csuse{Xafterrule@##1}%
5230         \print@Xnotes{##1}%
5231       \fi%
5232     }%
5233     \dolistloop{\@series}%
5234   \fi%
5235 }%
5236 %

```

\l@ddodoreintrafeet \l@ddodoreintrafeet is the code for catering for the extra footnotes within \@reinserts. We use the same category and ordering as in \l@ddoxtrafeet.

```

5237 \newcommand*\l@ddodoreintrafeet{%
5238   \IfStrEq{familiar-critical}{\@fnpos}
5239   {\@doreinfeetX\X@doreinfeet}%
5240   {%
5241     \IfStrEq{critical-familiar}{\@fnpos}%
5242     {\X@doreinfeet\@doreinfeetX}%
5243     {\@doreinfeetX\X@doreinfeet}%
5244   }%
5245 }
5246
5247 %

```

\X@doreinfeet \X@doreinfeet is the code for catering for the extra critical footnotes within \@reinserts.

```

5248 \newcommand*\X@doreinfeet{%
5249   \unless\ifnocritical@%
5250     \def\do##1{%
5251       \ifvoid\csuse{##1footins}\else%
5252         \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
5253       \fi}%
5254     \dolistloop{\@series}
5255   \fi%
5256 }

```

```

5257
5258 %

\print@notesX We have to add all the new kinds of familiar footnotes to the output routine. The normal
\do@feetX way to add one series. \print@Xnotes is replaced by reledpar when using \Pages.
@doreinfeetX
5259 \newcommand\print@notesX[1]{%
5260 \xdef\@currentseries{#1}%
5261 \csuse{footstart#1}{#1}%
5262 \csuse{footgroup#1}{#1}%
5263 }%
5264 %

We print all the series of notes by looping on them. We check before printing them that
they are not voided.

5265 \newcommand*\do@feetX{%
5266 \unless\ifnofamiliar%
5267 \gdef\firstseriesX@{}%
5268 \setbox\@outputbox \vbox{%
5269 \unvbox\@outputbox%
5270 \def\do##1{%
5271 \ifvoid\csuse{footins##1}\else%
5272 \global\skip\csuse{footins##1}=\csuse{beforenotesX@##1}%
5273 \global\advance\skip\csuse{footins##1} by\csuse{afterterruleX@##1}%
5274 \print@notesX{##1}%
5275 \fi%
5276 }%
5277 \dolistloop{\@series}}%
5278 \fi%
5279 }%
5280
5281 \newcommand{\@doreinfeetX}{%
5282 \unless\ifnofamiliar%
5283 \def\do##1{%
5284 \ifvoid\csuse{footins##1}\else
5285 \insert%
5286 \csuse{footins##1}
5287 {\unvbox\csuse{footins##1}}%
5288 \fi%
5289 }%
5290 \dolistloop{\@series}%
5291 \fi%
5292 }%
5293
5294 %

```

XXII.3 Patching standard output of commands

The memoir class does not use the ‘standard’ versions of `\@makecol` and `\@reinserts`, due to its sidebar insert. We had better add that code if memoir is used. (It can be

awkward dealing with `\if` code within `\if` code, so don't use `\ifl@dmemoir` here.)

```

5295 \@ifclassloaded{memoir}{%
5296 %

memoir is loaded so we use memoir's built in hooks.

5297 \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
5298 \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinxtrafeet}%
5299 }{%
5300 %

memoir has not been loaded, so patch \@makecol and \@reinserts.

5301 \@ifpackageloaded{fancyhdr}{%
5302   \patchcmd%
5303     {\latex@makecol}%
5304     {\xdef\@freelist{\@freelist\@midlist}}}%
5305     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5306     {}%
5307     {\led@error@fail@patch@@makecol}%
5308   }{%
5309     \patchcmd%
5310       {\@makecol}%
5311       {\xdef\@freelist{\@freelist\@midlist}}}%
5312       {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5313       {}%
5314       {\led@error@fail@patch@@makecol}%
5315     }%
5316
5317   \patchcmd%
5318     {\@reinserts}%
5319     {\ifvbox}%
5320     {\l@ddodoreinxtrafeet\ifvbox}%
5321     {}%
5322     {\led@error@fail@patch@@reinserts}%
5323   }
5324
5325 %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

5326 \newif\if@led@nofoot
5327
5328 %

5329 \@ifclassloaded{memoir}{%
5330 %

```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet31 \g@addto@macro{\@mem@extranofeet}{%%
5332 \def\do#1{%
5333 \unless\ifnocritical@%
5334 \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
5335 \fi%
5336 \unless\ifnofamiliar@%
5337 \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
5338 \fi%
5339 }
5340 \dolistloop{\@series}%
5341 }%
5342 }{%
5343 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot44 \newcommand*{\@led@testifnofoot}{%
\@doclearpage45 \@led@nofoottrue%
5346 \ifvoid\footins\else%
5347 \@led@nofootfalse%
5348 \fi%
5349 \def\do##1{%
5350 \unless\ifnocritical@%
5351 \ifvoid\csuse{##1footins}\else%
5352 \@led@nofootfalse%
5353 \fi%
5354 \fi%
5355 \unless\ifnofamiliar@%
5356 \ifvoid\csuse{footins##1}\else%
5357 \@led@nofootfalse%
5358 \fi%
5359 \fi%
5360 }%
5361 \dolistloop{\@series}%
5362 }%
5363
5364 \pretocmd%
5365 {\@doclearpage}%
5366 {\@led@testifnofoot}%
5367 {}%
5368 {\led@error@fail@patch@@doclearpage}%
5369
5370 \patchcmd%
5371 {\@doclearpage}%
5372 {\ifvoid\footins}%
5373 {\if@led@nofoot}%
5374 {}%
5375 {\led@error@fail@patch@@doclearpage}%
5376

```

```
5377 }
5378
5379 %
```

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.

```
5380 \list@create{\labelref@list}
5381 %
```

\zz@@@ A convenience macro to zero two labeling counters in one go.

```
5382 \newcommand*{\zz@@@}{000|000} % set two counters to zero in one go
5383
5384 %
```

\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.³²

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 Billett³³ found that the original code could be off by several pages. This version, hopefully cures that, and also allows for non-arabic page numbering.

```
5385 \newcommand*{\edlabel}[1]{%
5386   \ifl@dpairing\ifautopar%
5387     \strut%
5388   \fi\fi%
5389   \@bsphack%
5390   \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
```

³²The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

³³(jdb43@cam.ac.uk) via the ctt thread ‘ledmac cross referencing’, 25 August 2003.

```

5391 \ifXnote@%
5392 \protected@write\@auxout{%
5393   {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\the\c@pstartR|{#1}}}%
5394 \ifdef{\hypertarget}%
5395   {\Hy@raisedlink{\hypertarget{#1}{}}}%
5396   {}%
5397 \else%
5398   \write\linenum@outR{\string\@lab}%
5399   \ifx\labelref@listR\empty%
5400     \xdef\label@refs{\zz@@@}%
5401   \else%
5402     \gl@p\labelref@listR\to\label@refs%
5403   \fi%
5404   \ifvmode%
5405     \advancelabel@refs%
5406   \fi%
5407 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

5408 \protected@write\@auxout{%
5409   {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}}%
5410 \ifdef{\hypertarget}%
5411   {\Hy@raisedlink{\hypertarget{#1}{}}}%
5412   {}%
5413 \fi%
5414 }{%
5415 \ifXnote@%
5416 \protected@write\@auxout{%
5417   {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\the\c@pstartR|{#1}}}%
5418 \ifdef{\hypertarget}%
5419   {\Hy@raisedlink{\hypertarget{#1}{}}}%
5420   {}%
5421 \else%
5422   \write\linenum@out{\string\@lab}%
5423   \ifx\labelref@list\empty%
5424     \xdef\label@refs{\zz@@@}%
5425   \else%
5426     \gl@p\labelref@list\to\label@refs%
5427   \fi%
5428   \ifvmode%
5429     \advancelabel@refs%
5430   \fi%
5431 \protected@write\@auxout{%
5432   {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstart
|{#1}}}%
5433 \ifdef{\hypertarget}%

```

```

5434         {\Hy@raisedlink{\hypertarget{#1}{}}}%
5435         }%
5436     \fi%
5437 }%
5438 \@esphack}%
5439
5440 %

```

`\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.

```

5441 \newcounter{line}%
5442 \newcounter{subline}%
5443 \newcommand{\advancelabel@refs}{%
5444     \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
5445     \stepcounter{line}%
5446     \ifsublines%
5447         \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
5448     %
5449     \stepcounter{subline}{1}%
5450     \def\label@refs{\theline|\thesubline}%
5451     \else%
5452     \def\label@refs{\theline|0}%
5453     \fi%
5454 }
5455 \def\labelrefsparseline#1|#2{#1}
5456 \def\labelrefsparsesubline#1|#2{#2}
5457 %

```

`\l@dmake@labels` The `\l@dmake@labels` macro gets executed when the labels file is read. For each label it defines a macro, whose name is made up partly from the label you supplied, that contains the page, line and sub-line numbers. But first it checks to see whether the label has already been used (and complains if it has).

The initial use of `\newcommand` is to catch if `\l@dmake@labels` has been previously defined (by a class or package).

#1 page number, #2 line number, #3 sub-line number, #4 pstart number, #5 label.

```

5457 \newcommand*{\l@dmake@labels}{%
5458 \def\l@dmake@labels#1|#2|#3|#4|#5{%
5459     \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
5460     else
5461         \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
5462     \fi
5463     \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
5464     {#1|#2|#3|#4|\relax}%
5465     \ignorespaces}

```

```
5464
5465 %
```

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.

```
5466 \AtBeginDocument{%
5467   \def\l@dmake@labels#1|#2|#3|#4|#5{%
5468 }
5469
5470 %
```

\@lab The \@lab command, which appears in the \linenum@out file, appends the current values of page, line and sub-line to the \labelref@list. These values are defined by the earlier \@page, \@nl, and the \sub@on and \sub@off commands appearing in the \linenum@out file.

TeX uses the page counter for page numbers. However, it appears that this is not the right place to grab the page number. That task is now done in the \edlabel macro. This version of \@lab appends just the current line and sub-line numbers to \labelref@list.

```
5471
5472 \newcommand*{\@lab}{%
5473   \ifledRcol
5474     \xright@appenditem{\linenumr@p{\line@numR}}|{%
5475       \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi}%
5476     \to\labelref@listR
5477   \else
5478     \xright@appenditem{\linenumr@p{\line@num}}|{%
5479       \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi}%
5480     \to\labelref@list
5481   \fi}
5482 %
```

\applabel \applabel, if called in \edtext will insert automatically both a start and an end label for the current edtext lines.

```
5483 \newcommand*{\applabel}[1]{%
5484   \if@edtext@secondarg%
5485 %
```

Label should not be already defined.

```
5486   \ifcsundef{the@label#1}{%
5487     \csdef{the@label#1}{\applabel}%
5488   }%
5489   {%
5490     \led@warn@DuplicateLabel{#1 (\applabel)}%
5491   }%
5492 %
```

Parse the `\edtext` line numbers.

```
5493 \expandafter\l@dp@rsefootsspec\l@d@nums|%
5494 %
```

Use the \TeX standard hack for label.

```
5495 \@bsphack%
5496 %
```

And now, write the data in the auxiliary file.

```
5497 \ifledRcol%
5498 \protected@write\@auxout{}%
5499 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstartR|{#1:start}}%
5500 \ifdef{\hypertarget}%
5501 {\Hy@raisedlink{\hypertarget{#1:start}}}%
5502 {}%
5503 \protected@write\@auxout{}%
5504 {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|\the\c@pstartR|{#1:end}}%
5505 \else%
5506 \protected@write\@auxout{}%
5507 {\string\l@dmake@labels\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstart|{#1:start}}%
5508 \ifdef{\hypertarget}%
5509 {\Hy@raisedlink{\hypertarget{#1:start}}}%
5510 {}%
5511 \protected@write\@auxout{}%
5512 {\string\l@dmake@labels\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|\the\c@pstart|{#1:end}}%
5513 \fi%
5514 %
```

Use the \TeX standard hack for label.

```
5515 \@esphack%
5516 %
```

Warning if `\applabel` is called outside of `\edtext`.

```
5517 \else%
5518 \led@warn@AppLabelOutSecondArgEdtext{#1}%
5519 \fi%
5520 %
```

End of `\applabel`

```
5521 }%
5522 %
```

`\edlabels` `\edlabelS` and `\edlabelE` are just used to mark the beginning and the end of a passage.
`\edlabelE`
`\edlabelSE`

```

5523 \newcommand{\edlabelS}[1]{%
5524   \edlabel{#1:start}%
5525 }
5526 \newcommand{\edlabelE}[1]{%
5527   \edlabel{#1:end}%
5528 }
5529 \newcommand{\edlabelSE}[1]{%
5530   \edlabelS{#1}%
5531   \edlabelE{#1}%
5532 }
5533 %

```

\wrap@edcrossref \wrap@edcrossref is called around all reledmac crossref commands, except those which start with x. It adds the hyperlink.

```

5534 \newrobustcmd{\wrap@edcrossref}[2]{%
5535   \ifdef{\hyperlink}%
5536     {\hyperlink{#1}{#2}}%
5537     {#2}%
5538 }
5539 %

```

\edpageref If the specified label exists, \edpageref gives its page number.

\xpageref For this reference command, as for the other two, a special version with prefix x is provided for use in places where the command is to be scanned as a number, as in \linenum. These special versions have two limitations: they do not print error messages if the reference is unknown, and they can't appear as the first label or reference command in the file; you must ensure that a \edlabel or a normal reference command appears first, or these x-commands will always return zeros.

L^AT_EX already defines a \pageref, so changing the name to \edpageref.

```

5540 \newcommand*{\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
5541 \newcommand*{\xpageref}[1]{\l@dgetref@num{1}{#1}}
5542
5543 %

```

\edlineref If the specified label exists, \lineref gives its line number.

\xlineref

```

5544 \newcommand*{\edlineref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{2}{#1}\xflagref{#1}}}%
5545 \newcommand*{\xlineref}[1]{\l@dgetref@num{2}{#1}}%
5546
5547 %

```

\sublineref If the specified label exists, \sublineref gives its sub-line number.

\xsublineref

```

5548 \newcommand*{\sublineref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{3}{#1}}}

```

```

5549 \newcommand*\xsublineref}[1]{\l@dgetref@num{3}{#1}}
5550
5551 %

```

\pstartref If the specified label exists, \pstartref gives its pstart number.

```

\pstartref
5552 \newcommand*\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{4}{#1}}}
5553 \newcommand*\xpstartref}[1]{\l@dgetref@num{4}{#1}}
5554
5555 %

```

\xflagref \xflagref finds the side flag of any ref defined with \edlabel.

```

5556 \newcommand*\xflagref}[1]{\l@dgetref@num{5}{#1}}
5557 %

```

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.

```

5558 \newcommand*\l@dref@undefined}[1]{%
5559 \expandafter\ifx\csname the@label#1\endcsname\relax
5560 \led@warn@RefUndefined{#1}%
5561 \fi}
5562
5563 %

```

\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.

```

5564 \newcommand*\l@dgetref@num}[2]{%
5565 \expandafter
5566 \ifx\csname the@label#2\endcsname \relax
5567 000%
5568 \else
5569 \expandafter\expandafter\expandafter
5570 \l@dlabel@parse\csname the@label#2\endcsname|#1%
5571 \fi}
5572
5573 %

```

`\l@dlabel@parse` Notice that we slipped another `|` delimiter into the penultimate line of `\l@dgetref@num`, to keep the ‘switch-number’ separate from the reference numbers. This `|` is used as another parameter delimiter by `\l@dlabel@parse`, which extracts the appropriate number from its first arguments. The `|`-delimited arguments consist of the expanded label-macro (three reference numbers), followed by the switch-number (1, 2, 3 or 4) which defines which of the earlier five numbers to pick out. (It was earlier given as the first argument of `\l@dgetref@num`.)

```

5574 \newcommand*{\l@dlabel@parse}{%
5575 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6{%
5576   \ifcase #6%
5577   \or #1%
5578   \or #2%
5579   \or #3%
5580   \or #4%
5581   \or #5%
5582   \fi}
5583 %

```

`\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.

```

5584 \newcommand*{\xxref}[2]{%
5585   {%
5586     \expandafter\ifx\csname the@label#1\endcsname \relax%
5587     \expandafter\let\csname the@@label#1\endcsname\zz@@@%
5588     \else%
5589       \expandafter\def\csname the@@label#1\endcsname{\l@dgetref@num
5590 {1}{#1}|\l@dgetref@num{2}{#1}|\l@dgetref@num{3}{#1}}%
5591       \fi%
5592       \expandafter\ifx\csname the@label#2\endcsname \relax%
5593       \expandafter\let\csname the@@label#2\endcsname\zz@@@%
5594       \else%
5595       \expandafter\def\csname the@@label#2\endcsname{\l@dgetref@num
5596 {1}{#2}|\l@dgetref@num{2}{#2}|\l@dgetref@num{3}{#2}}%
5597       \fi%
5598       \letcs{\@tempa}{the@@label#1}%
5599       \letcs{\@tempb}{the@@label#2}%
5600       \linenum{\@tempa|
5601       \@tempb}}%
5602 %

```

`\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).

Here we define hooks similar to some those related to critical footnotes or endnotes. So, first declare the default value of the hooks for the pseudo-series. Also declare the internal toggle which are switch by `reledmac`.

```

5602 \def\Xtwolines@appref{}%
5603 \def\Xtwolines@SEref{}%
5604
5605 \def\Xmorethantwolines@appref{}%
5606 \def\Xmorethantwolines@SEref{}%
5607
5608 \def\Xlinerangeseparator@appref{\endashchar}%
5609 \def\Xlinerangeseparator@SEref{\endashchar}%
5610
5611 \def\Xsublinesep@appref{\fullstop}%
5612 \def\Xsublinesep@SEref{\fullstop}%
5613
5614 \newtoggle{Xtwolinesbutnotmore@appref}%
5615 \newtoggle{Xtwolinesbutnotmore@SEref}%
5616
5617 \newtoggle{Xtwolinesonlyinsamepage@appref}%
5618
5619 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
5620
5621 \newtoggle{Xlineflag@appref}%
5622 \toggletrue{Xlineflag@appref}%Here exception
5623 \newtoggle{Xlineflag@SEref}%
5624 \toggletrue{Xlineflag@SEref}%%Here exception
5625
5626 \def\Xendtwolines@apprefwithpage{}%
5627 \def\Xendtwolines@SErefwithpage{}%
5628
5629 \def\Xendmorethantwolines@apprefwithpage{}%
5630 \def\Xendmorethantwolines@SErefwithpage{}%
5631
5632 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
5633 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
5634 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
5635
5636 \def\Xendbeforepagenumber@apprefwithpage{p.}%
5637 \def\Xendbeforepagenumber@SErefwithpage{p.}%
5638 \def\Xendbeforepagenumber@SEonlypage{p.}%
5639
5640 \def\Xendafterpagenumber@apprefwithpage{ }%
5641 \def\Xendafterpagenumber@SErefwithpage{ }%
5642
5643

```

```

5644 \def\Xendlineprefixsingle@apprefwithpage{}%
5645 \def\Xendlineprefixsingle@SErefwithpage{}%
5646
5647 \def\Xendlineprefixmore@apprefwithpage{}%
5648 \def\Xendlineprefixmore@SErefwithpage{}%
5649
5650 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
5651 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
5652
5653 \def\Xendsublinesep@apprefwithpage{\fullstop}%
5654 \def\Xendsublinesep@SErefwithpage{\fullstop}%
5655
5656 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
5657 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
5658
5659 \newtoggle{Xendlineflag@apprefwithpage}
5660 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
5661 \newtoggle{Xendlineflag@SErefwithpage}
5662 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
5663
5664 %

```

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.

```

5665
5666 \xdef\Xboxstartlinenum@appref{0pt}
5667 \xdef\Xboxstartlinenum@SEref{0pt}
5668
5669 \xdef\Xboxendlinenum@appref{0pt}
5670 \xdef\Xboxendlinenum@SEref{0pt}
5671
5672 \xdef\Xendboxstartlinenum@apprefwithpage{0pt}
5673 \xdef\Xendboxstartlinenum@SErefwithpage{0pt}
5674
5675 \xdef\Xendboxendlinenum@apprefwithpage{0pt}
5676 \xdef\Xendboxendlinenum@SErefwithpage{0pt}
5677
5678 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

5679 \newcommand\@apprefprefixsingle{}%
5680 \newcommand\@SErefprefixsingle{}%
5681
5682 \newcommand\@apprefprefixmore{}%
5683 \newcommand\@SErefprefixmore{}%
5684
5685 \newcommand{\setapprefprefixsingle}[1]{%

```

```

5686 \gdef\@apprefprefixsingle{#1}%
5687 }
5688 \newcommand{\setSErefprefixsingle}[1]{%
5689 \gdef\@SErefprefixsingle{#1}%
5690 }
5691
5692 \newcommand{\setapprefprefixmore}[1]{%
5693 \gdef\@apprefprefixmore{#1}%
5694 }
5695 \newcommand{\setSErefprefixmore}[1]{%
5696 \gdef\@SErefprefixmore{#1}%
5697 }
5698
5699 %

```

And not `\setSErefonlypageprefixsingle` and `\setSErefonlypageprefixmore`.

```

5700 \let\setSErefonlypageprefixsingle\XendbeforepagenumberSErefonlypage%
5701 \newcommand{\setSErefonlypageprefixmore}[1]{%
5702 \gdef\SErefonlypage@prefixmore{#1}%
5703 }%
5704 %

```

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.

```

5705
5706 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
5707 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
5708
5709 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
5710 {#1}{#2}{appref}}
5711 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
5712 {#1}{#2}{SEref}}
5713
5714 \newcommand{\reformatted@}[3]{%
5715 \def\do##1{%
5716 \setkeys[mac]{truefootnoteoption}{##1}%
5717 }%
5718 \notblank{#1}{\docsvlist{#1}}}%
5719 \xdef\@currentseries{#3}%
5720 \ifcempty{@#3prefixmore}%
5721 {\@apprefprefixsingle}%
5722 {%
5723 \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
5724 {\csuse{@#3prefixsingle}}%

```

```

5725     {\csuse{@#3prefixmore}}}%
5726   }%
5727 \ifboolexpr{%
5728   test{\ifcsundef{the@label#2:start}}}%
5729   or test{\ifcsundef{the@label#2:end}}}%
5730 }%
5731 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}}%
5732 {%
5733   \def\@this@crossref@start{#2:start}%
5734   \def\@this@crossref@end{#2:end}%
5735   \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|}%
5736   \undef\@this@crossref@end%
5737   \undef\@this@crossref@start%
5738 }%
5739 \def\do##1{%
5740   \setkeys[mac]{falsefootnoteoption}{##1}%
5741 }%
5742 \notblank{#1}{\docsvlist{#1}}{ }%
5743 }%
5744
5745 \newcommand{\reformattedwithpage@}[3]{%
5746   \def\do##1{%
5747     \setkeys[mac]{truefootnoteoption}{##1}%
5748   }%
5749   \notblank{#1}{\docsvlist{#1}}{ }%
5750   \xdef\@currentseries{#3withpage}%
5751   \ifboolexpr{%
5752     test{\ifcsundef{the@label#2:start}}}%
5753     or test{\ifcsundef{the@label#2:end}}}%
5754   }%
5755   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}}%
5756   {%
5757     \def\@this@crossref@start{#2:start}%
5758     \def\@this@crossref@end{#2:end}%
5759     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|}%
5760     \undef\@this@crossref@end%
5761     \undef\@this@crossref@start%
5762   }%
5763   \def\do##1{%
5764     \setkeys[mac]{falsefootnoteoption}{##1}%
5765   }%
5766   \notblank{#1}{\docsvlist{#1}}{ }%
5767 }%
5768
5769 \newcommand{\reformattedonlypage@}[3]{%
5770   \def\do##1{%

```

```

5771 \setkeys[mac]{truefootnoteoption}{##1}%
5772 }%
5773 \notblank{#1}{\docsvlist{#1}}{}%
5774 \xdef\@currentseries{#3onlypage}%
5775 \ifboolexpr{%
5776   test{\ifcsundef{the@label#2:start}}}%
5777   or test{\ifcsundef{the@label#2:end}}}%
5778 }%
5779 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
5780 {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}}%
5781   {%
5782     \printnpnum{%
5783       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
5784     }%
5785   }%
5786 {%
5787   \ifcsvoid{#3onlypage@prefixmore}%
5788     {}%
5789     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
5790   \ifdefined\linrangesep@%
5791     \printnpnum{%
5792       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
5793     \linrangesep@%
5794     \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
5795   }%
5796   \else%
5797     \printnpnum{%
5798       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
5799       \csuse{Xendlinrangeseparator@\@currentseries}%
5800       \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
5801     }%
5802     \fi%
5803   }%
5804 }%
5805 \def\do##1{%
5806   \setkeys[mac]{falsefootnoteoption}{##1}%
5807   }%
5808   \notblank{#1}{\docsvlist{#1}}{}%
5809 }%
5810 %

```

`\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`.

```

5811 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\

```

```
endcsname{#2}}
```

```
5812  
5813 %
```

(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. 123 and V.9 p. 92), 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`.

```
5814 \pretocmd{\XR@test}%  
5815   {\XR@test@mac+++#1#2#3#4+++}%  
5816   {}%  
5817   {}%  
5818 %
```

`\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`.

```
5819 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}  
5820 %
```

`\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`.

```
5821 \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  
5822   \ifx#1\l@dmake@labels%  
5823     \l@dmake@labels#2%  
5824   \else  
5825     \ifx#1\l@dmake@labelsR%  
5826       \l@dmake@labelsR #2%  
5827     \fi%  
5828   \fi%  
5829 }%  
5830 %
```

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.

```

5831 \pretocmd{\@xympar}%
5832   {\ifnumberedpar@
5833     \led@warn@NoMarginpars
5834     \@esphack
5835   \else}%
5836   {}%
5837   {}%
5838
5839 \apptocmd{\@xympar}%
5840   {\fi}%
5841   {}
5842   {}
5843
5844 %

```

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

```

5845 \newcount\sidenote@margin
5846 \newcommand*{\sidenotemargin}[1]{\{%
5847   \l@dgetsidenote@margin{#1}%
5848   \ifnum\@l@dtmptcntb>\m@ne
5849     \ifledRcol
5850       \global\sidenote@marginR=\@l@dtmptcntb
5851     \else
5852       \global\sidenote@margin=\@l@dtmptcntb
5853     \fi
5854   \fi}}
5855 \newcommand*{\l@dgetsidenote@margin}[1]{\%
5856   \def\@tempa{#1}\def\@tempb{left}%
5857   \ifx\@tempa\@tempb
5858     \l@dtmptcntb \z@
5859   \else
5860     \def\@tempb{right}%
5861     \ifx\@tempa\@tempb

```

```

5862 \l@l@tempcntb \@ne
5863 \else
5864 \def\@tempb{outer}%
5865 \ifx\@tempa\@tempb
5866 \l@l@tempcntb \tw@
5867 \else
5868 \def\@tempb{inner}%
5869 \ifx\@tempa\@tempb
5870 \l@l@tempcntb \thr@@
5871 \else
5872 \led@warn@BadSidenotemargin
5873 \l@l@tempcntb \m@ne
5874 \fi
5875 \fi
5876 \fi
5877 \fi}
5878 \sidenotemargin{right}
5879
5880 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
5881 \newbox\l@dlp@rbox
5882 \newbox\l@drp@rbox
5883
5884 %

```

`\ledlsnotewidth` These specify the width of the left/right boxes (initialised to `\marginparwidth`), their distance from the text (initialised to `\linenumsep`), and the fonts used.

```

\ledrsnotewidth
\ledlsnotesep
5885 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
5886 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
5887 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup
5888 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
5889 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
5890 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
5891
5892 %

```

`\ledleftnote` `\ledrightnote`, `\ledinnernote`, `\ledouternote` are the user commands for left, right, inner and outer sidenotes. The two last one are just alias for the two first one, depending of the page number. `\ledsidenote{<text>}` is the command for a moveable sidenote.

```

\ledsidenote
5893 \newcommand*\ledleftnote[1]{\edtext{\l@dlsnote{#1}}}
5894 \newcommand*\ledrightnote[1]{\edtext{\l@drsnote{#1}}}
5895 \newcommand*\ledsidenote[1]{\edtext{\l@dcnote{#1}}}%
5896 \newcommand*\ledinnernote[1]{\edtext{\l@disnote{#1}}}%
5897 \newcommand*\ledouternote[1]{\edtext{\l@dosnote{#1}}}%
5898 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.

```

\l@drsnote
\l@dcsnote
\l@desnote
\l@disnote
\newif\ifrightnoteup
\rightnoteuptrue

\newcommand*{\l@dlsnote}[1]{%
  \begingroup%
  \newcommand{\content}{#1}%
  \ifnumberedpar@
    \ifledRcol%
      \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}{%
        \to\inserts@listR
      }%
      \global\advance\insert@countR \@ne%
    \else%
      \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}{%
        \to\inserts@list
      }%
      \global\advance\insert@count \@ne%
    \fi
  \fi%
  \ignorespaces%
\endgroup%
}%

\newcommand*{\l@drsnote}[1]{%
  \begingroup%
  \newcommand{\content}{#1}%
  \ifnumberedpar@
    \ifledRcol%
      \xright@appenditem{\noexpand\l@drsnote{\expandonce\content}}{%
        \to\inserts@listR
      }%
      \global\advance\insert@countR \@ne%
    \else%
      \xright@appenditem{\noexpand\l@drsnote{\expandonce\content}}{%
        \to\inserts@list
      }%
      \global\advance\insert@count \@ne%
    \fi
  \fi\ignorespaces%
\endgroup%
}%

\newcommand*{\l@dcsnote}[1]{%
  \begingroup%
  \newcommand{\content}{#1}%
  \ifnumberedpar@
    \ifledRcol%
      \xright@appenditem{\noexpand\l@dcsnote{\expandonce\content}}{%
        \to\inserts@listR
      }%
      \global\advance\insert@countR \@ne%
    \else%

```

```

5946 \xright@appenditem{\noexpand\vl@dcnote{\expandonce\content}}}%
5947 \to\inserts@list
5948 \global\advance\insert@count \@ne%
5949 \fi
5950 \fi\ignorespaces%
5951 \endgroup%
5952 }%
5953
5954 \newcommand*{\l@disnote}[1]{%
5955 \begingroup%
5956 \newcommand{\content}{#1}%
5957 \ifnumberedpar%
5958 \ifledRcol%
5959 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
5960 \to\inserts@listR%
5961 \global\advance\insert@countR \@ne%
5962 \else%
5963 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
5964 \to\inserts@list%
5965 \global\advance\insert@count \@ne%
5966 \fi%
5967 \fi\ignorespaces%
5968 \endgroup%
5969 }%
5970
5971 \newcommand*{\l@dosnote}[1]{%
5972 \begingroup%
5973 \newcommand{\content}{#1}%
5974 \ifnumberedpar%
5975 \ifledRcol%
5976 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
5977 \to\inserts@listR%
5978 \global\advance\insert@countR \@ne%
5979 \else%
5980 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
5981 \to\inserts@list%
5982 \global\advance\insert@count \@ne%
5983 \fi%
5984 \fi\ignorespaces%
5985 \endgroup%
5986 }%
5987
5988 %

```

\vl@dlsnote Put the left/right text into boxes, but just save the moveable text. **\l@dcnotetext**, **\vl@drsnote** **\l@dcnotetext@l** and **\l@dcnotetext@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@dcstotetext` in any cases.
- Store the content of `\rightsidenote` to:
 - `\l@dcstotetext` if `\ledsidenote` is to be put on right.
 - `\l@dcstotetext@r` if `\ledsidenote` is to be put on left.
- Store the content of `\leftsidenote` to:
 - `\l@dcstotetext` if `\ledsidenote` is to be put on left.
 - `\l@dcstotetext@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.

```

5989 \newcommand*{\vl@dlsnote}[1]{%
5990   \ifledRcol{%
5991     \@l@dttempcntb=\sidenote@marginR%
5992     \ifnum\@l@dttempcntb>\@ne%
5993       \advance\@l@dttempcntb by\page@numR%
5994     \fi%
5995   \else%
5996     \@l@dttempcntb=\sidenote@margin%
5997     \ifnum\@l@dttempcntb>\@ne%
5998       \advance\@l@dttempcntb by\page@num%
5999     \fi%
6000   \fi%
6001   \ifodd\@l@dttempcntb%
6002     \listgadd{\l@dcstotetext@l}{#1}%
6003   \else%
6004     \listgadd{\l@dcstotetext}{#1}%
6005   \fi
6006 }
6007 \newcommand*{\vl@drsnote}[1]{%
6008   \ifledRcol{%
6009     \@l@dttempcntb=\sidenote@marginR%
6010     \ifnum\@l@dttempcntb>\@ne%
6011       \advance\@l@dttempcntb by\page@numR%
6012     \fi%
6013   \else%
6014     \@l@dttempcntb=\sidenote@margin%
6015     \ifnum\@l@dttempcntb>\@ne%
6016       \advance\@l@dttempcntb by\page@num%
6017     \fi%
6018   \fi%
6019   \ifodd\@l@dttempcntb%
6020     \listgadd{\l@dcstotetext}{#1}%
6021   \else%
6022     \listgadd{\l@dcstotetext@r}{#1}%
6023   \fi%

```

```

6024 }
6025 \newcommand*{\vl@dcnote}[1]{\listgadd{\l@dcnotetext}{#1}}
6026
6027 \newcommand{\vl@disnote}[1]{%
6028   \ifledRcol{%
6029     \@tempcnta=\page@numR%
6030   }%
6031   \else%
6032     \@tempcnta=\page@num%
6033   \fi%
6034   \ifodd\@tempcnta% ODD => right page => inner side = left side
6035   \vl@dlsnote{#1}%
6036   \else%
6037     \vl@drsnote{#1}%
6038   \fi%
6039 }%
6040
6041 \newcommand{\vl@dosnote}[1]{%
6042   \ifledRcol{%
6043     \@tempcnta=\page@numR%
6044   }%
6045   \else%
6046     \@tempcnta=\page@num%
6047   \fi%
6048   \ifodd\@tempcnta% ODD => right page => outer side = right side
6049   \vl@drsnote{#1}%
6050   \else%
6051     \vl@dlsnote{#1}%
6052   \fi%
6053 }%

```

`\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.

```

6054 \newcommand*{\setl@dlp@rbox}[1]{%
6055   \begingroup%
6056   \parindent\z@ \hsize=\ledlsnotewidth%
6057   \ledlsnotefontsetup% We kept it outside of the vbox, because can affect
6058   the ragging
6059   \global\setbox\l@dlp@rbox%
6060   \ifleftnoteup%
6061     =\vbox to\z@{\{\ledlsnotefontsetup\vss #1}\}% We put \
6062     ledlsnotefontsetup inside footnote because required for color command. Note
6063     the {} to keep setting local.
6064   \else%
6065     =\vbox to 0.70\baselineskip{\{\ledlsnotefontsetup\strut#1\vss}\}%
6066   \fi%
6067   \endgroup%
6068 }

```

```

6066 \newcommand*\setl@drp@rbox}[1]{%
6067   \begingroup%
6068     \parindent\z@\hsize=\ledrsnotewidth%
6069     \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
6070     the ragging
6071     \global\setbox\l@drp@rbox%
6072     \ifrightnoteup%
6073       =\vbox to\z@{\{\ledrsnotefontsetup\vss#1\}}%We put \ledrsnotefontsetup
        inside footnote because required for color command. Note the {} to keep
        setting local.
6074       \else%
6075         =\vbox to0.7\baselineskip{\{\ledrsnotefontsetup\strut#1\vss\}}%
6076         \fi%
6077     \endgroup%
6078   }%
6079 \newif\ifleftnoteup
6080 \leftnoteuptrue
6081 %

```

\@sidenotesep This macro is used to separate sidenotes of the same line.

```

6082 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
6083 \newcommand{\@sidenotesep}{, }
6084 %

```

\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.

```

6085 \newcommand*\affixside@note){%
6086   \def\sidenotecontent@{}%
6087   \numgdef{\itemcount@}{0}%
6088   \def\do##1{%
6089     \ifnumequal{\itemcount@}{0}%
6090     {%
6091       \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
6092       {\appto\sidenotecontent@{\@sidenotesep ##1}}%
6093     }%
6094     \numgdef{\itemcount@}{\itemcount@+\@ne}%
6095   }%
6096   \dolistloop{\l@dcnotes@text}%
6097   \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
6098 %

```

And we do the same for left and right notes (not movable).

```

6099 \gdef\@templ@d{%
6100 \gdef\@templ@n{\l@dcsnotetext\l@dcsnotetext@1\l@dcsnotetext@r}%
6101 \ifx\@templ@d\@templ@n \else%
6102 \if@twocolumn%
6103 \if@firstcolumn%
6104 \setl@dlp@rbox{##1}{\sidenotecontent@}%
6105 \else%
6106 \setl@drp@rbox{\sidenotecontent@}%
6107 \fi%
6108 \else%
6109 \l@dtempcntb=\sidenote@margin%
6110 \ifnum\l@dtempcntb>\@ne%
6111 \advance\l@dtempcntb by\page@num%
6112 \fi%
6113 \ifodd\l@dtempcntb%
6114 \setl@drp@rbox{\sidenotecontent@}%
6115 \gdef\sidenotecontent@{%
6116 \numgdef{\itemcount@}{0}%
6117 \dolistloop{\l@dcsnotetext@1}%
6118 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
6119 \setl@dlp@rbox{\sidenotecontent@}%
6120 \else%
6121 \setl@dlp@rbox{\sidenotecontent@}%
6122 \gdef\sidenotecontent@{%
6123 \numgdef{\itemcount@}{0}%
6124 \dolistloop{\l@dcsnotetext@r}%
6125 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
6126 \setl@drp@rbox{\sidenotecontent@}%
6127 \fi%
6128 \fi%
6129 \fi%
6130 }
6131 %

```

XXV Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfeetbeginmini` These will be the hooks in `\@iiiminipage` and `\endminipage`.
`\l@dfeetendmini` They can be extended to handle other things if necessary.

```

6132 \ifnoledgroup@\else%
6133 \newcommand*{\l@dfeetbeginmini}{\ledgrouptrue\l@dedsbeginmini\l@dfeetbeginmini}
6134 \newcommand*{\l@dfeetendmini}{%
6135 \IfStrEq{critical-familiar}{\@mpfnpos}%

```

```

6136     {\l@dedendmini\l@dfamendmini}%
6137     {%
6138         \IfStrEq{familiar-critical}{\@mpfnpos}%
6139         {\l@dfamendmini\l@dedendmini}%
6140         {\l@dedendmini\l@dfamendmini}%
6141     }%
6142 }%
6143 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

```

\l@dedendmini
6144 \newcommand*{\l@dedbeginmini}{%
6145     \unless\ifnocritical@%
6146     \def\do##1{\csletcs{v##1footnote}{mpv##1footnote}}%
6147     \dolistloop{\@series}%
6148     \fi%
6149 }
6150 \newcommand*{\l@dedendmini}{%
6151     \unless\ifnocritical@%
6152     \ifl@dpairing%
6153         \ifledRcol%
6154             \flush@notesR%
6155         \else%
6156             \flush@notes%
6157         \fi%
6158     \fi
6159     \def\do##1{%
6160         \ifvoid\csuse{mp##1footins}\else%
6161         \ifl@dpairing\ifparledgroup%
6162             \ifledRcol%
6163                 \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR\
skip\@nameuse{mp##1footins}}%
6164             \else%
6165                 \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL
+ \skip\@nameuse{mp##1footins}}%
6166             \fi%
6167             \fi\fi%
6168             \csuse{mp##1footgroup}{##1}%
6169         \fi}%
6170     \dolistloop{\@series}%
6171     \fi%
6172 }%
6173 %
6174 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
6175 \newcommand*{\l@dfambeginmini}{%
6176     \unless\ifnofamiliar@%
6177     \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%

```

```

6178 \dolistloop{\@series}%
6179 \fi%
6180 }%
6181
6182 \newcommand*{\l@dfamendmini}{%
6183 \unless\ifnofamiliar%
6184 \def\do##1{%
6185 \ifvoid\csuse{mpfootins##1}\else%
6186 \csuse{mpfootgroup##1}{##1}%
6187 \fi}%
6188 \dolistloop{\@series}%
6189 \fi%
6190 }%
6191 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

6192 \patchcmd%
6193 {\@iiiminipage}%
6194 {\let\@footnotetext\@mpfootnotetext}%
6195 {\let\@footnotetext\@mpfootnotetext\l@dfeetbeginmini}%
6196 {}%
6197 {\led@error@fail@patch@\@iiiminipage}%
6198 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

6199 \patchcmd%
6200 {\endminipage}%
6201 {\footnoterule}%
6202 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
6203 {}%
6204 {\led@error@fail@patch@endminipage}%
6205
6206 \patchcmd%
6207 {\endminipage}%
6208 {\@minipagefalse}%
6209 {\l@dfeetendmini\@minipagefalse}%
6210 {}%
6211 {\led@error@fail@patch@endminipage}%
6212
6213 %

```

`\l@dunboxmpfoot` `\l@dunboxmpfoot` insert normal footnotes for `ledgroup`.
`edgroup@beforenormalnotes`

```

6214 \newcommand*{\l@dunboxmpfoot}{%
6215 \vskip\skip\@mpfootins
6216 \normalcolor
6217 \footnoterule
6218 \l@advance@parledgroup@beforenormalnotes

```

```

6219 \unvbox\@mpfootins%
6220 }
6221 %

```

When using parallel ledgroup, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

6222 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
6223   \ifparledgroup
6224     \ifl@pairing
6225       \ifledRcol
6226         \dimgdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
6227       \else
6228         \dimgdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
6229       \fi
6230     \fi
6231   \fi
6232 }
6233 %

```

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.

```

6234 \newenvironment{ledgroup}{%
6235   \resetprevpage@num%
6236   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
6237   \let\@footnotetext\@mpfootnotetext
6238   \l@dfeetbeginmini%
6239 }{%
6240   \par
6241   \unskip
6242   \ifvoid\@mpfootins\else
6243     \l@dunboxmpfoot
6244   \fi
6245   \l@dfeetendmini%
6246   \@ledgroupfalse%
6247 }
6248
6249
6250
6251 %

```

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.

```

6252 \newenvironment{ledgroupsize}[2][1]{%
6253   %

```

Set the various text measures.

```
6254 \hsize #2\relax
6255 %
```

Initialize fills for centering.

```
6256 \let\ledllfill\hfil
6257 \let\ledrlfill\hfil
6258 \def\@tempa{#1}\def\@tempb{1}%
6259 %
```

Left adjusted numbered lines

```
6260 \ifx\@tempa\@tempb
6261 \let\ledllfill\relax
6262 \else
6263 \def\@tempb{r}%
6264 \ifx\@tempa\@tempb
6265 %
```

Right adjusted numbered lines

```
6266 \let\ledrlfill\relax
6267 \fi
6268 \fi
6269 %
```

Set up the footnoting.

```
6270 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
6271 \let\@footnotetext\@mpfootnotetext
6272 \l@dfetbeginmini%
6273 }{
6274 \par
6275 \unskip
6276 \ifvoid\@mpfootins\else
6277 \l@dunboxmpfoot
6278 \fi
6279 \l@dfetendmini%
6280 }
6281 %
6282 %
```

Close the \ifnoledgroup@else.

```
6283 \fi%
6284 %
```

\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`.

```
6285 \newif\ifledgroupnotesL@
6286 \newif\ifledgroupnotesR@
6287 %
```

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`.

```

6288 \AtBeginDocument{%
6289   \unless\ifl@imakeidx%
6290     \ifpackageloaded{imakeidx}{\led@error@ImakeidxAfterEledmac}{}%
6291   \fi%
6292   \unless\ifl@indextools%
6293     \ifpackageloaded{indextools}{\led@error@indextoolsAfterEledmac}{}%
6294   \fi%
6295 }
6296 %

```

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
6297 \newcommand{\pagelinesep}{-}
6298 \newcommand{\edindexlab}{${&}}
6299 \newcounter{labidx}
6300 \setcounter{labidx}{0}
6301
6302 %

```

`\doedindexlabel` This macro sets an `\edlabel`.

```

6303 \newcommand{\doedindexlabel}{%
6304   \stepcounter{labidx}%
6305   \edlabel{\edindexlab\thelabidx}%
6306 }
6307
6308 %

```

`\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`.

```

6309 \newcounter{pageline}%
6310 \renewcommand{\thepageline}{%
6311   \thepage%
6312   \pagelinesep%
6313   \xlineref{\edindexlab\thelabidx}%
6314 }
6315 %

```

`\thestartpageline` These macros make up the page/line start/end number when the `\edindex` command is called in critical notes.
`\theendpageline`

```

6316 \newcommand{\thestartpageline}{%
6317   \l@dparsedstartpage%
6318   \pagelinesep%
6319   \l@dparsedstartline%
6320 }
6321 \newcommand{\theendpageline}{%
6322   \l@dparsedendpage%
6323   \pagelinesep%
6324   \l@dparsedendline%
6325 }
6326 %

```

XXVI.3 Code specific to \edindex in critical footnotes

`\if@edindex@fornote@true` This boolean test is switching at the beginning of each critical note, to allow index referring to this note.

```

6327 \newif\if@edindex@fornote@
6328 %

```

`\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 |.

```

6329 \newcommand{\prepare@edindex@fornote}[1]{%
6330   \l@dp@rsefootspec#1|%
6331   \@edindex@fornote@true%
6332 }
6333 %

```

`\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.

```

6334 \newcommand{\get@edindex@ledinnote@command}{%
6335   \ifxindy@%
6336     \gdef\@ledinnote@command{%
6337       ledinnote\thelabidx%
6338     }%
6339     \ifxindyhyperref@%
6340       \immediate\write\eledmac@xindy@out{%
6341         (define-attributes ("ledinnote\thelabidx"))^^J
6342         \space\space(markup-locref^^J
6343         \eledmacmarkuplocrefdepth^^J

```

```

6344 :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
}{"^^J
6345 :close "}"^^J
6346 :attr "ledinnote\thelabidx"^^J
6347 )
6348 }%
6349 \else%
6350 \immediate\write\eledmac@xindy@out{%
6351 (define-attributes ("ledinnote\thelabidx"))^^J
6352 \space\space(markup-locoref^^J
6353 \eledmacmarkuplocorefdepth^^J
6354 :open "\string\ledinnote{\@index@command}{"^^J
6355 :close "}"^^J
6356 :attr "ledinnote\thelabidx"^^J
6357 )
6358 }%
6359 \fi%
6360 %

```

If we do not use xindy option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

6361 \else%
6362 \gdef\@ledinnote@command{%
6363 ledinnote[\edindexlab\thelabidx]{\@index@command}%
6364 }%
6365 \fi%
6366 }
6367 %

```

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 |.

```

6368 \def\get@index@command#1|#2+{%
6369 \gdef\@index@txt{#1}%
6370 \gdef\@index@command{#2}%
6371 \xdef\@index@parenthesis{}%
6372 \IfBeginWith{\@index@command}{({}%
6373 \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
6374 \global\let\@index@command\@index@command@%
6375 \xdef\@index@parenthesis{({}%
6376 }{}%
6377 \IfBeginWith{\@index@command}{})}%
6378 \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
6379 \global\let\@index@command\@index@command@%
6380 \xdef\@index@parenthesis{)}%
6381 }{}%
6382 }
6383 %

```

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`

```

6384 \newcommandx{\ledinnote}[3][1,usedefault]{%
6385   \ifboolexpr{%
6386     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
6387     or%
6388     bool {xindyhyperref@}%
6389   }%
6390   {%
6391     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
6392   }%
6393   {%
6394     \csuse{#2}{\ledinnotemark{#3}}%
6395   }%
6396 }%
6397 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage
6398   {#2}}}%
6399 }%

```

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
6400 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
6401   \ifl@imakeidx%
6402     \if@edindex@fornote%
6403       \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6404       \get@edindex@ledinnote@command%
6405       \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
)}{\thestartpageline}%
6406       \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%
6407     \else%
6408       \get@edindex@hyperref{#2}%
6409       \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%

```

```

6410 \fi%
6411 \else%
6412 \if@edindex@fornote@%
6413 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6414 \get@edindex@ledinnote@command%
6415 \expandafter\protected@write\@indexfile{%
6416 {\string\indexentry{\@index@txt|(\@ledinnote@command){\thestartpageline}
6417 }%
6418 \expandafter\protected@write\@indexfile{%
6419 {\string\indexentry{\@index@txt|)\@ledinnote@command){\theendpageline}
6420 }%
6421 \else%
6422 \protected@write\@indexfile{%
6423 {\string\indexentry{#2}{\thepageline}
6424 }%
6425 \fi%
6426 \fi%
6427 \endgroup
6428 \@esphack%
6429 }
6430 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

6431 \pretocmd{\makeindex}{%
6432 \def\edindex{\@bsphack
6433 \doedindexlabel
6434 \begingroup
6435 \@sanitize
6436 \@wredindex}}{}{}
6437 \newcommand{\edindex}[1]{\@bsphack\@esphack}
6438 %

```

XXVI.7 Hyperlink

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```

6439 \newcommand{\hyperlinkformat}[3]{%
6440 \ifstrempy{#1}%
6441 {\hyperlink{#2}{#3}}%
6442 {\csuse{#1}{\hyperlink{#2}{#3}}%
6443 }}
6444 %

```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```

6445 \newcommand{\hyperlinkR}[2]{%

```

```

6446 \hyperlink{#1}{#2\@Rlineflag}%
6447 }%
6448
6449 %

```

\hyperlinkformatR \hyperlinkformatR command is to be used to create a internal hyperlink, a format and a \@Rlineflag, when indexing.

```

6450 \newcommand{\hyperlinkformatR}[3]{%
6451   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
6452 }%
6453
6454 %

```

\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).

```

6455 \newcommand{\get@edindex@hyperref}[1]{%
6456 %

```

We have to disable temporary spaces to work through a xstring bug (or feature?)

```

6457 \edef\temp@{%
6458   \catcode`\ =9 %space need for catcode
6459   \detokenize{#1}%For active character in unicode
6460   \catcode`\ =10 % space need for catcode
6461 }%
6462 %

```

Now, we define \@edindex@hyperref if the hyperindex of hyperref is enabled.

```

6463 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
6464   \IfSubStr{\temp@}{|}%
6465   {\get@index@command#1+%
6466     \ifledRcol%
6467     \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6468       hyperlinkformatR{\@index@command}%
6469       {\edindexlab\thelabidx}}%
6470     \else%
6471     \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6472       hyperlinkformat{\@index@command}%
6473       {\edindexlab\thelabidx}}%
6474     \fi%
6475   }%
6476   {\get@index@command#1|+%
6477     \ifledRcol%
6478     \gdef\@edindex@hyperref{|\hyperlinkR{\edindexlab\thelabidx}}%
6479     \else%
6480     \gdef\@edindex@hyperref{|\hyperlink{\edindexlab\thelabidx}}%
6481     \fi%
6482   }%
6483 }%
6484 %

```

```

6485 % If we use both xindy and hyperref, first get the \protect\cs{
      index@command} command.
6486 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
6487 % \begin{macrocode}
6488 {\ifxindyhyperref%
6489   \IfSubStr{\temp@}{|}%
6490   {\get@index@command#1+}%
6491   {\get@index@command#1|+}%
6492   \gdef\@edindex@hyperref{|eledmac\thelabidx}%
6493 %

```

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.

```

6494   \IfStrEq{\@index@parenthesis}{(}%
6495   {%
6496     \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
6497     \gdef\@edindex@hyperref{|(eledmac\thelabidx}%
6498   }%
6499   {}%
6500 %

```

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.

```

6501   \IfStrEq{\@index@parenthesis}{)}%
6502   {%
6503     \xdef\@edindex@hyperref{)|(eledmac\csuse{xindyparenthesis@\@index@txt}}%
6504     \global\csundef{xindyparenthesis@\@index@txt}%
6505   }%
6506 %

```

Write in the .xdy file the attributes of the location.

```

6507   {%
6508     \immediate\write\eledmac@xindy@out{%
6509       (define-attributes ("eledmac\thelabidx"))^^J
6510       \space\space(markup-locoref^^J
6511       \eledmacmarkuplocorefdepth^^J
6512       :open "\string\hyperlink%
6513         \ifledRcol R\fi%
6514         {\edindexlab\thelabidx}%
6515         {\ifdefempty{\@index@command}%
6516           {}%
6517           {\@backslashchar\@index@command}%
6518         {"^^J
6519       :close "})"^^J
6520       :attr "eledmac\thelabidx"^^J
6521     )

```

```

6522     }%
6523     }%
6524 %

```

And now, in any other case.

```

6525     \else%
6526         \gdef\@index@txt{#1}%
6527         \gdef\@edindex@hyperref{}%
6528     \fi%
6529     }%
6530 }
6531 %

```

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 `indextols` package. Its only argument is the note series.

```

6532 \newcommand{\led@set@index@fornote}[1]{%
6533     \ifbool{indtl@innote}%
6534     {\let\index\nindex}%
6535     {}%
6536     \ifbool{indtl@notenumber}%
6537     {%
6538         \renewcommand{\index}[2][\indtl@jobname]{%
6539             \orig@@index[##1]{%
6540                 ##2|innotenumber{\csuse{thefootnote#1}}}%
6541             }%
6542         }%
6543     }%
6544     {}%
6545 }%
6546 %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

6547 \newcommand{\led@reinit@index@fornote}{%
6548     \ifbool{indtl@innote}%
6549     {\let\index\orig@@index}%
6550     {}%
6551     \ifbool{indtl@notenumber}%
6552     {\let\index\orig@@index}%
6553     {}%
6554 }%
6555 %

```

XXVII Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```
6556 \AtBeginDocument{%
6557   \ifpackageloaded{glossaries}{%
6558     %
```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```
6559   \renewcommand{\do}[1]{%
6560     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
6561       \doedindexlabel%
6562       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
6563     }%
6564     \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,3,usedefault]{%
6565       \doedindexlabel%
6566       \csname#1\endcsname*[counter=pageline,##1]{##2}[##3]%
6567     }%
6568   }%
6569   \docsvlist{%
6570     gls,%
6571     Gls,%
6572     GLS,%
6573     glspl,%
6574     Glspl,%
6575     GLSpl,%
6576     glstext,%
6577     Glstext,%
6578     GLStext,%
6579     Glsfirst,%
6580     GLSfirst,%
6581     glsplural,%
6582     Glsplural,%
6583     GLSplural,%
6584     glsfirstplural,%
6585     Glsfirstplural,%
6586     GLSfirstplural,%
6587     glsname,%
6588     Glsname,%
6589     GLSname,%
6590     glssymbol,%
6591     Glsymbol,%
6592     GLSsymbol,%
6593     glsdesc,%
6594     Glsdesc,%
6595     GLSdesc,%
6596     glsuseri,%
```

```

6597     Glsuseri,%
6598     GLSuseri,%
6599     glsuserii,%
6600     Glsuserii,%
6601     GLSuserii,%
6602     glsuseriii,%
6603     Glsuseriii,%
6604     GLSuseriii,%
6605     glsuseriv,%
6606     Glsuseriv,%
6607     GLSuseriv,%
6608     glsuserv,%
6609     Glsuserv,%
6610     GLSuserv,%
6611     glsuservi,%
6612     Glsuservi,%
6613     GLSuservi%
6614     }%
6615 %

```

First those which arguments are [*options*]{*label*}{*link text*}.

```

6616     \renewcommand{\do}[1]{%
6617     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,
usedefault]{%
6618     \doedindexlabel%
6619     \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
6620     }%
6621     \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,usedefault]{%
6622     \doedindexlabel%
6623     \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
6624     }%
6625     }%
6626     \docsvlist{glsdisp,glslink}%
6627     }{%
6628     }%
6629 %

```

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.

```
6630 \def\@hangingsymbol{}
6631 \newcommand*{\sethangingsymbol}[1]{%
6632   \gdef\@hangingsymbol{#1}%
6633 }%
6634 \newif\ifinstanza
6635 %
```

`\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.

```
6636 \newif\ifinserthangingsymbol
6637 \newcommand{\inserthangingsymbol}{%
6638   \ifinserthangingsymbol%
6639     \ifinstanza%
6640       \@hangingsymbol%
6641     \fi%
6642   \fi%
6643 }
6644 %
```

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.

```
6645 \newcommand*{\ampersand}{\char`\&}
6646
6647 %
```

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.

```
6648 \chardef\body=\catcode`\@
6649 \catcode`\@=11
6650 \chardef\next=\catcode`\&
6651 \catcode`\&=\active
6652
6653 %
```

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.

```
6654 \newcount\stanza@count
6655 \newlength{\stanzaindentbase}
6656 \setlength{\stanzaindentbase}{20pt}
6657
6658 %
```

`\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.

```
6659 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
6660 \newcommand*{\setstanzavalues}[2]{\def\@tempa{#2,,|}%
6661   \stanza@count\z@
6662   \def\next{\expandafter\strip@szacnt\@tempa
6663     \ifx\@tempb\empty\let\next\relax\else
6664     \expandafter\mathchardef\csname #1@number\stanza@count
6665     \@endcsname\@tempb\relax
6666     \advance\stanza@count\@ne\fi\next}%
6667   \next}
6668
6669 %
```

`\setstanzaindents` In the original edmac, `\setstanzavalues{sza}{\langle...⟩}` had to be called to set the in-
`\setstanzapenalties` dents, and similarly `\setstanzavalues{szp}{\langle...⟩}`. 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).

```
6670 \newcommand*{\setstanzaindents}[1]{\setstanzavalues{sza}{#1}}
6671 \newcommand*{\setstanzapenalties}[1]{\setstanzavalues{szp}{#1}}
6672 %
6673 %
```

`\managestanza@modulo` Since version 0.13, the `stanzaindentrepetition` counter can be used when the indentation is repeated every *n* verses. The `\managestanza@modulo` is a command which modifies the counter `stanza@modulo`. The command adds 1 to `stanza@modulo`, but if `stanza@modulo` is equal to the `stanzaindentrepetition` counter, the command restarts it.

```
6674 \newcounter{stanzaindentrepetition}
6675 \newcount\stanza@modulo
6676
```

```

6677 \newcommand*{\managestanza@modulo}[0]{%
6678     \advance\stanza@modulo\@ne%
6679     \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
6680     \stanza@modulo\@ne%
6681     \fi%
6682 }
6683 %

```

\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 skips the current verse for the repetition of stanza indent.

\stanzaindent*

```

6684 \newcommand{\stanzaindent}[1]{%
6685     \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
6686     \ignorespaces%
6687 }%
6688 \WithSuffix\newcommand\stanzaindent*[1]{%
6689     \stanzaindent{#1}%
6690     \global\advance\stanza@modulo-\@ne%
6691     \ifnum\stanza@modulo=0%
6692     \global\stanza@modulo=\value{stanzaindentsrepetition}%
6693     \fi%
6694     \ignorespaces%
6695 }%
6696 %

```

XXVIII.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanzaa97 \newcounter{stanza}
6698 \renewcommand{\thestanza}{%
6699     \textbf{\arabic{stanza}}%
6700 }
6701 %

```

\ifnumberstanza Then, macro to activate automatically numbering of stanza.

```

6702 \newif\ifnumberstanza%
6703 %

```

\@insertstanzanumber Now, macro called at the first line of of verse of a stanza.

```

6704 \newcommand{\@insertstanzanumber}[0]{%
6705     \ifnumberstanza%
6706     \ifl@dpairing%
6707     \ifledRcol%
6708     \stanzanumwrapper{\thestanzaR}%
6709     \else%

```

```

6710     \stanzanumwrapper{\thestanzaL}%
6711     \fi%
6712     \else%
6713     \stanzanumwrapper{\thestanza}%
6714     \fi%
6715     \setline{1}%
6716     \fi%
6717 }%
6718 %

```

`\@advancestanzanumber` Also a command to advance the counter of stanza.

```

6719 \newcommand{\@advancestanzanumber}[0]{%
6720   \ifnumberstanza%
6721   \ifl@dpairing%
6722   \ifledRcol%
6723   \addtocounter{stanzaR}{1}%
6724   \else%
6725   \addtocounter{stanzaL}{1}%
6726   \fi%
6727   \else%
6728   \addtocounter{stanza}{1}%
6729   \fi%
6730 \fi%
6731 }%
6732 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

6733 \newcommand{\stanzanumwrapper}[1]{%
6734   \flagstanza{#1}%
6735 }%
6736 %

```

XXVIII.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

6737 \newcommand{\printstanza}[0]{%
6738   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
6739     l@dprintingcolumns}}{%
6740     \ifledRcol%
6741     \thestanzaR%
6742     \else%
6743     \thestanzaL%
6744     \fi%
6745   }{%
6746     \thestanza%
6747   }%

```

```
6747 }
6748 %
```

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.

```
6749 \newcommandx{\stanza@line}[1][1]{
6750   \ifnum\value{stanzaindentsrepetition}=0
6751     \parindent=\csname sza@\number\stanza@count
6752       @\endcsname\stanzaindentbase
6753   \else
6754     \parindent=\csname sza@\number\stanza@modulo
6755       @\endcsname\stanzaindentbase
6756   \managestanza@modulo
6757   \fi
6758   \pstart[#1]\stanza@hang\ignorespaces}
6759 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
6760   \hangindent\expandafter
6761   \noexpand\csname sza@0@\endcsname\stanzaindentbase
6762   \hangafter\@ne}
6763 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
6764   \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
6765   \penalty\fi\count@}
6766 %
```

`\@startstanza` Now we have the components of the `\stanza` macro, which appears at the start of a group of lines. This macro initializes the count and checks to see if hanging indentation and penalties are to be included. Hanging indentation suspends the line count, so that the enumeration is by verse line rather than by print line. If the print line count is desired, invoke `\let\startlock\relax` and do the same for `\endlock`. Here and above we have used `\xdef` to make the stored macros take up a bit less space, but it also makes them more obscure to the reader. Lines of the stanza are delimited by ampersands `&`. The last line of the stanza must end with `\&`.

```
6767 \xdef\@startstanza[#1]{%
6768   \noexpand\instanzatrue\expandafter
6769   \begingroup%
6770   \catcode`\noexpand\&\active%
6771   \global\stanza@count\@ne\stanza@modulo\@ne
6772   \noexpand\ifnum\expandafter\noexpand
```

```

6773 \csname sza@0@\endcsname=\z@\let\noexpand\stanza@hang\relax
6774 \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
6775 \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
6776 \expandafter\noexpand\csname szp@0@\endcsname=\z@
6777 \let\noexpand\sza@penalty\relax\noexpand\fi%
6778 \def\noexpand&{%
6779     \noexpand\newverse[] []}%
6780 \def\noexpand\&\noexpand\@stopstanza}%
6781 \noexpand\@advancestanza%
6782 \noexpand\stanza@line[#1]%
6783 \noexpand\@insertstanza%
6784 \let\par\relax\ignorespaces%No paragraph in verses
6785 }
6786
6787 \newcommand{\stanza}[1][1,usedefault]{%
6788     \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}} and test{\ifstrempy
6789     {\@startstanza[\at@every@stanza]}}%
6790     {\@startstanza[#1]}}%
6791 }%
6792
6793 \newcommand{\@stopstanza}[1][1,usedefault]{%
6794     \unskip%
6795     \endlock%
6796     \ifboolexpr{not test{\ifdefvoid{\at@every@stop@stanza}} and test{\
6797     ifstrempy{#1}}}%
6798     {\pend[\at@every@stop@stanza]}}%
6799     {\pend[#1]}}%
6800     \endgroup%
6801     \instanzafalse%
6802 }
6803
6804 \newcommand{\AtEveryStopStanza}[1]{%
6805     \ifstrempy{#1}%
6806     {\xdef\at@every@stop@stanza{}}%
6807     {\gdef\at@every@stop@stanza{#1}}%
6808 }%
6809 \def\at@every@stop@stanza{}%
6810
6811 \newcommand{\AtEveryStanza}[1]{%
6812     \ifstrempy{#1}%
6813     {\xdef\at@every@stanza{}}%
6814     {\gdef\at@every@stanza{#1}}%
6815 }%
6816 \def\at@every@stanza{}%
6817
6818 \newcommand*{\newverse}[2][1,2,usedefault]{%
6819     \unskip%
6820     \endlock\pend[#1]\sza@penalty\global%

```

```

6821 \advance\stanza@count\@ne\stanza@line[#2]%
6822 }
6823
6824 %

```

\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`.

```

6825 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
6826 \hskip -#1\llap{#2}\hskip #1\ignorespaces}
6827
6828 %

```

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.

```

6829 \catcode`\&=\next
6830 \catcode`\@=\body
6831 \setstanzavalues{szp}{0}
6832
6833 %

```

XXIX Apparatus of Manuscripts

XXIX.1 User level macro

\msdata The user level `\msdata` command only writes the manuscripts data in numbered auxiliary file.

```

6834 \newcommand{\msdata}[1]{%
6835 \leavevmode%
6836 \unless\ifstopmsdata@inserted@%
6837 \stopmsdata%
6838 \led@warning@msdatawithoutstop%
6839 \fi%
6840 \global\stopmsdata@inserted@false%
6841 \unless\ifledRcol%
6842 \protected@write\linenum@out{}{%
6843 \string\@msd{#1}%
6844 }%
6845 \else%
6846 \protected@write\linenum@outR{}{%

```

```

6847 \string\@msd{#1}%
6848 }%
6849 \fi%
6850 }%
6851 %

```

\stopmsdata The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

6852 \newcommand{\stopmsdata}[0]{%
6853 \leavevmode%
6854 \unless\ifledRcol%
6855 \protected@write\linenum@out{}{%
6856 \string\@stopmsd%
6857 }%
6858 \else%
6859 \protected@write\linenum@outR{}{%
6860 \string\@stopmsd%
6861 }%
6862 \fi%
6863 \global\stopmsdata@inserted@true%
6864 }%
6865 %

```

\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`

```

6866 \newif\ifstopmsdata@inserted@%
6867 %

```

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?

```

6868 \newcommand{\setmsdataseries}[1]{%
6869 \gdef\@msdata@series{#1}%
6870 }%
6871 \def\@msdata@series{A}%
6872 %

```

\setmsdatalabel The label for the manuscripts data.

```

6873 \def\ms@data@label{Ms.}%
6874 \newcommand{\setmsdatalabel}[1]{%
6875 \gdef\ms@data@label{#1}%
6876 }%
6877 %

```

XXIX.3 Counters and lists

\@msd@c \@msd@c is a counter incremented at each \@msd read in auxiliary file.

```
6878 \numdef{\@msd@c}{0}
6879 \numdef{\@msd@cR}{0}
6880 %
```

\@msd@c \@add@msd@c is a counter incremented at each \@add@msddata, that is at each time we prepare the insertion of manuscripts data footnote.

```
6881 \numdef{\@add@msd@c}{0}%
6882 \numdef{\@add@msd@cR}{0}%
6883 %
```

\@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.

```
6884 \def\@msdata@list{}%
6885 %
```

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.

```
6886 \newcommand{\@msd}[1]{%
6887   \unless\ifledRcol%
6888     \numdef{\@msd@c}{\@msd@c+\@ne}%
6889     \csgdef{\@msdata@\@msd@c @data}{#1}%
6890     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
6891     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
6892     \xright@appenditem{\the\absline@num}\to\actionlines@list%
6893     \xright@appenditem{-1010}\to\actions@list%
6894   \else%
6895     \numdef{\@msd@cR}{\@msd@cR+\@ne}%
6896     \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
6897     \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
6898     \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
6899     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
6900     \xright@appenditem{-1010}\to\actions@listR%
6901   \fi%
6902 }%
6903 %
```

\@endmsd Inserted in the auxiliary file by \@stopmsd, the \@stopmsd macro will store in two commands the line number and the absolute line number on which it is called.

```

6904 \newcommand{\@stopmsd}[0]{%
6905   \unless\ifledRcol%
6906     \ifcsundef{@msdata@\@msd@c @stoplinenumber}{%
6907       \csxdef{@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
6908       \csxdef{@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
6909     }{}%
6910   \else%
6911     \ifcsundef{@msdata@\@msd@cR @stoplinenumberR}{%
6912       \csxdef{@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
6913       \csxdef{@msdata@\@msd@cR @stoplinenumberR}{\the\line@numR}%
6914     }%
6915   }{}%
6916   \fi%
6917 }%
6918 %

```

XXIX.5 Action macro

\add@msdata \add@msdata is executed on each line when action code 1010 is seen. It will not insert immediately the manuscript data footnote, as action code are executed before the line be typeset, and, consequently, could be on the previous page. So it just store the manuscript data footnote to \@msdata@list.

```

6919 \newcommand{\add@msdata}{%
6920   \bgroup%
6921   \normalfont%
6922   \unless\ifledRcol%
6923     \numgdef{\add@msd@c}{\add@msd@c+\@ne}%
6924     \ifcsdef{@msdata@\add@msd@c @data}{%
6925       \letcs{\@data}{@msdata@\add@msd@c @data}%
6926       \edef\l@d@nums{%
6927         000| % Start page = we don't print it
6928         \csuse{@msdata@\add@msd@c @linenumber}| % Start line number
6929         000| % Start subline number, for now, not used
6930         000| % End page number, we don't print it
6931         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
6932         {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
6933         {\csuse{@lastline@forpage@\the\page@num}}}%Otherwiser, last
number of the page
6934         |%
6935         000| % End sub line number, for now, not used
6936         \edfont@info%Font
6937       }%
6938       \@msd@options@fullpagefalse%
6939       \if@firstlineofpage%Try if the data are for the full page. If yes
, will add options to the list.
6940       \unless\if@msdata@insertedfrompreviouspage%

```

```

6941         \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse
        {\@msdata@add@msd@c @stopabslinenumber}+\@ne}%
6942         {%
6943         \numdef{\@tmp}{\add@msd@c+\@ne}%
6944         \ifcsdef{@msdata@\@tmp @abslinenumber}%
6945         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
        @lastabsline@forpage@the\page@num}}}%
6946         {}%
6947         {\@msd@options@fullpagetrue}%
6948         }%
6949         {\@msd@options@fullpagetrue}%
6950         }%
6951         {}%
6952         \fi%
6953         \fi%
6954         \listxadd{\@msdata@list}{%
6955         \@msd@options@iffullpage%
6956         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
        expandonce\l@d@nums}{\msdata@label}{\expandonce\data}}%
6957         \reset@msd@options@iffullpage%
6958         }%
6959         }%
6960         {}%
6961         \else%
6962         \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
6963         \ifcsdef{@msdata@\add@msd@cR @dataR}{%
6964         \letcs{\@data}{@msdata@\add@msd@cR @dataR}%
6965         \edef\l@d@nums{%
6966         000| Start page = we don't print it
6967         \csuse{@msdata@\add@msd@cR @linenumberR}| Start line number
6968         000| Start subline number, for now, not used
6969         000| End page number, we don't print it
6970         \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
        csuse{@lastline@forpageR@the\page@numR}}%
6971         {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}%End line number
        if in the same page
6972         {\csuse{@lastline@forpageR@the\page@numR}}%Otherwise, last
        number of the page
6973         |%
6974         000| End sub line number, for now, not used
6975         \edfont@info%Font
6976         }%
6977         \@msd@options@fullpagefalse%
6978         \if@firstlineofpageR%
6979         \unless\if@msdata@insertedfrompreviouspage%
6980         \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
        csuse{@msdata@\add@msd@c @stopabslinenumberR}+\@ne}%
6981         {%
6982         \numdef{\@tmp}{\add@msd@cR+\@ne}%
6983         \ifcsdef{@msdata@\@tmp @abslinenumberR}%

```

```

6984         {\ifnumequal{\csuse{@msdata@\tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}}%
6985         {}%
6986         {\@msd@options@fullpagetrue}%
6987     }%
6988     {\@msd@options@fullpagetrue}%
6989     }%
6990     {}%
6991     \fi%
6992     \fi%
6993     \listxadd{\@msdata@list}{%
6994         \@msd@options@iffullpage%
6995         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
6996         \reset@msd@options@iffullpage%
6997     }%
6998     }%
6999     {}%
7000     \fi%
7001     \egroup%
7002 }%
7003 %

```

[\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.

```

7004 \newif\if@msdata@insertedfrompreviouspage%
7005 %

```

[\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.

```

7006 \newcommand{\add@msdata@firstlineofpage}{%
7007     \bgroup%
7008     \normalfont%
7009     \unless\ifledRcol{%
7010         \ifcsdef{@msdata@\add@msd@c @data}{%
7011             \ifnumless{\the\absline@num-\@one}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}}%
7012         {%
7013             \global\@msdata@insertedfrompreviouspagetrue%
7014             \letcs{\@data}{@msdata@\add@msd@c @data}%
7015             \edef\l@d@nums{%
7016                 000|% Start page = we don't print it
7017                 \numexpr\the\line@num+\@one\relax|% Start line number = first line
of the page. As \add@msdata@firstlineofpage is called before line number
has been incremented, we increment it for printing
7018                 000|% Start subline number, for now, not used
7019                 000|% End page number, we don't print it

```

```

7020 \ifnumless{\csuse{@msdata@add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}%
7021 {\csuse{@msdata@add@msd@c @stoplinenumber}}%End line number if
in the same page
7022 {\csuse{@lastline@forpage@the\page@num}}%Otherwise, last
number of the page
7023 |%
7024 000|% End sub line number, for now, not used
7025 \edfont@info%Font
7026 }%
7027 \msd@options@fullpagefalse%
7028 \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
7029 {%
7030 \numdef{\@tmp}{\add@msd@c+\@ne}%
7031 \ifcsdef{@msdata@\@tmp @abslinenumber}%
7032 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}%
7033 }%
7034 {\msd@options@fullpagetrue}%
7035 }%
7036 {\msd@options@fullpagetrue}%
7037 }%
7038 {}%
7039 \listxadd{\@msdata@list}{%
7040 \msd@options@iffullpage%
7041 \noexpand\csuse{v\@msdata@series footnote}}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
7042 \reset@msd@options@iffullpage%
7043 }%
7044 }%
7045 {\global\@msdata@insertedfrompreviouspagefalse}%
7046 }{}%
7047 \else%
7048 \ifcsdef{@msdata@add@msd@cR @dataR}{%
7049 \ifnumless{the\absline@numR-\@ne}{\csuse{@msdata@add@msd@cR
@stopabslinenumberR}}%
7050 {%
7051 \global\@msdata@insertedfrompreviouspagetrue%
7052 \letcs{\@data}{\@msdata@add@msd@cR @dataR}%
7053 \edef\l@d@nums{%
7054 000|% Start page = we don't print it
7055 \numexprthe\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
7056 000|% Start subline number, for now, not used
7057 000|% End page number, we don't print it
7058 \ifnumless{\csuse{@msdata@add@msd@cR @stopabslinenumberR}}{\csuse{@lastline@forpageR@the\page@numR}}%

```

```

7059         {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
7060         {\csuse{@lastline@forpageR@the\page@numR}}}%Otherwise, last
number of the page
7061         |%
7062         000|% End sub line number, for now, not used
7063         \edfont@info%Font
7064         }%
7065         \@msd@options@fullpagefalse%
7066         \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@ne}%
7067         {%
7068         \numdef{\@tmp}{\add@msd@cR+\@ne}%
7069         \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7070         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
7071         }%
7072         {\@msd@options@fullpagetrue}%
7073         }%
7074         {\@msd@options@fullpagetrue}%
7075         }%
7076         {}%
7077         \listxadd{\@msdata@list}{%
7078         \@msd@options@iffullpage%
7079         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}}%
7080         \reset@msd@options@iffullpage%
7081         }%
7082         }%
7083         {\global\@msdata@insertedfrompreviouspagefalse}%
7084         }{}%
7085         \fi%
7086         \egroup%
7087     }%
7088     %

```

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@msdata89 \newcommand{\insert@msdata}{%
7090     \def\do##1{##1}%
7091     \dolistloop{\@msdata@list}%
7092     \global\let\@msdata@list\relax%
7093 }%
7094 %

```

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.

```

7095 \newif\if@msd@options@fullpage%
7096 \newcommand{\@msd@options@iffullpage}[0]{%
7097   \if@msd@options@fullpage%
7098     \noexpand\toggletrue{nonum@}%
7099     \ifdefvoid{\ms@data@label}%
7100       {\noexpand\toggletrue{nosep@}}%
7101       {}%
7102   \fi%
7103 }%
7104 \newcommand{\reset@msd@options@iffullpage}[0]{%
7105   \noexpand\togglefalse{nonum@}%
7106   \noexpand\togglefalse{nosep@}%
7107 }%
7108 %

```

XXX Arrays and tables

XXX.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread ‘`eeq` and `amstex`’, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsngen` package.

```

7109 \newtoks\@emptytoks
7110
7111 %

```

The rest is from `amsmath`.

`\l@denvbody` A token register to contain the body.

```

7112 \newtoks\l@denvbody
7113
7114 %

```

`\addtol@denvbody` `\addtol@denvbody{arg}` adds `arg` to the token register `\l@denvbody`.

```

7115 \newcommand{\addtol@denvbody}[1]{%
7116   \global\l@denvbody\expandafter{\the\l@denvbody#1}}
7117
7118 %

```

`\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`.

```

7119 \newcommand{\l@dcollect@body}[1]{%
7120   \l@denvbody{\expandafter#1\expandafter{\the\l@denvbody}}%
7121   \edef\processl@denvbody{\the\l@denvbody\noexpand\end{\@currenenv}}%
7122   \l@denvbody\@emptytoks \def\l@dbegin@stack{b}%
7123   \begingroup
7124     \expandafter\let\csname\@currenenv\endcsname\l@dcollect@@body
7125     \edef\processl@denvbody{\expandafter\noexpand\csname\@currenenv\endcsname}%
7126     \processl@denvbody%
7127   }%
7128
7129 %

```

`\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.

```

7130 \def\l@dpush@begins#1\begin#2{%
7131   \ifx\end#2else b\expandafter\l@dpush@begins\fi}
7132
7133 %

```

`\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.

```

7134 \def\l@dcollect@@body#1\end#2{%
7135   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
7136     \expandafter\@gobble\l@dbegin@stack}%
7137   \ifx\@empty\l@dbegin@stack
7138     \endgroup
7139     \@checkend{#2}%
7140     \addtol@denvbody{#1}%
7141   \else
7142     \addtol@denvbody{#1\end{#2}}%
7143   \fi
7144   \processl@denvbody % A little tricky! Note the grouping
7145 }
7146
7147 %

```

There was a question on CTT about how to use `\collect@body` for a macro taking an argument. The following is part of that thread.

```
From: Heiko Oberdiek <oberdiek@uni-freiburg.de>
Newsgroups: comp.text.tex
Subject: Re: Using \collect@body with commands that take >1 argument
Date: Fri, 08 Aug 2003 09:03:20 +0200
```

```
eed132@psu.edu (Evan) wrote:
> I'm trying to make a new Latex environment that acts like the>
> \colorbox command that is part of the color package. I looked through
> the FAQ and ran across this bit about using the \collect@body command
> that is part of AMSLaTeX:
> http://www.tex.ac.uk/cgi-bin/texfaq2html?label=cmdasenv
>
> It almost works. If I do something like the following:
> \newcommand{\redbox}[1]{\colorbox{red}{#1}}
>
> \makeatletter
> \newenvironment{redbox}{\collect@body \redbox}{}
```

You will get an error message: Command `\redbox` already defined.
Thus you must rename either the command `\redbox` or the environment name.

```
> \begin{coloredbox}{blue}
> Yadda yadda yadda... this is on a blue background...
> \end{coloredbox}
> and can't figure out how to make the \collect@body take this.

> \collect@body \colorbox{red}
> \collect@body {\colorbox{red}}
```

The argument of `\collect@body` has to be one token exactly.

```
\documentclass{article}
\usepackage{color}
\usepackage{amsmath}

\newcommand{\redbox}[1]{\colorbox{red}{#1}}
\makeatletter
\newenvironment{coloredbox}[1]{%
  \def\next@{\colorbox{#1}}%
  \collect@body\next@
}{%

% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}[1]{%
  \def\next@{\mycoloredbox{#1}}%
  \collect@body\next@
```

```

}{}
\newcommand{\mycoloredbox}[2]{%
  \colorbox{#1}{\ignorespaces#2\unskip}%
}

% support of optional color model argument
\newcommand\coloredboxIII\endcsname{}
\def\coloredboxIII#1#{%
  \@coloredboxIII{#1}%
}
\def\@coloredboxIII#1#2{%
  \def\next@{\mycoloredboxIII{#1}{#2}}%
  \collect@body\next@
}
\newcommand{\mycoloredboxIII}[3]{%
  \colorbox{#1}{#2}{\ignorespaces#3\unskip}%
}

\makeatother

\begin{document}
  Black text before
  \begin{coloredbox}{blue}
    Hello World
  \end{coloredbox}
  Black text after

  Black text before
  \begin{coloredboxII}{blue}
    Hello World
  \end{coloredboxII}
  Black text after

  Black text before
  \begin{coloredboxIII}[rgb]{0,0,1}
    Hello World
  \end{coloredboxIII}
  Black text after

\end{document}

Yours sincerely
Heiko <oberdiek@uni-freiburg.de>

```

XXX.2 Tabular environments

This is based on the work by Herbert Breger in developing `tabmac.tex`.

The original `tabmac.tex` file was void of comments or any explanatory text other

than the above notice. The algorithm is Breger's. Peter Wilson have made some cosmetic changes to the original code and reimplemented some things so they are more LaTeX-like. All the commentary are from Peter Wilson, as are any mistake or errors.

However, Maïeul Rouquette has modified code in order to add new features of `eledmac` and `reledmac`.

XXX.2.1 Disabling and restoring commands

`\l@dtabnoexpands` More no expansion for critical and familiar footnotes in tabular environment.

```

7148 \newcommand*{\l@dtabnoexpands}{%
7149   \let\rtab=0%
7150   \let\ctab=0%
7151   \let\ltab=0%
7152   \let\rtabtext=0%
7153   \let\ltabtext=0%
7154   \let\ctabtext=0%
7155   \let\edbeforetab=0%
7156   \let\edaftertab=0%
7157   \let\edatleft=0%
7158   \let\edatright=0%
7159   \let\edvertline=0%
7160   \let\edvertdots=0%
7161   \let\edrowfill=0%
7162 }
7163
7164 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in `edtabularx` and `edarrayx` environments.

`\restore@familiarnotes`

```

7165 \newcommand{\disable@familiarnotes}{%
7166   \unless\ifnofamiliar%
7167     \def\do##1{%
7168       \csletcs{footnote@@##1}{footnote##1}%
7169       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
7170         \protected@csxdef{theftnmark##1}{\csuse{thefootnote##1}}%
7171         \csuse{@footnotemark##1}%
7172       }%
7173     }%
7174     \dolistloop{\@series}%
7175   \fi%
7176 }%
7177 \newcommand{\restore@familiarnotes}{%
7178   \unless\ifnofamiliar%
7179     \def\do##1{%
7180       \csletcs{footnote##1}{footnote@@##1}%
7181     }%
7182     \dolistloop{\@series}%
7183   \fi%

```

```

7184 }%
7185
7186 %

```

\disable@sidenotes The same, for side notes.

\restore@sidenotes

```

7187 \newcommand{\disable@sidenotes}{%
7188   \let\@@ledrightnote\ledrightnote%
7189   \let\@@ledleftnote\ledleftnote%
7190   \let\@@ledsidenote\ledsidenote%
7191   \let\ledrightnote@gobble%
7192   \let\ledleftnote@gobble%
7193   \let\ledsidenote@gobble%
7194 }%
7195 \newcommand{\restore@sidenotes}{%
7196   \let\ledrightnote\@@ledrightnote%
7197   \let\ledleftnote\@@ledleftnote%
7198   \let\ledsidenote\@@ledsidenote%
7199 }%
7200 %

```

\disable@notes Disable/restore side and familiar notes.

\restore@notes

```

7201 \newcommand{\disable@notes}{%
7202   \disable@sidenotes%
7203   \disable@familiarnotes%
7204 }%
7205 \newcommand{\restore@notes}{%
7206   \restore@sidenotes%
7207   \restore@familiarnotes%
7208 }%
7209 %

```

\EDTEXT We need to be able to modify the `\edtext` macros and also restore their original definitions.

\xedtext

```

7210 \let\EDTEXT=\edtext
7211 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
7212 %

```

\EDLABEL We need to be able to modify and restore the `\edlabel` macro.

\xedlabel

```

7213 \let\EDLABEL=\edlabel
7214 \newcommand*\xedlabel[1]{\EDLABEL{#1}}
7215 %

```

\EDINDEX Macros supporting modification and restoration of `\edindex`.

\xedindex

\nulledindex

```

7216 \let\EDINDEX=\edindex
7217 \newcommand{\xedindex}{\@bsphack%
7218   \ifnextchar [{\l@d@index}{\l@d@index[\jobname]}}
7219 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
7220
7221 %

```

\@line@num Macro supporting restoration of \linenum.

```

7222 \let\@line@num=\linenum
7223 %

```

\l@gobblearg `\l@gobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```

7224 \newcommand*\l@gobbleoptarg}[2][\relax]%
7225
7226 %

```

\Relax₂₇ `\let\Relax=\relax`

\NEXT₂₈ `\let\NEXT=\next`

```

7229
7230 %

```

\l@dmodforedtext Modify and restore various macros for when \edtext is used.
\l@drestoreforedtext

```

7231 \newcommand{\l@dmodforedtext}{%
7232   \let\edtext\relax
7233   \def\do##1{\global\csletcs{##1footnote}{\l@gobbleoptarg}}%
7234   \dolistloop{\@series}%
7235   \let\edindex\nulledindex
7236   \let\linenum\@gobble}
7237 \newcommand{\l@drestoreforedtext}{%
7238   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
7239   \dolistloop{\@series}%
7240   \let\edindex\xedindex}
7241 %

```

\l@dnullfills Nullify and restore some column fillers, etc.
\l@drestorefills

```

7242 \newcommand{\l@dnullfills}{%
7243   \def\edlabel##1{}%
7244   \def\edrowfill##1##2##3{}%
7245 }
7246 \newcommand{\l@drestorefills}{%
7247   \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
7248 }
7249
7250 %

```

`\letsforverteilen` Gathers some lets and other code that is common to the `*verteilen*` macros.

```

7251 \newcommand{\letsforverteilen}{%
7252   \let\edtext\xedtext
7253   \let\edindex\xedindex
7254   \def\do##1{\global\csletcs{##1footnote}{##1@@footnote}}
7255   \dolistloop{@series}%
7256   \let\linenum@line@num
7257   \hilfsskip=\l@dcwidth%
7258   \advance\hilfsskip by -\wd\hilfsbox
7259   \def\edlabel##1{\xedlabel{##1}}
7260
7261 %

```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for
`\enablel@dtabfeet` `\edtext`.

```

7262 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
7263 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
7264 %

```

XXX.2.2 Counters, boxes and lengths

`\l@dampcount` `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter
`\l@dcolcount` for the columns.

```

7265 \newcount\l@dampcount
7266 \l@dampcount=1\relax
7267 \newcount\l@dcolcount
7268 \l@dcolcount=0\relax
7269
7270 %

```

`\hilfsbox` Some (temporary) helper items.

```

7271 \hilfsskip
7272 \Hilfsbox
7273 \hilfscount
7274
7275 %
7276

```

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).

```

7277 \newdimen\dcoli
7278 \newdimen\dcolii
7279 \newdimen\dcoliii
7280 \newdimen\dcoliv
7281 \newdimen\dcolv

```

```

7282 \newdimen\dc colvi
7283 \newdimen\dc colvii
7284 \newdimen\dc colviii
7285 \newdimen\dc colix
7286 \newdimen\dc colx
7287 \newdimen\dc colxi
7288 \newdimen\dc colxii
7289 \newdimen\dc colxiii
7290 \newdimen\dc colxiv
7291 \newdimen\dc colxv
7292 \newdimen\dc colxvi
7293 \newdimen\dc colxvii
7294 \newdimen\dc colxviii
7295 \newdimen\dc colxix
7296 \newdimen\dc colxx
7297 \newdimen\dc colxxi
7298 \newdimen\dc colxxii
7299 \newdimen\dc colxxiii
7300 \newdimen\dc colxxiv
7301 \newdimen\dc colxxv
7302 \newdimen\dc colxxvi
7303 \newdimen\dc colxxvii
7304 \newdimen\dc colxxviii
7305 \newdimen\dc colxxix
7306 \newdimen\dc colxxx
7307 \newdimen\dc colerr % added for error handling
7308
7309 %

```

\l@dc colwidth This is a cunning way of storing the columnwidths indexed by the column number \l@dc colcount, like an array. (was \Dimenzuordnung)

```

7310 \newcommand{\l@dc colwidth}{\ifcase \the\l@dc colcount \dc oli %???
7311 \or \dc oli \or \dc olii \or \dc oliii
7312 \or \dc oliv \or \dc olv \or \dc olvi
7313 \or \dc olvii \or \dc olviii \or \dc olvix \or \dc olx
7314 \or \dc olxi \or \dc olxii \or \dc olxiii
7315 \or \dc olxiv \or \dc olxv \or \dc olxvi
7316 \or \dc olxvii \or \dc olxviii \or \dc olxix \or \dc olxx
7317 \or \dc olxxi \or \dc olxxii \or \dc olxxiii
7318 \or \dc olxxiv \or \dc olxxv \or \dc olxxvi
7319 \or \dc olxxvii \or \dc olxxviii \or \dc olxxix \or \dc olxxx
7320 \else \dc olerr \fi}
7321
7322 %

```

\step1@dc colcount This increments the column counter, and issues an error message if it is too large.

```

7323 \newcommand*{\step1@dc colcount}{\advance\l@dc colcount\@ne
7324 \ifnum\l@dc colcount>30\relax

```

```

7325 \led@err@TooManyColumns
7326 \fi}
7327
7328 %

```

\l@setmaxcolwidth Sets the column width to the maximum value seen so far.

```

7329 \newcommand{\l@setmaxcolwidth}{%
7330 \ifdim\l@dcwidth < \wd\hilfsbox
7331 \l@dcwidth = \wd\hilfsbox
7332 \else \relax \fi}
7333
7334 %

```

\measurecell Measure (recursively) the width required for a math cell.

```

7335 \def\measurecell #1{%
7336 \ifx #1\ \ifnum\l@dcwidth=0\let\NEXT\relax%
7337 \else\l@dcheckcols%
7338 \l@dcwidth=0%
7339 \let\NEXT\measurecell%
7340 \fi%
7341 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7342 \step\l@dcwidth%
7343 \l@setmaxcolwidth%
7344 \let\NEXT\measurecell%
7345 \fi\NEXT}
7346
7347 %

```

\measuretcell Measure (recursively) the width required for a text cell.

```

7348 \def\measuretcell #1{%
7349 \ifx #1\ \ifnum\l@dcwidth=0\let\NEXT\relax%
7350 \else\l@dcheckcols%
7351 \l@dcwidth=0%
7352 \let\NEXT\measuretcell%
7353 \fi%
7354 \else\setbox\hilfsbox=\hbox{#1}%
7355 \step\l@dcwidth%
7356 \l@setmaxcolwidth%
7357 \let\NEXT\measuretcell%
7358 \fi\NEXT}
7359
7360 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

7361 \def\measuremrow #1{%
7362 \ifx #1\ \let\NEXT\relax%

```

```

7363 \else\measuremcell #1&\\&\\&%
7364 \let\NEXT\measuremrow%
7365 \fi\NEXT}
7366 %

```

\measurerow Measure (recursively) the width required for a text row.

```

7367 \def\measurerow #1\\{%
7368 \ifx #1&\let\NEXT\relax%
7369 \else\measuremcell #1&\\&\\&%
7370 \let\NEXT\measurerow%
7371 \fi\NEXT}
7372
7373 %

```

\edtabcolsep The length \edtabcolsep controls the distance between columns.

```

7374 \newskip\edtabcolsep
7375 \global\edtabcolsep=10pt
7376
7377 %

```

```

\variab78 \newcommand{\variab}{\relax}

```

```

7379
7380 %

```

\l@dcheckcols Check that the number of columns is consistent.

```

7381 \newcommand*{\l@dcheckcols}{%
7382 \ifnum\l@dcolcount=1\relax
7383 \else
7384 \ifnum\l@dampcount=1\relax
7385 \else
7386 \ifnum\l@dcolcount=\l@dampcount\relax
7387 \else
7388 \l@d@err@UnequalColumns
7389 \fi
7390 \fi
7391 \l@dampcount=\l@dcolcount
7392 \fi}
7393
7394 %

```

\edfilldimen A length.

```

7395 \newdimen\edfilldimen
7396 \edfilldimen=0pt
7397
7398 %

```

`\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`

```
7399 \newcounter{addcolcount}
7400 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
7401 %
```

XXX.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.

```
7402 \def\setmcellright #1{\def\edlabel##1{}%
7403   \let\edindex\nulledindex
7404   \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
7405     \let\Next\relax%
7406     \else\l@dcolcount=0%
7407       \let\Next=\setmcellright%
7408     \fi%
7409   \else%
7410     \disablel@dtabfeet%
7411     \step1@dcolcount%
7412     \disable@notes%
7413     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7414     \restore@notes%
7415     \letsforverteilen%
7416     \hskip\hilfsskip$\displaystyle{#1}$%
7417     \hskip\edtabcolsep%
7418     \let\Next=\setmcellright%
7419   \fi\Next}
7420 %
7421 %
```

`\settcclright` Typeset (recursively) cells of text right justified.

```
7422 \def\settcclright #1{\def\edlabel##1{}%
7423   \let\edindex\nulledindex
7424   \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
7425     \let\Next\relax%
7426     \else\l@dcolcount=0%
7427       \let\Next=\settcclright%
7428     \fi%
7429   \else%
7430     \disablel@dtabfeet%
7431     \step1@dcolcount%
7432     \disable@notes%
7433     \setbox\hilfsbox=\hbox{#1}%
7434     \restore@notes%
7435     \letsforverteilen%
7436     \hskip\hilfsskip#1%
```

```

7437         \hskip\edtabcolsep%
7438         \let\Next=\settccllright%
7439     \fi\Next}
7440 %

```

\setmcellleft Typeset (recursively) cells of display math left justified.

```

7441 \def\setmcellleft #1&{\def\edlabel##1{}%
7442     \let\edindex\nulledindex
7443     \ifx #1\\ \ifnum\l@dc@colcount=0 \let\Next\relax%
7444         \else\l@dc@colcount=0%
7445         \let\Next=\setmcellleft%
7446         \fi%
7447     \else \disablel@dtabfeet%
7448         \stepl@dc@colcount%
7449         \disable@notes%
7450         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7451         \restore@notes%
7452         \letsforverteilen%
7453         $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
7454         \let\Next=\setmcellleft%
7455     \fi\Next}
7456 %
7457 %

```

\settcclleft Typeset (recursively) cells of text left justified.

```

7458 \def\settcclleft #1&{\def\edlabel##1{}%
7459     \let\edindex\nulledindex
7460     \ifx #1\\ \ifnum\l@dc@colcount=0 \let\Next\relax%
7461         \else\l@dc@colcount=0%
7462         \let\Next=\settcclleft%
7463         \fi%
7464     \else \disablel@dtabfeet%
7465         \stepl@dc@colcount%
7466         \disable@notes%
7467         \setbox\hilfsbox=\hbox{#1}%
7468         \restore@notes%
7469         \letsforverteilen%
7470         #1\hskip\hilfsskip\hskip\edtabcolsep%
7471         \let\Next=\settcclleft%
7472     \fi\Next}
7473 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

7474 \def\setmcellcenter #1&{\def\edlabel##1{}%
7475     \let\edindex\nulledindex
7476     \ifx #1\\ \ifnum\l@dc@colcount=0\let\Next\relax%
7477         \else\l@dc@colcount=0%

```

```

7478             \let\Next=\setmcellcenter%
7479         \fi%
7480     \else     \disablel@dtabfeet%
7481             \stepl@dcolcount%
7482             \disable@notes%
7483             \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7484             \restore@notes%
7485             \letsforverteilen%
7486             \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
7487             \hskip\edtabcolsep%
7488             \let\Next=\setmcellcenter%
7489     \fi\Next}
7490
7491 %

```

\settcellcenter Typeset (recursively) cells of text centered.

```

7492 \def\settcellcenter #1&{\def\edlabel##1{}%
7493     \let\edindex\nulledindex
7494     \ifx #1\\ \ifnum\l@dcolcount=0 \let\Next\relax%
7495         \else\l@dcolcount=0%
7496             \let\Next=\settcellcenter%
7497         \fi%
7498     \else     \disablel@dtabfeet%
7499             \stepl@dcolcount%
7500             \disable@notes%
7501             \setbox\hilfsbox=\hbox{#1}%
7502             \restore@notes%
7503             \letsforverteilen%
7504             \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
7505             \hskip\edtabcolsep%
7506             \let\Next=\settcellcenter%
7507     \fi\Next}
7508
7509 %

```

\NEXT₁₀ \let\NEXT=\relax

```

7511
7512 %

```

\setmrowright Typeset (recursively) rows of right justified math.

```

7513 \def\setmrowright #1\\{%
7514     \ifx #1& \let\NEXT\relax
7515     \else \centerline{\setmcellright #1&\\&\\&}
7516         \let\NEXT=\setmrowright
7517     \fi\NEXT}
7518 %

```

\settroright Typeset (recursively) rows of right justified text.

```

7519 \def\settroright #1\\{%
7520     \ifx #1& \let\NEXT\relax
7521     \else \centerline{\settcclright #1&\\&\\&}
7522         \let\NEXT=\settroright
7523     \fi\NEXT}
7524
7525 %

```

\setmrowleft Typeset (recursively) rows of left justified math.

```

7526 \def\setmrowleft #1\\{%
7527     \ifx #1& \let\NEXT\relax
7528     \else \centerline{\setmcclleft #1&\\&\\&}
7529         \let\NEXT=\setmrowleft
7530     \fi\NEXT}
7531 %

```

\settrorleft Typeset (recursively) rows of left justified text.

```

7532 \def\settrorleft #1\\{%
7533     \ifx #1& \let\NEXT\relax
7534     \else \centerline{\settcclleft #1&\\&\\&}
7535         \let\NEXT=\settrorleft
7536     \fi\NEXT}
7537
7538 %

```

\setmrowcenter Typeset (recursively) rows of centered math.

```

7539 \def\setmrowcenter #1\\{%
7540     \ifx #1& \let\NEXT\relax%
7541     \else \centerline{\setmcclcenter #1&\\&\\&}
7542         \let\NEXT=\setmrowcenter
7543     \fi\NEXT}
7544 %

```

\settrorcenter Typeset (recursively) rows of centered text.

```

7545 \def\settrorcenter #1\\{%
7546     \ifx #1& \let\NEXT\relax
7547     \else \centerline{\settcclcenter #1&\\&\\&}
7548         \let\NEXT=\settrorcenter
7549     \fi\NEXT}
7550
7551 %

```

```

\newcommand{\nullsetzen}{%
  \step1@dcolcount%
  \l@dcolwidth=0pt%
  \ifnum\l@dcolcount=30\let\NEXT\relax%
    \l@dcolcount=0\relax
  \else\let\NEXT\nullsetzen%
  \fi\NEXT}
%
```

\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.

```

\newcommand{\edatleft}[3][\@empty]{%
  \ifx#1\@empty
    \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
      depth 0pt \right. $\hss}\vfil}
  \else
    \vbox to 4pt{\vss\hbox{#1\left#2\vrule width0pt height #3
      depth 0pt \right. $\vfil}
  \fi}
%
```

\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.

```

\newcommand{\edatright}[3][\@empty]{%
  \ifx#1\@empty
    \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
      depth 0pt \right#2 $\hss}\vfil}
  \else
    \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
      depth 0pt \right#2 #1 $\vfil}
  \fi}
%
```

\edvertline `\edvertline{ $\langle len \rangle$ }` vertical line $\langle len \rangle$ high.

```

\newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
%
```

\edvertdots `\edvertdots{ $\langle len \rangle$ }` vertical dotted line $\langle len \rangle$ high.

```

\newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
  {\cleaders\hbox{$\math\hbox{.}\vbox to 0.5em{ }\vfil}}}}
%
```

\l@dtabaddcols `\l@dtabaddcols{<startcol>}{<endcol>}` adds the widths of the columns `<startcol>` through `<endcol>` to `\edfilldimen`. It is a \TeX style reimplementation of the original `\@add@`.

```

7587 \newcommand{\l@dtabaddcols}[2]{%
7588   \l@dcheckstartend{#1}{#2}%
7589   \ifl@dstartendok
7590   \setcounter{addcolcount}{#1}%
7591   \@whilenum \value{addcolcount}<#2\relax \do
7592   {\advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
7593    \advance\edfilldimen by \edtabcolsep
7594    \stepcounter{addcolcount}}%
7595   \advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
7596   \fi
7597 }
7598
7599 %

```

\ifl@dstartendok `\l@dcheckstartend{<startcol>}{<endcol>}` checks that the values of `<startcol>` and `<endcol>` are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```

7600 \newif\ifl@dstartendok
7601 \newcommand{\l@dcheckstartend}[2]{%
7602   \l@dstartendoktrue
7603   \ifnum #1<\@ne
7604     \l@dstartendokfalse
7605     \led@err@LowStartColumn
7606   \fi
7607   \ifnum #2>30\relax
7608     \l@dstartendokfalse
7609     \led@err@HighEndColumn
7610   \fi
7611   \ifnum #1>#2\relax
7612     \l@dstartendokfalse
7613     \led@err@ReverseColumns
7614   \fi
7615 }
7616
7617 %

```

\edrowfill `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a \TeX style reimplementation and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktel` macros.

```

7618 \newcommand*{\edrowfill}[3]{%
7619   \l@dtabaddcols{#1}{#2}%
7620   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
7621 \let\@edrowfill@=\edrowfill
7622 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}

```

7623
7624 %

\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`.

```
7625 \newcommand{\leftltab}[1]{%
7626   \hb@xt@0\z@{\vbox{\edtabindent%
7627     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
7628
7629 %
```

\leftrtab `\leftrtab{⟨text⟩}{⟨math⟩}` for `\edbeforetab` in `\rtab`.

```
7630 \newcommand{\leftrtab}[2]{%
7631   #2\hb@xt@0\z@{\vbox{\edtabindent%
7632     \advance\Hilfsskip by\dcoli%
7633     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
7634
7635 %
```

\leftctab `\leftctab{⟨text⟩}{⟨math⟩}` for `\edbeforetab` in `\ctab`.

```
7636 \newcommand{\leftctab}[2]{%
7637   \hb@xt@0\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
7638     \advance\Hilfsskip by 0.5\dcoli%
7639     \setbox\hilfsbox=\hbox{\def\edlabel##1}%
7640     \disablel@dtabfeet$\displaystyle{#2}$}%
7641     \advance\Hilfsskip by -0.5\wd\hilfsbox%
7642     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
7643   #2}
7644
7645 %
```

\rightctab `\rightctab{⟨math⟩}{⟨text⟩}` for `\edaftertab` in `\ctab`.

```
7646 \newcommand{\rightctab}[2]{%
7647   \setbox\hilfsbox=\hbox{\def\edlabel##1}%
7648   \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
7649   #1\hb@xt@0\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
7650     \advance\Hilfsskip by 0.5\l@dcolwidth%
7651     \advance\Hilfsskip by -\wd\hilfsbox%
7652     \setbox\hilfsbox=\hbox{\def\edlabel##1}%
7653     \disablel@dtabfeet$\displaystyle{#1}$}%
7654     \advance\Hilfsskip by -0.5\wd\hilfsbox%
```

```

7655 \advance\Hilfsskip by \edtabcolsep%
7656 \moveright\Hilfsskip\hbox{ #2}}\hss}%
7657 }
7658
7659 %

```

\rightltab `\rightltab{<math>}<{<text>}` for `\edaftertab` in `\ltab`.

```

7660 \newcommand{\rightltab}[2]{%
7661     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
7662         \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
7663         #1\hb@xt@{\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
7664         \advance\Hilfsskip by\l@dcolwidth%
7665         \advance\Hilfsskip by-\wd\hilfsbox%
7666         \setbox\hilfsbox=\hbox{\def\edlabel##1{%
7667             \disablel@dtabfeet$\displaystyle{#1}$}%
7668             \advance\Hilfsskip by-\wd\hilfsbox%
7669             \advance\Hilfsskip by\edtabcolsep%
7670             \moveright\Hilfsskip\hbox{ #2}}\hss}%
7671         }
7672     }
7673 %

```

\rightrtab `\rightrtab{<math>}<{<text>}` for `\edaftertab` in `\rtab`.

```

7674 \newcommand{\rightrtab}[2]{%
7675     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
7676         \disablel@dtabfeet#2}%
7677         #1\hb@xt@{\z@{\vbox{\edtabindent%
7678         \advance\Hilfsskip by-\wd\hilfsbox%
7679         \advance\Hilfsskip by\edtabcolsep%
7680         \moveright\Hilfsskip\hbox{ #2}}\hss}%
7681     }
7682
7683 %

```

\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.

```

7684 \newcommand{\rtab}[1]{%
7685     \l@dnnullfills
7686     \def\edbeforetab##1##2{\lefttrtab{##1}{##2}}%
7687     \def\edaftertab##1##2{\righttrtab{##1}{##2}}%
7688     \measurebody{#1}%
7689     \l@drestorefills
7690     \variab
7691     \setmrowright #1\\&\\%
7692     \enablel@dtabfeet}
7693
7694 %

```

\measurebody `\measurebody{<body>}` measures the array `<body>`.

```

7695 \newcommand{\measurebody}[1]{%
7696   \disablel@dtabfeet%
7697   \l@dc@colcount=0%
7698   \nullsetzen%
7699   \l@dc@colcount=0
7700   \measuremrow #1\\&\\%
7701   \global\l@dampcount=1}
7702
7703 %

```

\rtabtext `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```

7704 \newcommand{\rtabtext}[1]{%
7705   \l@dnullfills
7706   \measuretbody{#1}%
7707   \l@drestorefills
7708   \variab
7709   \setthrowright #1\\&\\%
7710   \enablel@dtabfeet}
7711
7712 %

```

\measuretbody `\measuretbody{<body>}` measures the tabular `<body>`.

```

7713 \newcommand{\measuretbody}[1]{%
7714   \disable@notes%
7715   \disablel@dtabfeet%
7716   \l@dc@colcount=0%
7717   \nullsetzen%
7718   \l@dc@colcount=0
7719   \measuretbody #1\\&\\%
7720   \restore@notes%
7721   \global\l@dampcount=1}
7722
7723 %

```

\ltab Array with entries left justified.

```

\edbeforetab
\edaftertab
7724 \newcommand{\ltab}[1]{%
7725   \l@dnullfills
7726   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
7727   \def\edaftertab##1##2{\rightltab{##1}{##2}}%
7728   \measuretbody{#1}%
7729   \l@drestorefills
7730   \variab
7731   \setmrowleft #1\\&\\%
7732   \enablel@dtabfeet}
7733
7734 %

```

\ltabtext Tabular with entries left justified.

```

7735 \newcommand{\ltabtext}[1]{%
7736   \l@dnnullfills
7737   \measuretbody{#1}%
7738   \l@drestorefills
7739   \variab
7740   \settrorleft #1\&\%
7741   \enablel@dtabfeet}
7742
7743 %

```

\ctab Array with centered entries.

```

\edbeforetab
\edaftertab
7744 \newcommand{\ctab}[1]{%
7745   \l@dnnullfills
7746   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
7747   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
7748   \measuretbody{#1}%
7749   \l@drestorefills
7750   \variab
7751   \setmrowcenter #1\&\%
7752   \enablel@dtabfeet}
7753
7754 %

```

\ctabtext Tabular with entries centered.

```

7755 \newcommand{\ctabtext}[1]{%
7756   \l@dnnullfills
7757   \measuretbody{#1}%
7758   \l@drestorefills
7759   \variab
7760   \settrorcenter #1\&\%
7761   \enablel@dtabfeet}
7762
7763 %

```

```

\spreadtext64 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
7765   \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
7766 %

```

```

\spreadmath67 \newcommand{\spreadmath}[1]{%
7768   \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
7769
7770 %

```

\HILFSskip More helpers.

\Hilfsskip

```

7771 \newskip\HILFSskip
7772 \newskip\Hilfsskip
7773
7774 %

```

```

\EDTABINDENT \newcommand{\EDTABINDENT}{%
7775
7776   \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
7777   \else\step1@dcolcount%
7778     \advance\Hilfsskip by\l@dcolwidth%
7779     \ifdim\l@dcolwidth=0pt\advance\hilfscount\@ne
7780     \else\advance\Hilfsskip by \the\hilfscount\edtabcolsep%
7781     \hilfscount=1\fi%
7782     \let\NEXT=\EDTABINDENT%
7783   \fi\NEXT}%
7784 %

```

`\edtabindent` (was `\tabindent`)

```

7785 \newcommand{\edtabindent}{%
7786   \l@dcolcount=0\relax
7787   \Hilfsskip=0pt%
7788   \hilfscount=1\relax
7789   \EDTABINDENT%
7790   \hilfsskip=\hsize%
7791   \advance\hilfsskip -\Hilfsskip%
7792   \Hilfsskip=0.5\hilfsskip%
7793 }%
7794
7795 %

```

`\EDTAB` (was `\TAB`)

```

7796 \def\EDTAB #1|#2|{%
7797   \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
7798   \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
7799   \advance\tabelskip -\wd\tabhilfbox%
7800   \advance\tabelskip -\wd\tabHilfbox%
7801   \unhbox\tabhilfbox\hskip\tabelskip%
7802   \unhbox\tabHilfbox}%
7803
7804 %

```

`\EDTABtext` (was `\TABtext`)

```

7805 \def\EDTABtext #1|#2|{%
7806   \setbox\tabhilfbox=\hbox{#1}%
7807   \setbox\tabHilfbox=\hbox{#2}%
7808   \advance\tabelskip -\wd\tabhilfbox%
7809   \advance\tabelskip -\wd\tabHilfbox%

```

```

7810 \unhbox\tabhilfbox\hskip\tabelskip%
7811 \unhbox\tabHilfbox}%
7812 %

```

\tabhilfbox Further helpers.

```

\btabHilfbox
7813 \newbox\tabhilfbox
7814 \newbox\tabHilfbox
7815
7816 %

```

XXX.2.4 Environments

edarrayl edarrayc edarrayr The ‘environment’ forms for `\ltab`, `\ctab` and `\rtab`.

```

7817 \newenvironment{edarrayl}{\l@collect@body\ltab}{\}
7818 \newenvironment{edarrayc}{\l@collect@body\ctab}{\}
7819 \newenvironment{edarrayr}{\l@collect@body\rtab}{\}
7820
7821 %

```

edtabularl edtabularc edtabularr The ‘environment’ forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

```

7822 \newenvironment{edtabularl}{\l@collect@body\ltabtext}{\}
7823 \newenvironment{edtabularc}{\l@collect@body\ctabtext}{\}
7824 \newenvironment{edtabularr}{\l@collect@body\rtabtext}{\}
7825
7826 %

```

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
\endquote
7827 \newcommand{\initnumbering@quote}{
7828   \ifnoquotation@else
7829   \renewcommand{\quotation}{\par\leavevmode%
7830                                   \parindent=1.5em%
7831                                   \skipnumbering%
7832                                   \ifautopar%
7833                                   \vskip-\parskip%
7834                                   \else%
7835                                   \vskip\topsep%
7836                                   \fi%
7837   \global\leftskip=\leftmargin%

```

```

7838 \global\rightskip=\leftmargin%
7839 }
7840 \renewcommand{\endquotation}{\par%
7841 \global\leftskip=0pt%
7842 \global\rightskip=0pt%
7843 \leavevmode%
7844 \skipnumbering%
7845 \ifautopar%
7846 \vskip-\parskip%
7847 \else%
7848 \vskip\topsep%
7849 \fi%
7850 }
7851 \renewcommand{\quote}{\par\leavevmode%
7852 \parindent=0pt%
7853 \skipnumbering%
7854 \ifautopar%
7855 \vskip-\parskip%
7856 \else%
7857 \vskip\topsep%
7858 \fi%
7859 \global\leftskip=\leftmargin%
7860 \global\rightskip=\leftmargin%
7861 }
7862 \renewcommand{\endquote}{\par%
7863 \global\leftskip=0pt%
7864 \global\rightskip=0pt%
7865 \leavevmode%
7866 \skipnumbering%
7867 \ifautopar%
7868 \vskip-\parskip%
7869 \else%
7870 \vskip\topsep%
7871 \fi%
7872 }
7873 \fi
7874 }
7875 %

```

XXXII Section's title commands

XXXII.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```

7876 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
7877 %

```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```

7878 \newcommand{\ledsectnomark}{%
7879   \let\chaptermark\@gobble%
7880   \let\sectionmark\@gobble%
7881   \let\subsectionmark\@gobble%
7882 }
7883 %

```

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).

```

7884 \catcode`\#=12
7885 \notbool{@noeled@sec}{%
7886 %

```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```

7887 \ifl@dmemoir
7888   \newcommand\beforeeledchapter{%
7889     \clearforchapter%
7890   }
7891 \else
7892   \newcommand\beforeeledchapter{%
7893     \if@openright%
7894       \cleardoublepage%
7895     \else%
7896       \clearpage%
7897     \fi%
7898   }
7899 \fi
7900 %

```

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 line numbers. They include tests for RTL languages.

```

7901 \def\print@rightmargin@eledsection{%
7902   \if@eled@sectioning%
7903     \begingroup%
7904     \if@RTL%
7905       \let\llap\rlap%
7906       \let\leftlinenum\rightlinenum%
7907       \let\leftlinenumR\rightlinenumR%
7908       \let\l@drd@ta\l@dld@ta%
7909       \let\l@drsn@te\l@dlsn@te%
7910     \fi%
7911     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
7912   \endgroup%
7913 \fi%
7914 }%
7915
7916 \def\print@leftmargin@eledsection{%
7917   \if@eled@sectioning%
7918     \leavevmode%
7919     \begingroup%
7920     \if@RTL%
7921       \let\rlap\llap%
7922       \let\rightlinenum\leftlinenum%
7923       \let\rightlinenumR\leftlinenumR%
7924       \let\l@dld@ta\l@drd@ta%
7925       \let\l@dlsn@te\l@drsn@te%
7926     \fi%

```

```

7927 \l@ddld@ta\csuse{LR}{\l@dlsn@te}%
7928 \endgroup%
7929 \fi%
7930 }%
7931
7932 %

```

XXXII.5 Patching standard commands

`\M@sect` 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.

```

7933 \AtBeginDocument{%
7934
7935
7936 \pretocmd{\M@sect}
7937   {\let\old@edtext=\edtext%
7938    \let\edtext=\dummy@edtext@showlemma%
7939   }
7940   {}
7941   {}
7942
7943 \apptocmd{\M@sect}
7944   {\let\edtext=\old@edtext}
7945   {}
7946   {}
7947
7948 \patchcmd{\M@sect}
7949   { #9}
7950   { #9%
7951    \print@rightmargin@eledsection%
7952   }
7953   {}
7954   {}
7955
7956 \patchcmd{\M@sect}
7957   {\hskip #3\relax}
7958   {\hskip #3\relax%
7959    \print@leftmargin@eledsection%
7960   }
7961   {}
7962   {}

```

```

7963 \patchcmd{\@mem@old@ssect}
7964   {\#5}
7965   {\#5%
7966   \print@leftmargin@eledsection%
7967   }
7968   {}
7969   {}
7970   {}
7971
7972 \patchcmd{\@mem@old@ssect}
7973   {\hskip #1}
7974   {\hskip #1%
7975   \print@rightmargin@eledsection%
7976   }
7977   {}
7978   {}
7979
7980
7981
7982 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
7983   \if@eled@sectioning\else%
7984   \ifl@dprintingpages\else%
7985   \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a
7986   \fi}%
7987   \fi}%
7988   }
7989   {}
7990   {}
7991
7992 \patchcmd{\@makechapterhead}
7993   {\#1}
7994   {\print@leftmargin@eledsection%
7995   #1%
7996   \print@rightmargin@eledsection%
7997   }
7998   {}
7999   {}
8000
8001 \patchcmd{\@makechapterhead}% For BIDI
8002   {\if@RTL\raggedleft\else\raggedright\fi}%
8003   {\if@eled@sectioning\else%
8004   \if@RTL\raggedleft\else\raggedright\fi%
8005   \fi%
8006   }%
8007   {}%
8008   {}%
8009

```

```

8010 \patchcmd{\@makeschapterhead}
8011   {#1}
8012   {\print@leftmargin@eledsection%
8013     #1%
8014     \print@rightmargin@eledsection%
8015   }
8016   {}
8017   {}
8018
8019 \pretocmd{\@sect}
8020   {\let\old@edtext=\edtext
8021     \let\edtext=\dummy@edtext@showlemma%
8022   }
8023   {}
8024   {}
8025
8026 \apptocmd{\@sect}
8027   {\let\edtext=\old@edtext}
8028   {}
8029   {}
8030
8031 \pretocmd{\@ssect}
8032   {\let\old@edtext=\edtext%
8033     \let\edtext=\dummy@edtext@showlemma%
8034   }
8035   {}
8036   {}
8037
8038 \apptocmd{\@ssect}
8039   {\let\edtext=\old@edtext}
8040   {}
8041   {}
8042
8043 %

```

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.

```

8044 \@ifpackageloaded{nameref}{
8045
8046   \patchcmd{\NR@sect}
8047     {#8}
8048     {#8%
8049       \print@rightmargin@eledsection%
8050     }
8051     {}
8052     {}
8053
8054   \patchcmd{\NR@sect}
8055     {\hskip #3\relax}
8056     {\hskip #3\relax%

```

```

8057 \print@leftmargin@eledsection%
8058 }
8059 {}
8060 {}
8061
8062 \patchcmd{\NR@ssect}
8063 {#5}
8064 {#5%
8065 \print@rightmargin@eledsection%
8066 }
8067 {}
8068 {}
8069
8070 \patchcmd{\NR@ssect}
8071 {\hskip #1}
8072 {\hskip #1%
8073 \print@leftmargin@eledsection%
8074 }
8075 {}
8076 {}
8077 }%
8078 {
8079 \patchcmd{\@sect}
8080 {#8}
8081 {#8%
8082 \print@rightmargin@eledsection%
8083 }
8084 {}
8085 {}
8086
8087 \patchcmd{\@sect}
8088 {\hskip #3\relax}
8089 {\hskip #3\relax%
8090 \print@leftmargin@eledsection%
8091 }
8092 {}
8093 {}
8094
8095 \patchcmd{\@ssect}
8096 {#5}
8097 {#5%
8098 \print@rightmargin@eledsection%
8099 }
8100 {}
8101 {}
8102
8103 \patchcmd{\@ssect}
8104 {\hskip #1}
8105 {\hskip #1%
8106 \print@leftmargin@eledsection%

```

```

8107 }
8108 {}
8109 {}
8110 }%
8111 }%
8112 %

```

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 `#`.

```

8113 {}}%
8114 \protect\catcode`\#=6 %Space NEEDS by \catcode
8115 %

```

\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.

```

8116 \AtBeginDocument{%
8117 \patchcmd{\chapter}{\clearforchapter}{%
8118   \if@eled@sectioning\else%
8119   \ifl@dprintingpages\else%
8120     \clearforchapter%
8121   \fi%
8122 \fi%
8123 }%
8124 {}%
8125 {}%
8126
8127 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8128   \if@eled@sectioning\else%
8129   \ifl@dprintingpages%
8130     \endgraf%
8131   \else%
8132     \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
8133   \fi%
8134 \fi%
8135 }%
8136 {}%
8137 {}%
8138 }%
8139 %

```

\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.

```

8140 \newif\if@eled@sectioning%
8141 %

```

We reopen a new \notbool{@noeled@sec} statement, as we will define the \elesection commands.

```
8142 \notbool{@noeled@sec}{%
8143 %
```

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.

```
8144 \newwrite\eled@sectioning@out
8145 %
```

`\eledchapter` `\eledsection` And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsubsection
\eledsubsubsection
\eledchapter*
\eledsection*
\eledsubsection*
\eledsubsubsection*
8146 \newcommand{\eledchapter}[2] [] {%
8147   #2%
8148   \ifledRcol%
8149     \immediate\write\eled@sectioningR@out{%
8150       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
8151     }%
8152   \else%
8153     \immediate\write\eled@sectioning@out{%
8154       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
8155     }%
8156   \fi%
8157 }
8158
8159 \newcommand{\eledsection}[2] [] {%
8160   #2%
8161   \ifledRcol%
8162     \immediate\write\eled@sectioningR@out{%
8163       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
8164     }%
8165   \else%
8166     \immediate\write\eled@sectioning@out{%
8167       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
8168     }%
8169   \fi%
8170 }
8171
8172 \newcommand{\eledsubsection}[2] [] {%
8173   #2%
8174   \ifledRcol%
8175     \immediate\write\eled@sectioningR@out{%
8176       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
8177     }%

```

```

8178 \else%
8179 \immediate\write\eled@sectioning@out{%
8180 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
8181 }%
8182 \fi%
8183 }
8184 \newcommand{\eledsubsubsection}[2][{}]{%
8185 #2%
8186 \ifledRcol%
8187 \immediate\write\eled@sectioningR@out{%
8188 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR
8189 }{-}{R}
8190 }%
8191 \else%
8192 \immediate\write\eled@sectioning@out{%
8193 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL
8194 }{-}{-}
8195 }%
8196 \fi%
8197 }
8198 \WithSuffix\newcommand\eledchapter*[2][{}]{%
8199 #2%
8200 \ifledRcol%
8201 \immediate\write\eled@sectioningR@out{%
8202 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8203 }%
8204 \else%
8205 \immediate\write\eled@sectioning@out{%
8206 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{-}
8207 }%
8208 \fi%
8209 }
8210 \WithSuffix\newcommand\eledsection*[2][{}]{%
8211 #2%
8212 \ifledRcol%
8213 \immediate\write\eled@sectioningR@out{%
8214 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8215 }%
8216 \else%
8217 \immediate\write\eled@sectioning@out{%
8218 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{-}
8219 }%
8220 \fi%
8221 }
8222 }
8223 \WithSuffix\newcommand\eledsubsection*[2][{}]{%
8224 #2%

```

```

8226 \ifledRcol%
8227 \immediate\write\eled@sectioningR@out{%
8228 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{
R}
8229 }%
8230 \else%
8231 \immediate\write\eled@sectioning@out{%
8232 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
}{*}{
8233 }%
8234 \fi%
8235 }
8236
8237 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
8238 #2%
8239 \ifledRcol%
8240 \immediate\write\eled@sectioningR@out{%
8241 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR
}{*}{R}
8242 }%
8243 \else%
8244 \immediate\write\eled@sectioning@out{%
8245 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
}{*}{
8246 }%
8247 \fi%
8248 }
8249 %

```

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.

```

8250 \def\eled@chapter#1#2#3#4#5{%
8251 \ifstrempy{#4}%
8252 {%
8253 \ifstrempy{#1}%
8254 {%
8255 \csgdef\eled@sectioning@#3#5{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}}%

```

```

8256 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
      {#2}}%
8257 }%Need for \pairs, because of using parbox.
8258 {%
8259 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
8260 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
      {#2}}%Need for \pairs, because of using parbox.
8261 }%
8262 }%
8263 {%
8264 \ifstrempy{#1}%
8265 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
8266 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#1}{#2}}}%Bug in LaTeX!
8267 }%
8268 \listcsgadd{eled@sections#5@@}{#3}%
8269 }
8270 \def\eled@section#1#2#3#4#5{%
8271 \ifstrempy{#4}%
8272 {\ifstrempy{#1}%
8273 {%
8274 \csgdef{eled@sectioning@#3#5}{\section{#2}}%
8275 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
      {#2}}%Need for \pairs, because of using parbox.
8276 }%
8277 {%
8278 \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
8279 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
      {#1}}%Need for \pairs, because of using parbox.
8280 }%
8281 }%
8282 {\ifstrempy{#1}%
8283 {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
8284 {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
8285 }
8286 \listcsgadd{eled@sections#5@@}{#3}%
8287 }
8288 \def\eled@subsection#1#2#3#4#5{%
8289 \ifstrempy{#4}%
8290 {\ifstrempy{#1}%
8291 {%
8292 \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
8293 \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)
8294 }%
8295 {%
8296 \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%

```

```

8297 \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)
8298 }%
8299 }%
8300 {\ifstrempy{#1}%
8301 {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
8302 {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
8303 }
8304 \listcsgadd{eled@sections#5@@}{#3}%
8305 }
8306 \def\eled@subsubsection#1#2#3#4#5{%
8307 \ifstrempy{#4}%
8308 {\ifstrempy{#1}%
8309 {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
8310 {\csgdef{eled@sectioning@#3#5}{\subsubsection{#1}{#2}}}%
8311 }%
8312 {\ifstrempy{#1}%
8313 {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
8314 {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in
LaTeX!
8315 }
8316 \listcsgadd{eled@sections#5@@}{#3}%
8317 }
8318
8319 %

```

End of the conditional test about noeledsec option.

```

8320 }{}
8321 %

```

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.

```

8322 \def\normal@page@break{}
8323 %

```

`\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.

```
8324 \def\l@prev@pb{}
8325 \def\l@prev@nopb{}
8326 %
```

`\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.

```
8327 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
8328 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
8329 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
8330 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
8331 %
```

`\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.

```
8332 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
8333 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
8334 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
8335 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
8336 %
```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

```
8337 \def\led@pb@setting{before}
8338 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
8339 %
```

`\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`.

```
8340 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
8341 \newcommand{\led@check@nopb}{%
8342   \IfStrEq{\led@pb@setting}{before}{%
8343     \xifinlist{\the\absline@num}{\l@prev@nopb}%
8344     {\numdef{\abs@prevline}{\the\absline@num-1}%
8345     \xifinlist{\abs@prevline}{\normal@page@break}%
8346     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
8347     }{}%
8348   }{}%}
```

```

8349     {}%
8350   {}%
8351   \IfStrEq{\led@pb@setting}{after}{%
8352     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
8353       \xifinlist{\the\absline@num}{\normal@page@break}%
8354       {\nopagebreak[4]\enlargethispage{\baselineskip}}%
8355     }%
8356 }%
8357   {}%
8358   {}%
8359   {}%
8360 }
8361 %

```

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.

```

8362 \newcommand{\check@pb@in@verse}{%
8363   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
8364   enabling page breaks in verse control, while on a hanging verse.
8365   \ifnum\page@num=\last@page@num\else%If we have change page
8366     \IfStrEq{\led@pb@setting}{before}{%
8367       \numgdef{\abs@line@verse}{\the\absline@num-1}%
8368       \ledpbnum{\abs@line@verse}%
8369     }{}%
8370     \IfStrEq{\led@pb@setting}{after}{%
8371       \numgdef{\abs@line@verse}{\the\absline@num-1}%
8372       \lednopbnum{\abs@line@verse}%
8373     }{}%
8374   \fi%
8375 }
8376 %

```

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.³⁴

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.³⁵

```

8377 \def\Hy@raisedlink@left#1{%
8378     \ifvmode
8379         #1%
8380     \else
8381         \Hy@SaveSpaceFactor
8382         \llap{\smash{%
8383             \begingroup
8384                 \let\HyperRaiseLinkLength\@tempdima
8385                 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
8386                 \HyperRaiseLinkHook
8387             \expandafter\endgroup
8388             \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
8389                 \Hy@RestoreSpaceFactor
8390                 #1%
8391                 \Hy@SaveSpaceFactor
8392             }%
8393         }}%
8394         \Hy@RestoreSpaceFactor
8395         \penalty\@M\hskip\z@\relax
8396     \fi
8397 }
8398 %

```

XXXVI Compatibility with eledmac

Here, we define some commands for the `eledmac-compat` option.

```

8399 \ifeledmaccompat%
8400
8401     \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
8402     \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
8403     \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
8404     \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
8405
8406     \unless\ifnocritical@
8407         \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
8408         \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%

```

³⁴<http://tex.stackexchange.com/a/17138/7712>.

³⁵The code is inspired by an answer given by @unbonpetit. Thanks to him. <http://texnique.fr:80/osqa/questions/781/hyraisedlink-perturbe-la-maniere-dont-se-fait-la-coupe-de-ligne/801>.

```

8409 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
8410 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
8411 \let\hsizetwocol\Xhsizetwocol
8412 \let\hsizethreecol\Xhsizethreecol
8413 \let\bhookXnote\Xbhooknote
8414 \let\boxsymlinenum\Xboxsymlinenum
8415 \let\symlinenum\Xsymlinenum
8416 \let\beforenumberinfootnote\Xbeforenumber
8417 \let\afternumberinfootnote\Xafternumber
8418 \let\beforeXsymlinenum\XbeforeXsymlinenum
8419 \let\afterXsymlinenum\XafterXsymlinenum
8420 \let\inplaceofnumber\Xinplaceofnumber
8421 \let\Xlemmaseparator\lemmaseparator
8422 \let\afterlemmaseparator\Xafterlemmaseparator
8423 \let\beforelemmaseparator\Xbeforelemmaseparator
8424 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
8425 \let\txbeforeXnotes\Xtxbeforenotes
8426 \let\afterXrule\Xafterrule
8427 \let\numberonlyfirstinline\Xnumberonlyfirstinline
8428 \let\numberonlyfirstintwoline\Xnumberonlyfirstintwoline
8429 \let\nonumberinfootnote\Xnonumberinfootnote
8430 \let\pstartinfootnote\Xpstart
8431 \let\pstartinfootnoteeverytime\Xpstarteverytime
8432 \let\onlyXpstart\Xonlypstart
8433 \let\Xnonumberinfootnote\Xnonumber
8434 \let\nonbreakableafternumber\Xnonbreakableafternumber
8435 \let\maxhXnotes\Xmaxhnotes
8436 \let\beforeXnotes\Xbeforenotes
8437 \let\boxlinenum\Xboxlinenum
8438 \let\boxlinenumalign\Xboxlinenumalign
8439 \let\boxstartlinenum\Xboxstartlinenum
8440 \let\boxendlinenum\Xboxendlinenum
8441 \let\twoline\Xtwoline
8442 \let\morethantwoline\Xmorethantwoline
8443 \let\twolinebutnotmore\Xtwolinebutnotmore
8444 \let\twolineonlyinsamepage\Xtwolineonlyinsamepage
8445 \fi
8446
8447 \unless\ifnofamiliar@
8448 \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
8449 \fi
8450 \newcommandx{\parafootsep}[2][1,usedefault]{%
8451 \Xparafootsep[#1]{#2}%
8452 \parafootsepX[#1]{#2}
8453 }%
8454
8455 \newcommandx{\afternote}[2][1,usedefault]{%
8456 \Xafternote[#1]{#2}%
8457 \afternoteX[#1]{#2}%
8458 }%

```

```

8459 \unless\ifnoend@
8460   \let\XendXtwolines\Xendtwolines
8461   \let\XendXmorethantwolines\Xendmorethantwolines
8462   \let\hookXendnote\Xendhooknote
8463   \let\boxXendlinenum\Xendboxlinenum%
8464   \let\boxXendlinenumalign\Xendboxlinenumalign%
8465   \let\boxXendstartlinenum\Xendboxstartlinenum%
8466   \let\boxXendendlinenum\Xendboxendlinenum%
8467   \let\XendXlemmaseparator\Xendlemmaseparator
8468   \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
8469   \let\XendXafterlemmaseparator\Xendafterlemmaseparator
8470   \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
8471 \fi
8472
8473 \AtBeginDocument{%
8474   \ifdef\lineref{}\let\lineref\edlineref}%
8475 }%
8476
8477
8478
8479 \fi%
8480 %

```

</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 changed³⁶ 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:

<pre>I saw my friend \critext{Smith} \Afootnote{Jones C, D.}/ on Tuesday.</pre>	<pre>1 I saw my friend 2 Smith on Tuesday. 2 Smith] Jones C, D.</pre>
---	---

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:

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

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
```

³⁶A name like `\text` is likely to be defined by other \TeX packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.

```
\let<=\critext
```

Then you might say `<\{Smith\}\variant{Jones}/`. This of course destroys the ability to use `<` in any new macro definitions, so long as it remains in effect; hence it should be used with care.

Changing the character at the end of the command requires more work:

```
\catcode`\<=\active
\def\xtext#1#2>{\critext{#1}{#2}/}
\let<=\xtext
```

This allows you to say `<\{Smith\}\Afootnote{Jones}>`.

Aliases for `\critext` of the first kind shown here also can't be nested—that is, you can't use the alias in the text that forms the first argument to `\critext`. (See VI p. 115 to find out why.) Aliases of the second kind may be nested without any problem.

If you really have to use `\critext` in any of the tabular or array environments, then `\edtext` must not be used in the same environment. If you use `\critext` in one of these environments then you have to issue the declaration `\usingcritext` beforehand. The declaration `\usingedtext` must be issued to revert to the default assumption that `\edtext` will be used.

Appendix A.2 Migration from ledmac to eledmac

In `eledmac`, some changes were made in the code to allow easy customization. This may cause problems for people who have already made their own. The next sections explain how to handle this.

If you have created your own series using `\addfootins` and `\addfootinsX`, you must use instead the `\newseries` command (see 6.6.1 p. 32), 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. 33) or `\Xfootnote` options (6.2.2 p. 24). Otherwise please add a new ticket on Github to request a new function for doing this.³⁷

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]
```

³⁷<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. 46). 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. 61.

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 `\printnnum`. This spurious space has been deleted. However, if you want to keep the previous spurious space, you may load the package with the `oldprintnnumspace` 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. 64 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 <code>\renewcommand</code></i>	<i>Replaced with</i>
<code>\@led@extranofeet</code>	<code>\newseries</code>
<code>\apprefprefixmore</code>	<code>\setapprefprefixmore</code>
<code>\apprefprefixsingle</code>	<code>\setapprefprefixsingle</code>
<code>\endstanzaextra</code>	Optional argument of <code>\&</code>
<code>\hangingsymbol</code>	<code>\sethangingsymbol</code>
<code>\ledfootinsdim</code>	<code>\Xmaxhnotes</code> and <code>\maxhnotesX</code>
<code>\parafootftmsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\notenumfont</code>	<code>\Xnotenumfont</code> , <code>\Xendnotenumfont</code> and <code>\notenumfontX</code>
<code>\notefontsetup</code>	<code>\Xnotefontsize</code> , <code>\Xendnotefontsize</code> and <code>\notefontsizeX</code>
<code>\sidenotesep</code>	<code>\setsidenotsep</code>
<code>\startstanzahook</code>	Optional argument of <code>\stanza</code>
<code>\symplinenum</code>	<code>\Xsymplinenum</code>

Appendix A.9.5 Commands the names of which have been changed

In order to help the migration from `eledmac` to `reledmac`, you may load `reledmac` with `eledmac-compat` option. However, it is advised not to, and to change the command names themselves instead. In many cases, you use only a few of them, except the `\footparagraph` command.

<i>Old command</i>	<i>New command</i>
<code>\footparagraph</code>	<code>\Xarrangement</code>
<code>\footnormal</code>	<code>\Xarrangement</code>
<code>\foottwocol</code>	<code>\Xarrangement</code>
<code>\footthreecol</code>	<code>\Xarrangement</code>
<code>\footparagraphX</code>	<code>\arrangementX</code>
<code>\footnormalX</code>	<code>\arrangementX</code>
<code>\foottwocolX</code>	<code>\arrangementX</code>
<code>\footthreecolX</code>	<code>\arrangementX</code>
<code>\afterlemmaseparator</code>	<code>\Xafterlemmaseparator</code>
<code>\afternote</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\afternumberinfootnote</code>	<code>\Xafternumber</code>
<code>\afterXrule</code>	<code>\Xafterrule</code>
<code>\afterXsymplinenum</code>	<code>\Xaftersymplinenum</code>
<code>\beforelemmaseparator</code>	<code>\Xbeforelemmaseparator</code>
<code>\beforenumberinfootnote</code>	<code>\Xbeforenumber</code>
<code>\beforeXnotes</code>	<code>\Xbeforenotes</code>
<code>\beforeXsymplinenum</code>	<code>\Xbeforesymplinenum</code>

<i>Old command</i>	<i>New command</i>
<code>\bhookXnote</code>	<code>\Xbhookendnote</code>
<code>\bhookXnote</code>	<code>\Xbhooknote</code>
<code>\boxendlinenum</code>	<code>\Xboxendlinenum</code>
<code>\boxlinenum</code>	<code>\Xboxlinenum</code>
<code>\boxlinenumalign</code>	<code>\Xboxlinenumalign</code>
<code>\boxstartlinenum</code>	<code>\Xboxstartlinenum</code>
<code>\boxsymlinenum</code>	<code>\Xboxsymlinenum</code>
<code>\boxXendlinenum</code>	<code>\Xendboxlinenum</code>
<code>\boxXendlinenumalign</code>	<code>\Xendboxlinenumalign</code>
<code>\boxXendstartlinenum</code>	<code>\boxXendstartlinenum</code>
<code>\letboxXendendlinenum</code>	<code>\Xendletboxendlinenum</code>
<code>\hsizetwocol</code>	<code>\Xhsizetwocol</code>
<code>\hsizethreecol</code>	<code>\Xhsizethreecol</code>
<code>\inplaceoflemmaseparator</code>	<code>\Xinplaceoflemmaseparator</code>
<code>\inplaceofnumber</code>	<code>\Xinplaceofnumber</code>
<code>\lemmaseparator</code>	<code>\Xlemmaseparator</code>
<code>\maxhXnotes</code>	<code>\Xmaxhnotes</code>
<code>\morethantwolines</code>	<code>\Xmorethantwolines</code>
<code>\nonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\notesXwidthliketwocolumns</code>	<code>\noteswidthliketwocolumnsX</code>
<code>\noXlemmaseparator</code>	<code>\Xnolemmaseparator</code>
<code>\numberonlyfirstinline</code>	<code>\Xnumberonlyfirstinline</code>
<code>\numberonlyfirstintwolines</code>	<code>\Xnumberonlyfirstintwolines</code>
<code>\nonbreakableafternumber</code>	<code>\Xnonbreakableafternumber</code>
<code>\onlyXpstart</code>	<code>\Xonlypstart</code>
<code>\parafootsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\pstartinfootnote</code>	<code>\Xpstart</code>
<code>\pstartinfootnoteeverytime</code>	<code>\Xpstarteverytime</code>
<code>\symlinenum</code>	<code>\Xsymlinenum</code>
<code>\twolines</code>	<code>\Xtwolines</code>
<code>\twolinesbutnotmore</code>	<code>\Xtwolinesbutnotmore</code>
<code>\twolinesonlyinsamepage</code>	<code>\Xtwolinesonlyinsamepage</code>
<code>\txtbeforeXnotes</code>	<code>\Xtxtbeforenotes</code>
<code>\XendXafterlemmaseparator</code>	<code>\Xendafterlemmaseparator</code>
<code>\XendXbeforelemmaseparator</code>	<code>\Xendbeforelemmaseparator</code>
<code>\XendXinplaceoflemmaseparator</code>	<code>\Xendinplaceoflemmaseparator</code>
<code>\XendXlemmaseparator</code>	<code>\Xendlemmaseparator</code>
<code>\XendXmorethantwolines</code>	<code>\Xendmorethantwolines</code>
<code>\XendXtwolines</code>	<code>\Xendtwolines</code>
<code>\Xnonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\lineref</code>	<code>\edlineref</code>

Appendix A.9.6 Endnotes

With `reledmac`, there is now one auxiliary file for every endnotes set (`.Aend`, `.Bend`, `.Cend` etc.). If you have overridden `\doendnotes` (which you would not have done) you must adapt your code.

Appendix A.9.7 Z Series

The ‘Z’ series of notes has been removed. Only five series are provided now by default: A, B, C, D, E.

Appendix A.9.8 Internal commands

Users who have overridden internal commands, which is wrong, must adapt according to the following. Or better, they should not override any of such commands and use `reledmac` options instead.

- If you have modified `\Xfootfmt`, note that the fourth argument is now mandatory.
- `\unvxh` has been replaced with `\Xunvxh` and `\unvxhX` with two mandatory arguments.

Appendix A.10 Migration to `reledmac` 2.1.0

`Reledmac` 2.1.0 fix some bugs when using `\Xbhooknote` and `\bhooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }.`
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` in the argument of `\Xbhooknote/\bhooknoteX`.

Appendix A.11 Migration to `reledmac` 2.1.3

`Reledmac` 2.1.3 fix an historical bug, (style in `ledmac` 0.7!) which doubled the space before the rules of paragraphed familiar footnotes. Consequently, if you use paragraphed familiar footnotes, you should maybe adapt it, playing with `\beforenotesX`.

Appendix A.12 Migration to `reledmac` 2.3.0

Before `reledmac` 2.3.0, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use optional argument of `\stanza` or `\endverse`.

Appendix A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{\langle...\rangle}` to the use of `\Xlinangeseparator` or `/` and `\Xendlinangeseparator` (7.2.3 p. 34).

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. 38.

`\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 be printed in right side critical footnotes, even if not explicitly asked by using `\Xlineflag`.

The version 2.13.1 solves this issue. Please use `\Xlineflag` if you want to add the right flag.

References

- [Bre96] Herbert Breger. `tabmac`. October 1996. (Available from CTAN in `macros/plain/contrib/tabmac`)
- [Bur01] John Burt. ‘Typesetting critical editions of poetry’. *TUGboat*, **22**, 4, pp 353–361, December 2001. (Code available from CTAN in `macros/latex/contrib/poemscol`)
- [Eck03] Matthias Eckermann. *The Parallel-Package*. April 2003. (Available from CTAN in `macros/latex/contrib/parallel`)
- [Fai03] Robin Fairbairns. *footmisc—a portmanteau package for customising footnotes in L^AT_EX*. February 2003. (Available from CTAN in `macros/latex/contrib/footmisc`)
- [LW90] John Lavagnino and Dominik Wujastyk. ‘An overview of `edmac`: a PLAIN TeX format for critical editions’. *TUGboat*, **11**, 4, pp. 623–643, November 1990. (Code available from CTAN in `macros/plain/contrib/edmac`)
- [Lüc03] Uwe Lück. ‘`ednotes`—critical edition typesetting with LaTeX’. *TUGboat*, **24**, 2, pp. 224–236, 2003. (Code available from CTAN in `macros/latex/contrib/ednotes`)
- [Sul92] Wayne G. Sullivan. *The file `edstanzan.doc`*. June 1992. (Available from CTAN in `macros/plain/contrib/edmac`)
- [Wil02] Peter Wilson. *The memoir class for configurable typesetting*. November 2002. (Available from CTAN in `macros/latex/contrib/memoir`)
- [Wil04] Peter Wilson and Maïeul Rouquette. *Parallel typesetting for critical editions: the `eledpar` package*. December 2004. (Available from CTAN in `macros/latex/contrib/ledmmac`)

Index

Symbols

<code>\&</code>	45
<code>\@EDROWFILL@</code>	1
<code>\@adv</code>	1
<code>\@advancestanzanumber</code>	1
<code>\@doclearpage</code>	1
<code>\@doreinfeetX</code>	1
<code>\@edindex@hyperref</code>	1
<code>\@edrowfill@</code>	1
<code>\@edtext@level</code>	1
<code>\@emptytoks</code>	1
<code>\@endmsd</code>	1

\@fnpos	1
\@footnotemark	1
\@footnotetext	1
\@getfirstseries	1
\@gobblefive	1
\@gobblefour	1
\@gobblethree	1
\@h	1
\@hangingsymbol	1
\@iiiminipage	1
\@insertstanzaumber	1
\@k	1
\@l@tempcnta	1
\@l@tempcntb	1
\@lab	1
\@led@testifnofoot	1
\@lemma	1
\@line@@num	1
\@lock	1
\@lopL	1
\@lopR	1
\@makechapterhead	1
\@makeschapterhead	1
\@mem@extranofeet	1
\@mem@old@ssect	1
\@mpfnpos	1
\@msd	1
\@msd@c	1
\@msd@options@iffullpage	1
\@msdata@list	1
\@nl	1
\@nl@reg	1
\@opXfeet	1
\@pend	1
\@pendR	1
\@ref	1
\@ref@reg	1
\@sect	1
\@series	1
\@set	1
\@sidenotesep	1
\@ssect	1
\@startstanza	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1
\@xloop	1
\@xympar	1
CLASSarticle	64

CLASSbook	64, 318
CLASSmemoir	183, 184, 239–241, 271, 318, 337, 394, 398
CLASSscrbook	398
COMMAND*footnote	65
COMMAND\...\@footnotemark...	186
COMMAND\...d@ta	138
COMMAND\<hook	
@<series	229
COMMAND\<hookname	
<pseudoseries	231
COMMAND\<type	
footfmt	174
COMMAND\@@line	166
COMMAND\@MM	154, 395
COMMAND\@Rlineflag	273, 395
COMMAND\@SErefprefix	251
COMMAND\@SErefprefixmore	251
COMMAND\@add@	308
COMMAND\@adv	101
COMMAND\@apprefprefixmore	251
COMMAND\@apprefprefixsingle	251
COMMAND\@bsphack	242
COMMAND\@doclearpage	240, 241, 388, 398
COMMAND\@doreinfeetX	398
COMMAND\@dprintingcolumns	395
COMMAND\@edindex@hyperref	273, 274
COMMAND\@edtext@level	118
COMMAND\@esphack	242
COMMAND\@fnpos	204
COMMAND\@footnotemark	184, 388, 398
COMMAND\@footnotetext	184, 185, 388
COMMAND\@gobble	116
COMMAND\@gobblefive	226, 397
COMMAND\@gobblefour	394
COMMAND\@gobblethree	387
COMMAND\@h	169
COMMAND\@hangingsymbol	277
COMMAND\@iiiminipage	263, 265, 387, 398
COMMAND\@iiiminipage	263
COMMAND\@l	393
COMMAND\@l@dtmpecnta	141, 142, 149
COMMAND\@l@dtmpecntb	142
COMMAND\@l@reg	393
COMMAND\@lab	97, 242, 245, 248, 387
COMMAND\@ldunboxmpfoot	265
COMMAND\@led@extranofeet	338
COMMAND\@ledinnote@command	269, 270
COMMAND\@lemma	121, 123
COMMAND\@lock	92, 278
COMMAND\@lopL	388

COMMAND\@lopR	388
COMMAND\@makecol	237, 239, 240, 398
COMMAND\@mpfnpos	204
COMMAND\@msd	286
COMMAND\@msd@c	286
COMMAND\@msd@options@iffullpage	292
COMMAND\@msdata@list	286, 287
COMMAND\@nl	97, 98, 100–102, 111, 245, 387, 388
COMMAND\@nl@reg	98, 335, 388, 393
COMMAND\@opXfeet	388
COMMAND\@opfeetX	398
COMMAND\@opextrafeeti	398
COMMAND\@page	100, 245
COMMAND\@pend	388
COMMAND\@pendR	388
COMMAND\@ref	98, 107–109, 112, 116
COMMAND\@ref@reg	108, 388
COMMAND\@reinserts	237–240, 398
COMMAND\@secondoftwo	66
COMMAND\@sect	320
COMMAND\@series	227
COMMAND\@set	102
COMMAND\@sidenotesep	262
COMMAND\@stopmsd	286
COMMAND\@sw	109, 124, 127, 128
COMMAND\@tag	118, 119, 122
COMMAND\@tempcnta	78
COMMAND\@tempcntb	78
COMMAND\@toksa	83
COMMAND\@toksb	83
COMMAND\@xloop	150
COMMAND\@xympar	256, 398
COMMAND\Aendnote	15, 24
COMMAND\Afootfmt	154
COMMAND\Afootgroup	154
COMMAND\Afootnote	8, 15, 23, 24, 27, 120, 161, 183, 205, 219, 397
COMMAND\Afootstart	154
COMMAND\AtEveryPend	18, 48, 133, 395, 396, 398
COMMAND\AtEveryPstart	18, 48, 395, 396, 398, 401
COMMAND\AtEveryStanza	47, 402
COMMAND\AtEveryStopStanza	48, 402
COMMAND\Bendnote	15, 23
COMMAND\Bfootnote	8, 15, 183, 205, 219
COMMAND\Centering	41
COMMAND\Cfootnote	183
COMMAND\Columns	78, 158
COMMAND\Dfootnote	183
COMMAND\Efootnote	183
COMMAND\Gls	57
COMMAND\Hy@raisedlink	330

COMMAND\Hy@raisedlink@left	330
COMMAND\LTR	41
COMMAND\NR	320
COMMAND\Pages	78, 237, 239
COMMAND\ProcessOptionsX	69
COMMAND\RL	40
COMMAND\RaggedLeft	41
COMMAND\RaggedRight	41
COMMAND\SEonlypage	250, 400
COMMAND\Seref	51–53, 250, 252, 401
COMMAND\Serefonlypage	51–53, 341, 400
COMMAND\Serefwithpage	51, 53, 250, 252, 341, 400
COMMAND\Stanza	393
COMMAND\Waklam	308
COMMAND\X@doreinfeet	238, 398
COMMAND\XR@prefix	255
COMMAND\XR@test	255
COMMAND\XR@test@mac	255
COMMAND\XR@test@mac@test	255
COMMAND\XXXXXXfmt	334
COMMAND\XXXXXXfmt	334
COMMAND\Xafterlemmaseparator	38, 338
COMMAND\Xafternote	42, 337, 338
COMMAND\Xafternumber	36, 338
COMMAND\Xafterrule	43, 206, 338, 393, 396
COMMAND\Xaftersymlinenum	36, 338
COMMAND\Xarrangement	33, 65, 154, 155, 229, 338
COMMAND\Xarrangement@footparagraph	159
COMMAND\Xarrangement@normal	155
COMMAND\Xarrangement@paragraph	159
COMMAND\Xbeforeinserting	41
COMMAND\Xbeforelemmaseparator	38, 338
COMMAND\Xbeforenotes	43, 205, 338, 393, 396
COMMAND\Xbeforenumber	34, 36, 338
COMMAND\Xbeforesymlinenum	36, 338
COMMAND\Xbhookendnote	339
COMMAND\Xbhookgroup	42, 400, 401
COMMAND\Xbhooknote	41, 339, 340, 398, 399
COMMAND\Xboxendlinenum	37, 339, 397
COMMAND\Xboxlinenum	37, 339
COMMAND\Xboxlinenumalign	37, 339, 397
COMMAND\Xboxstartlinenum	37, 339, 397
COMMAND\Xboxsymlinenum	37, 339
COMMAND\Xcolalign	41, 396
COMMAND\Xdo@feet	237, 388, 398
COMMAND\Xend	225
COMMAND\XendXafterlemmaseparator	339
COMMAND\XendXbeforelemmaseparator	339
COMMAND\XendXinplaceoflemmaseparator	339
COMMAND\XendXlemmaseparator	339

COMMAND\XendXmorethantwolines	339
COMMAND\XendXtwolines	339
COMMAND\Xendafterenumber	36, 399
COMMAND\Xendafterlemmaseparator	39, 339
COMMAND\Xendafternote	44, 401
COMMAND\Xendafternumber	38
COMMAND\Xendafterpagenumber	38, 53
COMMAND\XendafterpagenumberSErefonlypage	341
COMMAND\Xendaftersymlinenum	36, 38, 399
COMMAND\Xendahookinplaceofnumber	38, 399
COMMAND\Xendahooklinenum	38, 399
COMMAND\Xendbeforelemmaseparator	39, 339
COMMAND\Xendbeforelinenum	38
COMMAND\Xendbeforenumber	36, 399
COMMAND\Xendbeforepagenumber	38, 52, 53
COMMAND\XendbeforepagenumberSErefonlypage	52
COMMAND\Xendbeforesymlinenum	36, 38, 399
COMMAND\Xendbhookinplaceofnumber	38, 399
COMMAND\Xendbhooklinenum	38, 399
COMMAND\Xendbhooknote	41
COMMAND\Xendboxendlinum	37, 397
COMMAND\Xendboxlinenum	37, 339, 395
COMMAND\Xendboxlinenumalign	37, 339, 397
COMMAND\Xendboxstartlinenum	37, 397
COMMAND\Xendboxsymlinenum	37, 399
COMMAND\Xendhangindent	40, 399, 401
COMMAND\Xendinplaceoflemmaseparator	24, 39, 339
COMMAND\Xendinplaceofnumber	37, 398
COMMAND\Xendinsertsep@	212
COMMAND\Xendlemmadisablefontselection	39
COMMAND\Xendlemmafont	39, 400
COMMAND\Xendlemmaseparator	25, 39, 339
COMMAND\Xendletboxendlinum	339
COMMAND\Xendlineflag	53
COMMAND\Xendlineprefixmore	38, 53
COMMAND\Xendlineprefixsingle	38, 53
COMMAND\Xendlinerrangeseparator	34, 53, 341, 399
COMMAND\Xendmorethantwolines	24, 35, 53, 339, 396, 397
COMMAND\Xendnonumber	35, 398
COMMAND\Xendnote	208, 224, 225, 396
COMMAND\Xendnotefontsize	40, 338
COMMAND\Xendnotenumfont	38, 39, 338
COMMAND\Xendnotes	211
COMMAND\Xendnumberonlyfirstinline	34, 399
COMMAND\Xendnumberonlyfirstintwolines	34, 399
COMMAND\Xendparagraph	44, 393
COMMAND\Xendsep	44
COMMAND\Xendsublinesep	36, 53
COMMAND\Xendsymlinenum	34, 399
COMMAND\Xendtwolines	24, 35, 53, 339, 396, 397

COMMAND\Xendtwolinesbutnotmore	35, 53, 396, 397
COMMAND\Xendtwolinesonlyinsamepage	35, 53, 396, 397
COMMAND\Xendwrapcontent	40, 402
COMMAND\Xendwraplemma	40, 402
COMMAND\Xfootfmt	340
COMMAND\Xfootgroup	158
COMMAND\Xfootins	157
COMMAND\Xfootnote	50, 56, 118, 334, 390, 394, 396, 400, 402
COMMAND\Xfootstarts	158
COMMAND\Xhangindent	40, 399
COMMAND\Xhsize	341, 400, 401
COMMAND\Xsizethreecol	42, 44, 339
COMMAND\Xsizetwocol	42, 44, 230, 339
COMMAND\Xinplaceoflemmaseparator	24, 38, 339
COMMAND\Xinplaceofnumber	37, 339, 396, 397
COMMAND\Xinsertparafootsep	164, 166
COMMAND\Xledsetnormalparstuff	336, 337, 396
COMMAND\Xlemmadisablefontselection	39
COMMAND\Xlemmafont	39, 400
COMMAND\Xlemmaseparator	38, 232, 234, 236, 339
COMMAND\Xlineflag	52, 341, 342, 402
COMMAND\Xlinangeseparator	34, 53, 341, 399
COMMAND\Xmaxhnotes	43, 64, 65, 338, 339, 393, 395
COMMAND\Xmorethantwolines	24, 35, 53, 339, 395
COMMAND\Xnoindent	399
COMMAND\Xnolemmaseparator	38, 236, 339
COMMAND\Xnonbreakableafternumber	36, 339, 391
COMMAND\Xnonumber	35, 339
COMMAND\Xnonumberinfootnote	339
COMMAND\Xnotefontsize	40, 338
COMMAND\Xnotefontsize@<s>	164, 168, 169
COMMAND\Xnotenumfont	39, 338
COMMAND\Xnoteswidthliketwocolumns	44, 394
COMMAND\Xnumberonlyfirstinline	34, 94, 231, 232, 234, 339, 390, 395
COMMAND\Xnumberonlyfirstintwolines	34, 339, 390
COMMAND\Xonlypstart	36, 339, 390, 395
COMMAND\Xparafootsep	42, 95, 338, 339, 402
COMMAND\Xparafootsep@series	164
COMMAND\Xparinden	341
COMMAND\Xparindent	40, 396, 399, 401
COMMAND\Xprenotes	43, 206, 402
COMMAND\Xprenotes@	157, 206, 390
COMMAND\Xpstart	35, 36, 336, 339, 390, 395
COMMAND\Xpstarteverytime	35, 336, 339, 395
COMMAND\Xragged	42
COMMAND\Xrigidbalance	166, 341, 400
COMMAND\Xstanza	36, 48
COMMAND\Xstanzaseparator	36
COMMAND\Xsublinesep	21, 36, 53, 341
COMMAND\Xsublinesepside	21, 36

COMMAND\Xsymlinenum	34, 42, 338, 339, 397
COMMAND\Xtoendnotes	25, 225
COMMAND\Xtwolines	24, 35, 53, 179, 180, 231, 339, 395
COMMAND\Xtwolinesappref	231
COMMAND\Xtwolinesbutnotmore	35, 53, 339, 396
COMMAND\Xtwolinesbutnotmoreappref	231
COMMAND\Xtwolinesonlyinsamepage	35, 53, 339, 396
COMMAND\Xtxtbeforenotes	42, 339, 402
COMMAND\Xunvxh	162, 340
COMMAND\Xwidth	44, 341, 401
COMMAND\Xwrapcontent	40, 402
COMMAND\Xwraplemma	40, 41, 402
COMMAND\&	338
COMMAND\absline@num	91, 140
COMMAND\accent	117
COMMAND\actionlines@list	92, 141
COMMAND\actions@list	92
COMMAND\add@inserts	92, 148
COMMAND\add@inserts@next	148
COMMAND\add@msd@	286
COMMAND\add@msdata	286, 287
COMMAND\add@msdata@firstlineofpage	289
COMMAND\add@msddata	286
COMMAND\add@penalties	139, 149
COMMAND\addcontentsline	315
COMMAND\addfootins	334, 337
COMMAND\addfootinsX	334, 337
COMMAND\addtoennotes	225
COMMAND\advancelabel@refs	244
COMMAND\advanceline	21, 22, 94, 101, 113, 398
COMMAND\advancepageno	237
COMMAND\affixlin@num	262
COMMAND\affixline@num	142, 145, 146, 388
COMMAND\affixpstart@num	146
COMMAND\afterXrule	338
COMMAND\afterXsymlinenum	338
COMMAND\afterenumber	36
COMMAND\aftergroup	116, 120
COMMAND\afterlemmaseparator	338
COMMAND\afternote	338
COMMAND\afternoteX	42, 337, 338
COMMAND\afternumberinfootnote	338
COMMAND\afterruleX	43, 393, 396
COMMAND\applabel	52, 245, 246, 396, 402
COMMAND\appref	50, 52, 53, 250, 252, 400, 401
COMMAND\apprefprefixmore	52, 338
COMMAND\apprefprefixsingle	52, 338
COMMAND\apprefwithpage	52, 53, 250, 252, 397, 400
COMMAND\arrangementX	33, 65, 187, 229, 338
COMMAND\arrangementX@normal	192

COMMAND\article	14
COMMAND\at@every@pend	133
COMMAND\autopar	17, 131, 134, 135, 202, 389, 391, 392, 396
COMMAND\ballast	65
COMMAND\ballast@count	139, 149
COMMAND\baselineskip	33, 160, 164
COMMAND\beforeXnotes	338
COMMAND\beforeXsymlinenum	338
COMMAND\beforeedchapter	10, 62, 316, 317
COMMAND\beforeinsertingX	41
COMMAND\beforelemmaseparator	338
COMMAND\beforenotesX	43, 340, 392, 393, 396
COMMAND\beforenumberinfootnote	338
COMMAND\begin	293
COMMAND\beginnumbering	16, 18, 19, 79, 82, 91, 95, 96, 110, 134, 208, 285, 390, 393, 397, 398
COMMAND\bf	390
COMMAND\bfseries	39, 390
COMMAND\bhookXnote	339
COMMAND\bhookgroupX	42, 400
COMMAND\bhooknoteX	41, 340, 398, 399
COMMAND\body	278
COMMAND\bodyfootmarkA	31
COMMAND\book	14
COMMAND\boxXendlinenum	339
COMMAND\boxXendlinenumalign	339
COMMAND\boxXendstartlinenum	339
COMMAND\boxendlinenum	339
COMMAND\boxlinefootnote	176
COMMAND\boxlinenum	339
COMMAND\boxlinenumalign	339
COMMAND\boxstartlinenum	339
COMMAND\boxsymlinenum	339
COMMAND\break	34, 162
COMMAND\brokenpenalty	149
COMMAND\centering	41
COMMAND\ch@ck@l@ck	388
COMMAND\ch@cksub@l@ck	145, 388
COMMAND\chapter	61, 318, 322, 393, 396, 398, 402
COMMAND\chaptermark	316
COMMAND\check@pb@in@verse	329
COMMAND\colalignX	41, 396
COMMAND\collect@body	294
COMMAND\color	402
COMMAND\colorbox	66
COMMAND\columns	44
COMMAND\columnwidth	160, 394
COMMAND\command names	230, 231
COMMAND\copyright	117
COMMAND\correct@Xfootins@box	395
COMMAND\correct@footinsX@box	395

COMMAND\count	168
COMMAND\critex	389
COMMAND\critext	123, 333, 334, 337
COMMAND\csname	70, 126
COMMAND\ctab	309, 314
COMMAND\ctabtext	314
COMMAND\dcoll	303
COMMAND\def	67
COMMAND\detokenize	126
COMMAND\dimen	168
COMMAND\dimexpr	44
COMMAND\discretionary	161
COMMAND\displaywidowpenalty	149
COMMAND\do@actions	139–141, 388
COMMAND\do@actions@fixedcode	388
COMMAND\do@actions@next	140, 141
COMMAND\do@ballast	139, 140, 149
COMMAND\do@feetX	398
COMMAND\do@insidelinehook	391
COMMAND\do@line	92, 115, 132, 135, 138, 139, 148, 149, 278, 388, 389, 391, 393
COMMAND\do@linehook	388
COMMAND\do@lockoff	93
COMMAND\do@lockon	93
COMMAND\dodoreintrafeet	387
COMMAND\doendnotes	25, 212, 340, 397
COMMAND\doendnotesbysection	25, 212, 226, 397
COMMAND\doinsidelinehook	22, 394
COMMAND\dolinehook	22, 394
COMMAND\doreintrafeeti	398
COMMAND\doreintrafeetii	398
COMMAND\doxtrafeet	237, 387
COMMAND\doxtrafeeti	398
COMMAND\doxtrafeetii	398
COMMAND\dummy@ref	116
COMMAND\edaftertab	60, 309, 310
COMMAND\edatleft	60, 307
COMMAND\edatright	60, 307
COMMAND\edbeforetab	60, 309
COMMAND\edfilldimen	308
COMMAND\edfont@info	122
COMMAND\edgls	57, 268
COMMAND\edgls...	401
COMMAND\edindex	55–57, 268, 269, 272, 274, 297, 391, 394, 395, 398, 399
COMMAND\edindexlab	57
COMMAND\edlabel	49–52, 54, 117, 242, 244, 245, 247, 248, 254, 268, 297, 330, 387, 390–392, 395, 400
COMMAND\edlabelE	51, 246
COMMAND\edlabelS	51, 246
COMMAND\edlabelSE	51
COMMAND\edlineref	49, 50, 242, 339, 395, 397, 400
COMMAND\edmakelabel	51, 254

COMMAND\edpageref	49, 50, 242, 247, 254
COMMAND\edrowfill	308
COMMAND\edtabcolsep	302
COMMAND\edtext	6, 23, 24, 26–29, 45, 50–52, 55, 58, 65, 92, 107–109, 112, 115– 123, 125–127, 129, 245, 246, 249, 297, 299, 318, 330, 333, 334, 337, 387, 389, 391, 393–397
COMMAND\edtext@level	397
COMMAND\edvertdots	61, 307
COMMAND\edvertline	60, 61, 307
COMMAND\elechapter	62
COMMAND\eled@sectioning@out	323
COMMAND\eledchapter	61, 337, 394, 398
COMMAND\eledchapter*	61
COMMAND\eledmac@error	387
COMMAND\eledsection	6, 15, 61, 116, 137, 316, 337, 396
COMMAND\eledsection*	61
COMMAND\eledsubsection	61, 337
COMMAND\eledsubsection*	61
COMMAND\eledsubsubsection	61, 337
COMMAND\eledsubsubsection*	61
COMMAND\eledxxx	10, 62, 322, 323, 393
COMMAND\eledxxxx	316
COMMAND\elsection	323
COMMAND\else	267, 316
COMMAND\empty	78, 143, 144, 242
COMMAND\end	293
COMMAND\end@lemmas	116
COMMAND\endashchar	44, 172
COMMAND\endgraf	132, 164, 202
COMMAND\endlock	21, 93, 114, 282
COMMAND\endminipage	263, 265, 387, 398
COMMAND\endmsdata	31
COMMAND\endnotes	396, 400
COMMAND\endnumbering	16, 19, 79, 81, 82, 388, 397
COMMAND\endprint	208, 211, 226, 336
COMMAND\endstanzaextra	338
COMMAND\endsub	21, 93, 112
COMMAND\endverse	340
COMMAND\everypar	134
COMMAND\extensionchars	63, 79
COMMAND\externaldocument	54, 255
COMMAND\fix@l@cks	388
COMMAND>falseverse	337, 391, 393
COMMAND\fi	316
COMMAND\firstlinenum	19, 142, 389
COMMAND\firstsublinenum	19, 389
COMMAND\fix@page	98, 100, 388
COMMAND\flag@end	112, 122, 393
COMMAND\flag@start	112, 122, 393, 394
COMMAND\flagstanza	48
COMMAND\floatingpenalty	154, 395

COMMAND\flush@notes	149, 150
COMMAND\fnpos	204, 392
COMMAND\footfmt	153, 156
COMMAND\footfmt...	187
COMMAND\footfootmarkA	31
COMMAND\footfudgefactor	162
COMMAND\footfudgefiddle	65, 160, 387
COMMAND\footgroup	153
COMMAND\footins	157
COMMAND\footnormal	230, 338, 387
COMMAND\footnormalX	338
COMMAND\footnote	31, 64, 183–185, 335, 388
COMMAND\footnote@lang	173
COMMAND\footnoteA	16, 31
COMMAND\footnoteB	16
COMMAND\footnoteC	23
COMMAND\footnoteE	31
COMMAND\footnoteX	8, 222
COMMAND\footnoteX@reading	224
COMMAND\footnoteXmk	236
COMMAND\footnotelang@lua	152
COMMAND\footnotelang@poly	153
COMMAND\footnoteoption@	152, 399
COMMAND\footnoterule	168
COMMAND\footnotesize	40
COMMAND\footparagraph	160, 230, 338, 393
COMMAND\footparagraphX	197, 338, 393
COMMAND\footsplitskips	388, 395
COMMAND\footstart	153, 157, 168
COMMAND\footstrut	164
COMMAND\foothreecol	338
COMMAND\foothreecolX	338, 397
COMMAND\foottwocol	338
COMMAND\foottwocolX	338, 397
COMMAND\foreignlanguage	40
COMMAND\fullstop	44, 341
COMMAND\get@edindex@hyperref	273
COMMAND\get@edindex@ledinnote@command	269
COMMAND\get@fnmark	185
COMMAND\get@index@command	392
COMMAND\get@linelistfile	388
COMMAND\get@thisfootnote	191
COMMAND\getline@num	139, 140
COMMAND\gl@p	84
COMMAND\global	98
COMMAND\globaldefs	98
COMMAND\gls	57, 276
COMMAND\hangindentX	40, 396, 399
COMMAND\hangingsymbol	338, 389
COMMAND\hbox	161, 162

COMMAND\hfill	392
COMMAND\hidenumbering	22, 106, 396
COMMAND\hidenumberingonleftpage	22, 106, 402
COMMAND\hidenumberingonrightpage	22, 106, 402
COMMAND\hline	58
COMMAND\hrulefill	308
COMMAND\hspace	34, 157, 160–162, 168, 171, 203, 388, 394
COMMAND\hspaceX	341, 400, 401
COMMAND\hsizethreecol	339
COMMAND\hsizethreecolX	42, 44
COMMAND\hsizetwocol	339
COMMAND\hsizetwocolX	42, 44
COMMAND\hyperlinkR	272
COMMAND\hyperlinkformat	272
COMMAND\hyperlinkformatR	273
COMMAND\if@RTL	71
COMMAND\if@edtext@	394, 397
COMMAND\if@eled@sectioning	322
COMMAND\if@firstlineofpage	71
COMMAND\if@firstlineofpageR	71
COMMAND\if@msd@options@fullpage	292
COMMAND\if@msdata@insertedfrompreviouspage	289
COMMAND\if@nobreak	133
COMMAND\if@noneed@Footnote	112
COMMAND\ifXnote@	78
COMMAND\ifbypage@	84
COMMAND\ifbypage@R	84
COMMAND\ifbypstart@	84
COMMAND\ifbypstart@R	84
COMMAND\iffirst@linenum@out@	110
COMMAND\ifindtl@innote	79
COMMAND\ifindtl@notenumber	79
COMMAND\ifinserthangingsymbol	278
COMMAND\ifinstanza	278
COMMAND\ifstwofollowinglines	180
COMMAND\ifl@d@Xmorethantwolines	177, 395
COMMAND\ifl@d@Xtwolines	177
COMMAND\ifl@d@dash	177
COMMAND\ifl@d@elin	177
COMMAND\ifl@d@esl	177
COMMAND\ifl@d@pnum	177
COMMAND\ifl@d@ssub	177
COMMAND\ifl@dend@X	224
COMMAND\ifl@dmemoir	387
COMMAND\ifl@dpadding	394
COMMAND\ifl@dpairing	78, 389
COMMAND\ifl@dprintingpages	395
COMMAND\ifl@dskipnumber	143
COMMAND\ifl@dstartendok	308
COMMAND\ifl@imakeidx	71

COMMAND\ifledRcol	78, 389
COMMAND\ifledRcol@	78, 393
COMMAND\iflemmacommand@	394
COMMAND\ifnoend@	213
COMMAND\ifnoedgroup@	267
COMMAND\ifnoteschanged@	94
COMMAND\ifnumberedpar@	130
COMMAND\ifnumbering	79, 82
COMMAND\ifnumberingR	78, 389
COMMAND\ifnumberline	122, 143
COMMAND\ifpst@rted	389
COMMAND\ifpst@rtedL	80
COMMAND\ifseriesbefore	228
COMMAND\ifstopmsdata@inserted@	285
COMMAND\ifsublines@	91, 103
COMMAND\iftrue	397
COMMAND\ifvmode	244
COMMAND\ifxxx	316
COMMAND\ignorespaces	120
COMMAND\imki@wrindexentry	71
COMMAND\immediate	110, 207
COMMAND\indent	18, 134, 340
COMMAND\index	275
COMMAND\indtl@wrindexentry	71
COMMAND\initnumbering@quote	314, 398
COMMAND\initnumbering@reg	388
COMMAND\initnumbering@sectcmd	398
COMMAND\inplaceoflemmaseparator	339
COMMAND\inplaceofnumber	339
COMMAND\insert	147, 153, 156, 187
COMMAND\insert@Xtxtbeforenotes	151
COMMAND\insert@count	107, 112, 119
COMMAND\insert@countR	119
COMMAND\insert@msdata	286, 291
COMMAND\insertthangingsymbol	392
COMMAND\insertlines@list	92, 107, 108
COMMAND\insertparafootsepX	201
COMMAND\inserts@list	115, 131, 148, 161
COMMAND\interAfootnotelinepenalty	389
COMMAND\interfootnotelinepenalty	389
COMMAND\interlinepenalty	154
COMMAND\interparanoteglue	337
COMMAND\justifying	41
COMMAND\l@advance@parledgroup@beforenormalnotes	398
COMMAND\l@d@@wrindexhyp	394
COMMAND\l@d@add	124
COMMAND\l@d@end	208, 224
COMMAND\l@d@nums	119, 122–124, 177
COMMAND\l@d@section	208
COMMAND\l@d@set	102, 113

COMMAND\l@dampcount	299
COMMAND\l@dbfnote	185, 388
COMMAND\l@dcheckstartend	308
COMMAND\l@dchset@num	102
COMMAND\l@dcolcount	299, 300
COMMAND\l@dcollect@body	293
COMMAND\l@dcollect@body	293
COMMAND\l@dcsnote	393
COMMAND\l@dcsnotetext	138, 259, 260
COMMAND\l@dcsnotetext@l	138, 259, 260
COMMAND\l@dcsnotetext@r	138, 259, 260
COMMAND\l@ddodoreintrafeet	238, 387
COMMAND\l@ddoxtrafeet	238, 387
COMMAND\l@demptyd@ta	389
COMMAND\l@dend@close	207
COMMAND\l@dend@open	207
COMMAND\l@dend@stuff	208
COMMAND\l@denvbody	293
COMMAND\l@dfeetbeginmini	387
COMMAND\l@dfeetendmini	387
COMMAND\l@dgetline@margin	389
COMMAND\l@dgetlock@disp	389
COMMAND\l@dgetref@num	248, 249
COMMAND\l@dgetsidenote@margin	256, 389
COMMAND\l@dgobbeloptarg	394
COMMAND\l@dgonblearg	394
COMMAND\l@dgonbleoptarg	298
COMMAND\l@dlabel@parse	248, 249
COMMAND\l@dld@ta	142, 144
COMMAND\l@dld@rbox	261
COMMAND\l@dlsn@te	389
COMMAND\l@dlsnote	393
COMMAND\l@dmake@labels	244, 255
COMMAND\l@dmake@labelsR	255
COMMAND\l@dnumpsstartL	80, 389
COMMAND\l@dp@rsefootspec	178
COMMAND\l@dpush@begins	293
COMMAND\l@drd@ta	142, 144
COMMAND\l@dref@undefined	248
COMMAND\l@drrsn@te	389
COMMAND\l@drrsnote	393
COMMAND\l@dtabaddcols	308
COMMAND\l@dtabnoexpands	387
COMMAND\l@dumboxmpfoot	398
COMMAND\l@dunboxmpfoot	389
COMMAND\l@dzeropenalties	389, 394
COMMAND\l@pb	328
COMMAND\l@prev@nopb	328
COMMAND\l@prev@pb	328
COMMAND\l@reg	335

COMMAND\label	18, 51, 54, 57, 242, 243, 249
COMMAND\label@refs	242
COMMAND\labelstarttrue	18, 390
COMMAND\labelref@list	242, 245
COMMAND\language	161
COMMAND\last@page@num	388
COMMAND\lastbox	134
COMMAND\lastskip	112
COMMAND\leavevmode	18, 134
COMMAND\led@check@nopb	328
COMMAND\led@check@pb	328
COMMAND\led@nopb	328, 329
COMMAND\led@nopbnum	328
COMMAND\led@pb	328, 329
COMMAND\led@pb@macro	328
COMMAND\led@pbnum	328
COMMAND\led@reinit@index@fornote	275
COMMAND\led@set@index@fornote	275
COMMAND\ledRflag	272
COMMAND\ledchapter	337, 391
COMMAND\ledfootinsdim	338
COMMAND\ledinnernote	54, 257, 393, 402
COMMAND\ledinnote	271, 336, 397
COMMAND\ledinnotemark	56, 336, 396
COMMAND\ledleftnote	54, 257
COMMAND\ledlinenum	90, 389
COMMAND\ledllfill	139
COMMAND\ledlsnotefontsetup	402
COMMAND\ledlsnotesep	55
COMMAND\ledlsnotewidth	54
COMMAND\lednopb	63, 328
COMMAND\lednopbinverse	329
COMMAND\lednopbinversetrue	47, 63
COMMAND\lednopbnum	328
COMMAND\ledouternote	54, 257, 393, 402
COMMAND\ledpb	63, 328, 337
COMMAND\ledpbnum	328
COMMAND\ledpbsetting	63, 328, 399
COMMAND\ledrightnote	54, 257
COMMAND\ledrsnotefontsetup	402
COMMAND\ledrsnotesep	55
COMMAND\ledrsnotewidth	54
COMMAND\ledsection	337
COMMAND\ledsectnomark	316
COMMAND\ledsectnotoc	315
COMMAND\ledsetnormalparstuff	336, 337, 396
COMMAND\ledsetnormalparstuff@common	202
COMMAND\ledsetnormalparstuffX	336, 337, 396
COMMAND\ledsidenote	54, 257, 260
COMMAND\ledsubsection	337

COMMAND\ledsubsubsection	337
COMMAND\ledxxx	393
COMMAND\left	60
COMMAND\leftctab	309
COMMAND\leftheadline	90
COMMAND\leftlinenum	20, 89, 387, 389
COMMAND\leftltab	309
COMMAND\leftnoteupfalse	54
COMMAND\leftstartnum	146
COMMAND\leftftab	309
COMMAND\leftsidenote	260
COMMAND\leftskip	157, 160, 162
COMMAND\lemma	3, 24, 26–29, 115, 118, 120, 122, 123, 125, 333, 389, 390, 397, 398, 400
COMMAND\lemmaseparator	339
COMMAND\let	27, 45, 282, 387
COMMAND\letboxXendendlinenum	339
COMMAND\line	166, 169
COMMAND\line@list	92, 108, 122
COMMAND\line@list@stuff	79, 95, 96, 110, 387, 389
COMMAND\line@list@version	98
COMMAND\line@margin	86, 144, 256
COMMAND\line@num	91, 93, 142, 387
COMMAND\line@set	123, 124
COMMAND\lineation	20, 85
COMMAND\linebreak	34
COMMAND\linenum	24, 26, 27, 50–52, 115, 123, 247, 249, 255, 333, 401
COMMAND\linenum@out	110, 242, 245
COMMAND\linenumberlist	19, 20, 78, 143, 144, 387
COMMAND\linenumberstyle	22, 89, 387
COMMAND\linenumincrement	19, 389
COMMAND\linenummargin	20, 86, 256
COMMAND\linenumr@p	89, 387, 389
COMMAND\linenumrep	89, 389
COMMAND\linenumsep	20, 55, 90, 257
COMMAND\linerangesep@	236
COMMAND\lineref	242, 247, 254, 339, 395
COMMAND\list@clear	83
COMMAND\list@clearing@reg	389
COMMAND\list@create	83
COMMAND\lock@disp	88
COMMAND\lock@off	105
COMMAND\lock@on	104
COMMAND\lockdisp	21, 88
COMMAND\loop	150, 278
COMMAND\ltab	309, 310, 314
COMMAND\ltabtext	314
COMMAND\m@mmf@prepare	184
COMMAND\makeatletter	138
COMMAND\makehboxoffhboxes	163, 164
COMMAND\makeindex	55, 272

COMMAND\makelabel	254
COMMAND\managestanza@modulo	279
COMMAND\marginpar	54, 64, 255, 256, 388
COMMAND\marginparwidth	54, 257
COMMAND\markboth	138
COMMAND\mathchardef	279
COMMAND\maxhXnotes	339
COMMAND\maxhnotesX	43, 65, 338, 392, 393, 395–397
COMMAND\maxlinesinpar@list	95
COMMAND\measurebody	311
COMMAND\measuretbody	311
COMMAND\memorybreak	19
COMMAND\morenoexpands	65, 66, 115, 117
COMMAND\morethantwolines	339
COMMAND\mpfnpos	204, 392
COMMAND\mpnormalfootgroup	388
COMMAND\mpnormalvfootnote	388
COMMAND\msdata	30, 31, 284–286
COMMAND\multfootsep	32, 183
COMMAND\multiplefootnotemarker	183
COMMAND\musixtex	393
COMMAND\n@num	389, 396
COMMAND\n@num@ref	396
COMMAND\new@line	111, 388
COMMAND\newcommand	27, 67, 183, 244
COMMAND\newcommandx	27
COMMAND\newhookarg@specific	236
COMMAND\newhookcommand@series	230, 232, 396
COMMAND\newhookcommand@series@reload	232
COMMAND\newhookcommand@toggle@reload	231, 394
COMMAND\newhooktoggle@series	231, 396
COMMAND\newhooktoggle@specific	235
COMMAND\newif	396
COMMAND\newline	34
COMMAND\newlinechar	224
COMMAND\newseries	32, 334, 337, 338
COMMAND\newseries@	216, 217, 229
COMMAND\newverse	48, 337, 393
COMMAND\next	278
COMMAND\next@action	96
COMMAND\next@actionline	96
COMMAND\next@insert	148
COMMAND\nl@regR	98
COMMAND\no@expands	65, 122, 387
COMMAND\noXlemmaseparator	339
COMMAND\nobreak	177
COMMAND\nocritical	217
COMMAND\noeledsec	62, 337
COMMAND\noendnotes	337
COMMAND\noexpand	335

COMMAND\nofamiliar	233
COMMAND\noindent	18, 134, 399
COMMAND\noindentX	399
COMMAND\nomk@	236
COMMAND\nonbreakableafternumber	339
COMMAND\nonnumberinfootnote	339
COMMAND\normal@footnotemarkX	187
COMMAND\normal@page@break	327
COMMAND\normal@pars	202
COMMAND\normalbfnoteX	389
COMMAND\normalbodyfootmarkX	187
COMMAND\normalfootfmt	45, 157, 164, 173, 208, 401
COMMAND\normalfootfmtX	188, 189
COMMAND\normalfootfootmarkX	189
COMMAND\normalfootgroup	158
COMMAND\normalfootgroupX	189
COMMAND\normalfootnoterule	154
COMMAND\normalfootstart	157, 160
COMMAND\normalfootstartX	189
COMMAND\normalvfootnote	156
COMMAND\normalvfootnoteX	187
COMMAND\notbool	316
COMMAND\notfontsetup	338
COMMAND\notfontsizeX	40, 338
COMMAND\notenumfont	338
COMMAND\notenumfontX	39, 338
COMMAND\notesXwidthliketwocolumns	339
COMMAND\noteswidthliketwocolumnsX	44, 339, 394, 396
COMMAND\num@lines	130, 149
COMMAND\numberlinefalse	19
COMMAND\numberlinetrue	19
COMMAND\numberonlyfirstinline	228, 339
COMMAND\numberonlyfirstintwolines	339
COMMAND\numberpstartfalse	18
COMMAND\numberpstarttrue	18, 35, 336, 390, 398
COMMAND\numberstanza	36
COMMAND\numberstanzafalse	48
COMMAND\numberstanzatrue	48
COMMAND\numlabfont	20, 44, 45, 90
COMMAND\none@line	130
COMMAND\nonehalfspacing	399
COMMAND\nonlyXpstart	339
COMMAND\nonlysideX	223
COMMAND\npage@action	93, 103
COMMAND\npage@start	93, 389
COMMAND\npagecontents	93
COMMAND\npagelinesep	56
COMMAND\npageno	237
COMMAND\npageparbreak	337
COMMAND\npageref	51, 247

COMMAND\par	26, 34, 134, 202
COMMAND\par@line	130, 149
COMMAND\para@footgroup	161
COMMAND\para@footgroupX	200
COMMAND\para@footsetup	160, 387
COMMAND\para@footsetupX	198, 387, 394
COMMAND\para@vfootnoteX	199
COMMAND\parafootfmt	163, 164, 401
COMMAND\parafootfmtX	200
COMMAND\parafootftm	166
COMMAND\parafootftmX	201
COMMAND\parafootftmsep	338
COMMAND\parafootsep	339, 392, 397
COMMAND\parafootsepX	42, 95, 338, 339, 402
COMMAND\parafootstart	160
COMMAND\parafootstartX	198
COMMAND\paravfootnote	161, 164
COMMAND\parfillskip	163
COMMAND\parindent	399
COMMAND\parindentX	40, 341, 399, 401
COMMAND\parshape	65
COMMAND\parskip	134
COMMAND\pausenumbering	19, 82, 96, 97, 135, 392, 394, 401
COMMAND\penalty	163
COMMAND\pend	2, 6, 17–20, 22, 62, 113, 115, 118, 124, 130–135, 146, 147, 335, 392, 393, 402
COMMAND\preXnotes	396, 402
COMMAND\prenotesX	43, 207, 396
COMMAND\prepare@Xprenotes	205
COMMAND\prev@nopb	328
COMMAND\prev@pb	328
COMMAND\prevlineX	94
COMMAND\prevpageX@num	95
COMMAND\print@Xfootnoterule	396
COMMAND\print@Xnotes	237, 239
COMMAND\print@Xnotes@forpages	395
COMMAND\print@eledsection	137
COMMAND\print@footnoteXrule	396
COMMAND\print@leftmargin@eledsection	317
COMMAND\print@lemma	173
COMMAND\print@line	136
COMMAND\print@notesX@forpages	395
COMMAND\print@rightmargin@eledsection	317
COMMAND\printendlines	213, 252, 341, 387, 389
COMMAND\printlinefootnote	174, 176, 395
COMMAND\printlinefootnotearea	176, 177, 395
COMMAND\printlinefootnotenumbers	174
COMMAND\printlines	157, 172, 177, 178, 213, 252, 341, 387, 389, 395, 400
COMMAND\printnpnum	336, 341
COMMAND\printpstart	173
COMMAND\protect	117, 335

COMMAND\providecommand	183, 387
COMMAND\pstart	2, 6, 17–20, 22, 61, 62, 102, 113, 118, 124, 130–134, 137, 147, 322, 335, 389–391, 393, 394, 396–398, 401, 402
COMMAND\pstartinfootnote	339
COMMAND\pstartinfootnoteeverytime	339
COMMAND\pstartnum	146
COMMAND\pstartref	50, 242, 248, 392
COMMAND\pstarts	390
COMMAND\raggedX	42
COMMAND\raggedleft	41
COMMAND\raggedright	41
COMMAND\raw@text	130, 131
COMMAND\rbracket	38, 39, 44
COMMAND\read@linelist	95, 97
COMMAND\ref	51, 54, 57
COMMAND\reformatted@	252
COMMAND\reformattedwithpage	252
COMMAND\relax	18, 102, 140, 148, 282, 298, 335
COMMAND\renewcommand	65, 338, 341
COMMAND\reset@msd@options@iffullpage	292
COMMAND\resetprevline@	94
COMMAND\resetprevpage@	95
COMMAND\resumenumbering	19, 79, 82, 96, 97, 135, 389, 393, 394, 401
COMMAND\right	60
COMMAND\rightctab	309
COMMAND\rightlinenum	20, 89, 387, 389
COMMAND\rightltab	310
COMMAND\rightnoteupfalse	54
COMMAND\rightrtab	310
COMMAND\rightsidenote	260
COMMAND\rightright	157, 160, 162, 163
COMMAND\rightstartnum	146
COMMAND\rigidbalance	166, 168, 169, 341, 400
COMMAND\rigidbalanceX	166, 341, 400
COMMAND\robustify	34
COMMAND\roman	303, 400
COMMAND\rtab	309, 310, 314
COMMAND\rtabtext	311, 314
COMMAND\sameword	27–29, 124–126, 129, 395, 397, 399, 402
COMMAND\sameword@inedtext	125, 126
COMMAND\saweword	125
COMMAND\scriptsize	90
COMMAND\section	61, 389
COMMAND\section@num	79
COMMAND\sectionmark	316
COMMAND\select@lemmafont	45, 151
COMMAND\series	216, 217
COMMAND\series@	217
COMMAND\seriesatbegin	32, 227, 396
COMMAND\seriesatend	32, 228, 396

COMMAND\set@Xtxtbeforenotes	150
COMMAND\set@line	122
COMMAND\set@line@action	93, 103
COMMAND\setSErefonlypageprefixmore	52, 252, 401
COMMAND\setSErefonlypageprefixsingle	52, 252, 401
COMMAND\setSErefprefixmore	52
COMMAND\setSErefprefixsingle	52
COMMAND\setapprefprefixmore	52, 338
COMMAND\setapprefprefixsingle	52, 338, 400
COMMAND\setcommand@series	230
COMMAND\sethangingsymbol	47, 277, 338, 399
COMMAND\sethangingsymbol	46
COMMAND\setistwofollowinglines	180
COMMAND\setl@dlprbox	261
COMMAND\setline	21, 22, 94, 98, 102, 113, 117, 132, 398
COMMAND\setlinenum	22, 98, 102, 113, 387
COMMAND\setmsdatalabel	31
COMMAND\setmsdataseries	31
COMMAND\setprintendlines	213, 214, 389
COMMAND\setprintlines	178, 180, 213, 389
COMMAND\setsidenotesep	55
COMMAND\setsidenotsep	338
COMMAND\setstanzaindent	280
COMMAND\setstanzaindents	46, 279, 335
COMMAND\setstanzapenalties	279
COMMAND\setstanzavalues	279
COMMAND\settoggle@series	229, 390, 394
COMMAND\showlemma	116, 387, 388
COMMAND\showwordrank	29, 126
COMMAND\sidenote@margin	388
COMMAND\sidenotemargin	54, 388, 393
COMMAND\sidenotesep	338
COMMAND\sidepstartnumtrue	18
COMMAND\skip	157
COMMAND\skipnumbering	22, 105, 106, 114, 389, 396, 397
COMMAND\skipnumbering@reg	396
COMMAND\small	40
COMMAND\special	12
COMMAND\splitmaxdepth	154, 168
COMMAND\splitoff	166
COMMAND\splittopskip	154, 168, 169
COMMAND\stanza	21, 22, 47, 48, 282, 338, 340, 399
COMMAND\stanza@hang	282
COMMAND\stanza@line	282
COMMAND\stanzaindent	46, 280, 395
COMMAND\stanzaindent*	46
COMMAND\stanzaindentbase	279
COMMAND\stanzanumwrapper	48
COMMAND\startlock	21, 93, 114, 282
COMMAND\startstanzahook	338

COMMAND\startsub	21, 93, 112
COMMAND\stopmsd	286
COMMAND\stopmsdata	30, 285
COMMAND\strip@pt	160
COMMAND\strutbox	168
COMMAND\sub@action	93, 103
COMMAND\sub@lock	92
COMMAND\sub@off	101, 245
COMMAND\sub@on	101, 245
COMMAND\subline@num	91, 93
COMMAND\sublinenum@rep	387
COMMAND\sublinenumberstyle	22, 89, 387
COMMAND\sublinenumincrement	19
COMMAND\sublinenumr@p	89, 387, 389
COMMAND\sublinenumrep	89, 389
COMMAND\sublineref	50, 242, 247
COMMAND\subsectionmark	316
COMMAND\sw@inthisedtext	119
COMMAND\sw@list@inedtext	122, 129
COMMAND\symlinenum	339
COMMAND\symplinenenum	338
COMMAND\sza@penalty	282
COMMAND>tag	395
COMMAND\text	333
COMMAND\text{ <i>language</i> }	40
COMMAND\textcolor	66
COMMAND\textheight	65
COMMAND\the	387
COMMAND\thefootnoteA	31
COMMAND\thefootnoteX	391
COMMAND\thelabidx	274
COMMAND\thepage	98
COMMAND\thepstart	18
COMMAND\thepstartL	390
COMMAND\thepstartR	390
COMMAND\thestanza	48
COMMAND>this@line@list@version	110
COMMAND>thisfootnote	191
COMMAND\threecolfootfmt	169, 401
COMMAND\threecolfootfmtX	196
COMMAND\threecolfootgroup	168
COMMAND\threecolfootgroupX	196
COMMAND\threecolfootsetup	168
COMMAND\threecolfootsetupX	195
COMMAND\threecolvfootnote	168
COMMAND\threecolvfootnoteX	196
COMMAND\toendnotes	25, 211, 402
COMMAND\twocolfootfmt	401
COMMAND\twocolfootfmtX	194
COMMAND\twocolfootgroupX	194

COMMAND\twocolfootsetupX	193
COMMAND\twocolvfootnoteX	194
COMMAND\twolines	229, 339
COMMAND\twolines@A	229
COMMAND\twolines@B	229
COMMAND\twolines@C	229
COMMAND\twolinesbutnotmore	339
COMMAND\twolinesonlyinsamepage	339
COMMAND\txtbeforeXnotes	339
COMMAND\unhbox	161
COMMAND\unpenalty	163, 164
COMMAND\unskip	163
COMMAND\unvxh	163, 340
COMMAND\unvxhX	340
COMMAND\upbracefill	308
COMMAND\usingcritext	334, 337
COMMAND\usingdtext	334, 337
COMMAND\VAfootnote	154
COMMAND\variant	27
COMMAND\ vbox	132, 134, 162, 166, 204
COMMAND\ vfootnote	153, 157, 161, 168
COMMAND\vl@dbfnote	185, 388
COMMAND\vl@disnote	260
COMMAND\vl@dlsnote	260
COMMAND\vl@dosnote	260
COMMAND\vl@drsnote	260
COMMAND\ vnumfootnoteX	389
COMMAND\ vsize	43, 65
COMMAND\ vsplit	149
COMMAND\waklam	308
COMMAND\waklamec	308
COMMAND\wapunktel	308
COMMAND\wastricht	308
COMMAND\widthX	44, 341, 401
COMMAND\wrap@edcrossref	247, 394
COMMAND\wrapcontentX	40, 402
COMMAND\wrapped@bodyfootmarkX	202
COMMAND\wrapped@footfootmarkX	201
COMMAND\x...	50
COMMAND\xdef	83, 282
COMMAND\xflagref	50, 248, 341, 400
COMMAND\xleft@appenditem	84, 116
COMMAND\xlineref	50, 341, 400
COMMAND\xpageref	50
COMMAND\xpstartref	50, 392
COMMAND\xr	54
COMMAND\xright@appenditem	83, 84
COMMAND\xsublineref	50
COMMAND\xxref	51, 249, 255, 392, 395
COMMAND\zz@@@	387

ENVIRONMENTastanza	400
ENVIRONMENTedarrayc	314
ENVIRONMENTedarrayl	314
ENVIRONMENTedarrayr	314
ENVIRONMENTedtabularc	314
ENVIRONMENTedtabularl	314
ENVIRONMENTedtabularr	314
ENVIRONMENTledgroup	70, 266, 341, 400
ENVIRONMENTledgroupsized	266
PACKAGE(r)(e)ledmac	32
PACKAGEEledmac	11, 67, 93, 271, 336, 337, 395, 396
PACKAGEEledpar	396
PACKAGEEtoolbox	70
PACKAGEParallel	343
PACKAGEREledmac	340, 341
PACKAGEamsgen	292
PACKAGEamsmath	292
PACKAGEbabel	40, 66, 303, 400
PACKAGEbiblatex	64
PACKAGEbidi	40, 41, 71, 399
PACKAGEccaption	78
PACKAGEcolor	66
PACKAGEedmac	1, 6, 10, 12, 13, 67, 177, 183, 242, 279, 333, 343, 387
PACKAGEedstanza	1, 13, 277
PACKAGEeledmac	1, 10, 13–16, 56, 124, 183, 268, 271, 296, 318, 330, 334, 336–338, 391, 393, 395
PACKAGEeledpar	78, 154, 316, 343, 389, 393–395
PACKAGEetex	399
PACKAGEetoolbox	83, 124, 217, 229, 237, 259, 316, 327, 328
PACKAGEfloatrow	64, 341
PACKAGEfootmisc	32, 66, 183, 343
PACKAGEgeometry	14
PACKAGEglossaries	57, 276, 400
PACKAGEhandout	394
PACKAGEhyperlink	224
PACKAGEhyperref	50, 118, 201, 202, 243, 273, 320, 330, 392–394, 401
PACKAGEifluatex	70
PACKAGEifxetex	70
PACKAGEimakeidx	55, 64, 70, 71, 268, 271, 337, 391–393, 395
PACKAGEindextols	275
PACKAGEindextool	337
PACKAGEindextools	55, 64, 71, 79, 268, 271, 275, 337, 395, 400
PACKAGEinputenc	126
PACKAGEledarab	66
PACKAGEledmac	1, 10, 13, 14, 66, 83, 271, 333, 334, 337, 340
PACKAGEledpar	66
PACKAGEMemoir	70, 271, 337, 343, 394
PACKAGEMorewrites	64
PACKAGEMusixtex	393
PACKAGEperpage	400
PACKAGEpolyglossia	38, 66, 120, 153, 173, 400

PACKAGEragged2e	41, 70
PACKAGEreledmac	1, 2, 10–12, 14–16, 18, 19, 22–24, 26–28, 30, 32, 34, 37, 40, 41, 43–45, 47, 48, 50–52, 54–58, 62, 64–68, 84, 87, 92, 94, 97, 98, 100, 109, 110, 117, 147, 155, 157, 162, 183, 208, 217, 221, 222, 229, 237, 247, 250, 255, 271, 289, 296, 316, 317, 329, 330, 337, 338, 340, 341, 398, 401
PACKAGEreledpar	1, 4, 6, 8, 15, 18, 44, 50, 52, 53, 62–64, 66, 68, 78, 84, 95, 100, 119, 155, 158, 203, 204, 217, 223, 236, 237, 239, 267, 277, 399, 400
PACKAGESuffix	70
PACKAGETabmac	1, 13, 343
PACKAGEuninormalize	28
PACKAGEXargs	27, 70
PACKAGEXkeyval	68, 236
PACKAGEXr	4, 54, 255, 401
PACKAGEXref	255
PACKAGEXstring	70, 273

A

\absline@num	1
Abu Kamil Shuja' b. Aslam	13
\actionlines@list	1
\actions@list	1
\add@inserts	1
\add@inserts@next	1
\add@msdata	1
\add@msdata@firstlineofpage	1
\add@penalties	1
\addtol@denvbody	1
Adelard II	13
\advancelabel@refs	1
\advanceline	1, 21
\advancepageno	1
\Aendnote	24
\affixline@num	1
\affixpstart@num	1
\affixside@note	1
\Afootnote	24
\afternoteX	42
\afterruleX	43
\ampersand	1, 48
\applabel	1, 52
\appref	1, 52
\apprefwithpage	1, 52
\arrangementX	1, 33
\arrangementX@normal	1
\arrangementX@threecol	1
\arrangementX@twocol	1
\at@every@pend	1
\AtEveryPend	1, 18
\AtEveryPstart	1, 18
\AtEveryStanza	1

\AtEveryStopStanza	1
\autopar	1, 17

B

\ballast	65
\ballast@count	1
Beeton, Barbara Ann Neuhaus Friend	18
\beforeedchapter	1
\beforeinsertingX	41
\beforenotesX	43
\beginnumbering	1, 16
\Bendnote	24
\Bfootnote	24
\bhookgroupX	42
\bhooknoteX	41
\bodyfootmarkA	31
\boxfootnotenumbers	1
Bredon, Simon	13
Breger, Herbert	13, 295
Brey, Gerhard	13
Busard, Hubert L. L.	13
\bypage@false	1
\bypage@true	1
\bypstart@false	1
\bypstart@true	1

C

\c@addcolcount	1
\c@ballast	1
\c@firstlinenum	1
\c@firstsublinenum	1
\c@labidx	1
\c@linenumincrement	1
\c@sublinenumincrement	1
\Cendnote	24
\Cfootnote	24
\ch@ck@l@ck	1
\ch@cksub@l@ck	1
\chapter	1
\check@pb@in@verse	1
Chester, Robert of	13
Claassens, Geert H. M.	13
\colalignX	41
Copernicus, Nicolaus	13
\critext	333
\ctab	1
\ctabtext	1

D

Dekker, Dirk-Jan	66
------------------	----

<code>\Dendnote</code>	24
<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@feetX</code>	1
<code>\do@insidelinehook</code>	1
<code>\do@line</code>	1
<code>\do@linehook</code>	1
<code>\do@lockoff</code>	1
<code>\do@lockoffL</code>	1
<code>\do@lockon</code>	1
<code>\do@lockonL</code>	1
<code>\doedindexlabel</code>	1
<code>\doendnotes</code>	1, 25
<code>\doendnotesbysection</code>	1, 25
<code>\doinsidelinehook</code>	1, 22
<code>\dolinehook</code>	1, 22
<code>\dosplits</code>	1
Downes, Michael	65, 161, 163
<code>\doextrafeet</code>	1
<code>\dummy@edtext</code>	1
<code>\dummy@edtext@showlemma</code>	1
<code>\dummy@ref</code>	1

E

<code>\edaftertab</code>	1, 60, 309
<code>edarrayc</code> (environment)	58
<code>edarrayl</code> (environment)	58
<code>edarrayr</code> (environment)	58
<code>\edatleft</code>	1, 60
<code>\edatright</code>	1, 60
<code>\edbeforetab</code>	1, 60, 309
<code>\edfilldimen</code>	1
<code>\edfont@info</code>	1
<code>\EDINDEX</code>	1
<code>\edindex</code>	1, 55
<code>\edindexlab</code>	1, 57
<code>\EDLABEL</code>	1
<code>\edlabel</code>	1, 49
<code>\edlabelE</code>	1, 51
<code>\edlabelS</code>	1, 51
<code>\edlabelSE</code>	1, 51
<code>\edlineref</code>	1, 49
<code>\edmakelabel</code>	1, 51

<code>\edpageref</code>	1, 49
<code>\edrowfill</code>	1, 59
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 58
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc</code> (environment)	58
<code>edtabularl</code> (environment)	58
<code>edtabularr</code> (environment)	58
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 23
<code>\edvertdots</code>	1, 60
<code>\edvertline</code>	1, 60
<code>\Eendnote</code>	24
<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, 21
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 16
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 21
<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	58
<code>edarrayl</code>	58
<code>edarrayr</code>	58
<code>edtabularc</code>	58
<code>edtabularl</code>	58
<code>edtabularr</code>	58
<code>ledgroup</code>	49

ledgroupsize	49
minipage	49
Euclid	13
\extensionchars	1, 63

F

\f@x@l@cks	1
Fairbairns, Robin	32
\first@linenum@out@false	1
\first@linenum@out@true	1
\firstlinenum	1, 19
\firstseriesX@	1
\firstsublinenum	1, 19
\firstXseries@	1
\fix@page	1
\flag@end	1
\flag@start	1
\flagstanza	1, 48
\flush@notes	1
\fnpos	1, 32
Folkerts, Menso	13
\footfootmarkA	31
\footfudgefiddle	1, 65
\footnote	1
\footnoteA	31
\footnoteB	31
\footnoteC	31
\footnoteD	31
\footnoteE	31
\footnotelang@lua	1
\footnotelang@poly	1
\footnoteoptions@	1
\footspitskips	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

<code>\h@num</code>	1
<code>\hangindentX</code>	40
<code>\hidenumbering</code>	1, 22
<code>\hidenumberingonleftpage</code>	1, 22
<code>\hidenumberingonrightpage</code>	1
<code>\Hilfsbox</code>	1
<code>\hilfsbox</code>	1
<code>\hilfscount</code>	1
<code>\HILFSskip</code>	1
<code>\Hilfsskip</code>	1
<code>\hilfsskip</code>	1
<code>\hsizethreecolX</code>	42
<code>\hsizetwocolX</code>	42
<code>\Hy@raisedlink@left</code>	1
<code>\hyperlinkformat</code>	1
<code>\hyperlinkformatR</code>	1
<code>\hyperlinkR</code>	1

I

<code>\if@addsw</code>	1
<code>\if@edindex@fornote@true</code>	1
<code>\if@edtext@secondarg@</code>	1
<code>\if@eled@sectioning</code>	1
<code>\if@firstlineofpage</code>	1
<code>\if@led@nofoot</code>	1
<code>\if@ledgroup</code>	1
<code>\if@lemmacommand@</code>	1
<code>\if@mldata@insertedfrompreviouspage</code>	1
<code>\if@noeled@sec</code>	1
<code>\if@noneed@Footnote</code>	1
<code>\if@RTL</code>	1
<code>\ifautopar@pause</code>	1
<code>\ifbypage@</code>	1
<code>\ifbypage@R</code>	1
<code>\ifbypstart@</code>	1
<code>\ifbypstart@R</code>	1
<code>\ifeledmacompat@</code>	1
<code>\iffirst@linenum@out@</code>	1
<code>\ifindtl@innote</code>	1
<code>\ifindtl@notenumber</code>	1
<code>\ifinserthangingsymbol</code>	1
<code>\ifinstanza</code>	1
<code>\ifl@d@dash</code>	1
<code>\ifl@d@elin</code>	1
<code>\ifl@d@esl</code>	1
<code>\ifl@d@pnum</code>	1
<code>\ifl@d@ssub</code>	1
<code>\ifl@d@Xmorethantwolines</code>	1
<code>\ifl@d@Xtwolines</code>	1

<code>\ifl@dend@X</code>	1
<code>\ifl@dhidenumber</code>	1
<code>\ifl@dmemoir</code>	1
<code>\ifl@dpaging</code>	1
<code>\ifl@dpairing</code>	1
<code>\ifl@dprintingcolumns</code>	1
<code>\ifl@dprintingpages</code>	1
<code>\ifl@dskipnumber</code>	1
<code>\ifl@dskipversenumber</code>	1
<code>\ifl@dstartendok</code>	1
<code>\ifl@imakeidx</code>	1
<code>\ifl@indextools</code>	1
<code>\ifledfinal</code>	1, 63
<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>\inserthangingymbol</code>	1
<code>\insertlines@list</code>	1
<code>\insertparafootsepX</code>	1
<code>\inserts@list</code>	1

J

Jayaditya 13

K

Kabelschacht, Alois 150

L

\l@advance@parledgroup@beforenormalnotes 1
 \l@d@add 1
 \l@d@nums 1
 \l@d@section 1
 \l@d@set 1
 \l@d@Xend 1
 \l@dampcount 1
 \l@dbfnote 1
 \l@dcheckcols 1
 \l@dcheckstartend 1
 \l@dchset@num 1
 \l@dcolcount 1
 \l@dcollect@body 1
 \l@dcollect@body 1
 \l@dcolwidth 1
 \l@dcsnote 1
 \l@dcsnotetext 1
 \l@dcsnotetext@l 1
 \l@dcsnotetext@r 1
 \l@ddodoreinextrafeet 1
 \l@dedbeginmini 1
 \l@dedendmini 1
 \l@demptyd@ta 1
 \l@dend@close 1
 \l@dend@open 1
 \l@dend@stuff 1
 \l@dend@Xfalse 1
 \l@dend@Xtrue 1
 \l@denvbody 1
 \l@desnote 1
 \l@dfambeginmini 1
 \l@dfamendmini 1
 \l@dfeetbeginmini 1
 \l@dfeetendmini 1
 \l@dgetline@margin 1
 \l@dgetlock@disp 1
 \l@dgetref@num 1
 \l@dgetsidenote@margin 1
 \l@dobblearg 1
 \l@disnote 1
 \l@dlabel@parse 1
 \l@dld@ta 1
 \l@dlp@rbox 1

<code>\ldlsn@te</code>	1
<code>\ldlsnote</code>	1
<code>\ldmake@labels</code>	1
<code>\ldmodforedtext</code>	1
<code>\ldnullfills</code>	1
<code>\ldnumpstartsL</code>	1
<code>\ldoldold@footnotetext</code>	1
<code>\ldp@rsefootspec</code>	1
<code>\ldparsedendline</code>	1
<code>\ldparsedendpage</code>	1
<code>\ldparsedendsub</code>	1
<code>\ldparsedstartline</code>	1
<code>\ldparsedstartpage</code>	1
<code>\ldparsedstartsub</code>	1
<code>\ldpush@begins</code>	1
<code>\ldrd@ta</code>	1
<code>\ldref@undefined</code>	1
<code>\ldrestorefills</code>	1
<code>\ldrestoreforedtext</code>	1
<code>\ldrp@rbox</code>	1
<code>\ldrsn@te</code>	1
<code>\ldrsnote</code>	1
<code>\ldsetmaxcolwidth</code>	1
<code>\ldskipnumberfalse</code>	1
<code>\ldskipnumbertrue</code>	1
<code>\ldtabaddcols</code>	1
<code>\ldtabnoexpands</code>	1
<code>\ldunboxmpfoot</code>	1
<code>\ldunhbox@line</code>	1
<code>\ldzeropenalties</code>	1
<code>\label</code>	51
<code>\labelpstartfalse</code>	1
<code>\labelpstarttrue</code>	1, 18
<code>\labelref@list</code>	1
<code>\labelrefsparseline</code>	1
<code>\labelrefsparsesubline</code>	1
<code>\last@page@num</code>	1
Lavagnino, John	12
<code>\led@check@nopb</code>	1
<code>\led@check@pb</code>	1
<code>\led@err@AutoparNotNumbered</code>	1
<code>\led@err@edtextoutsidepstart</code>	1
<code>\led@err@EdtextWithoutFootnote</code>	1
<code>\led@err@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

\led@err@NumberingNotStarted	1
\led@err@NumberingShouldHaveStarted	1
\led@err@NumberingStarted	1
\led@err@NumberingWithoutPstart	1
\led@err@PendNoPstart	1
\led@err@PendNotNumbered	1
\led@err@PstartInPstart	1
\led@err@PstartNotNumbered	1
\led@err@ReverseColumns	1
\led@err@toendnotes@outsidenumbering	1
\led@err@TooManyColumns	1
\led@err@UnequalColumns	1
\led@error@fail@patch@@docclearpage	1
\led@error@fail@patch@@iiiminipage	1
\led@error@fail@patch@@makecol	1
\led@error@fail@patch@@reinserts	1
\led@error@fail@patch@endminipage	1
\led@error@ImakeidxAfterEledmac	1
\led@error@IndextoolsAfterEledmac	1
\led@mess@NotesChanged	1
\led@mess@SectionContinued	1
\led@nopb	1
\led@nopbnum	1
\led@pb	1
\led@pb@setting	1
\led@pbnum	1
\led@reinit@index@fornote	1
\led@set@index@fornote	1
\led@toksa	1
\led@toksb	1
\led@warn@AppLabelOutSecondArgEdtext	1
\led@warn@BadAction	1
\led@warn@BadAdvancelineLine	1
\led@warn@BadAdvancelineSubline	1
\led@warn@BadLineation	1
\led@warn@BadLinenummargin	1
\led@warn@BadLockdisp	1
\led@warn@BadSetline	1
\led@warn@BadSetlinenum	1
\led@warn@BadSidenotemargin	1
\led@warn@BadSublockdisp	1
\led@warn@DuplicateLabel	1
\led@warn@LineFileObsolete	1
\led@warn@NoIndexFile	1
\led@warn@NoLineFile	1
\led@warn@NoMarginpars	1
\led@warn@RefUndefined	1
\led@warn@SeriesStillExist	1
\led@warning@hsizeX@deprecated	1
\led@warning@msdatawithoutstop	1

<code>\led@warning@preXnotes@deprecated</code>	1
<code>\led@warning@Xhsize@deprecated</code>	1
<code>ledgroup (environment)</code>	49
<code>ledgroupsize (environment)</code>	49
<code>\ledinnernote</code>	1, 54
<code>\ledinnote</code>	1
<code>\ledinnotehyperpage</code>	1
<code>\ledinnotemark</code>	1
<code>\ledleftnote</code>	1, 54
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 55
<code>\ledlsnotesep</code>	1, 55
<code>\ledlsnotewidth</code>	1, 54
<code>\lednopb</code>	1, 62
<code>\lednopbinversetrue</code>	63
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	54
<code>\ledouterote</code>	1
<code>\ledpb</code>	1, 62
<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 63
<code>\ledrightnote</code>	1, 54
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 55
<code>\ledrsnotesep</code>	1, 55
<code>\ledrsnotewidth</code>	1, 54
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 54
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 20
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	54
<code>\leftpstartnum</code>	1
<code>\leftfttab</code>	1
<code>Leibniz</code>	13
<code>\lemma</code>	1, 26
<code>\letsforverteilen</code>	1
<code>\line@list</code>	1
<code>\line@list@stuff</code>	1
<code>\line@list@version</code>	1
<code>\line@margin</code>	1
<code>\line@num</code>	1
<code>\line@set</code>	1
<code>\lineation</code>	1, 20
<code>\linenum</code>	1, 26
<code>\linenum@out</code>	1

<code>\linenumberlist</code>	1, 19
<code>\linenumberstyle</code>	1, 22
<code>\linenumincrement</code>	1, 19
<code>\linenummargin</code>	1, 20
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 20
<code>\linerangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 21
Lorch, Richard	13
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	69

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>	43
Mayer, Gyula	13
<code>\measurebody</code>	1
<code>\measuremcell</code>	1
<code>\measuremrow</code>	1
<code>\measuretbody</code>	1
<code>\measuretbody</code>	1
<code>\measuretbody</code>	1
<code>\measuretbody</code>	1
<code>\measuretbody</code>	1
Middleton, Thomas	13, 91
<code>minipage</code> (environment)	49
Mittelbach, Frank	12
<code>\morenoexpands</code>	1, 65
<code>\mpfnpos</code>	1, 32
<code>\mpnormalfootgroup</code>	1
<code>\mpnormalfootgroupX</code>	1
<code>\mpnormalvfootnote</code>	1
<code>\mpnormalvfootnoteX</code>	1
<code>\mppara@footgroupX</code>	1
<code>\mppara@vfootnoteX</code>	1
<code>\mpparafootgroup</code>	1
<code>\mpparavfootnote</code>	1
<code>\mpthreecolfootgroup</code>	1
<code>\mpthreecolfootgroupX</code>	1
<code>\mpthreecolfootsetup</code>	1

\mpthreecolfootsetupX	1
\mptwocolfootgroup	1
\mptwocolfootgroupX	1
\mptwocolfootsetup	1
\mptwocolfootsetupX	1
\msdata	1, 30
\multfootsep	1, 32
\multiplefootnotemarker	1

N

\n@num	1
\n@num@stanza	1
\new@line	1
\newhookarg@specific	1
\newhookcommand@series	1
\newhookcommand@series@reload	1
\newhooktoggle@series	1
\newhooktoggle@series@reload	1
\newhooktoggle@specific	1
\newseries@	1
\newverse	1
\NEXT	1
\no@expands	1
\noeledsec	62
\nomk@	1
\normal@footnotemarkX	1
\normal@page@break	1
\normal@pars	1
\normalbfnoteX	1
\normalbodyfootmarkX	1
\normalfootfmt	1
\normalfootfmtX	1
\normalfootfootmarkX	1
\normalfootgroup	1
\normalfootgroupX	1
\normalfootnoterule	1
\normalfootnoteruleX	1
\normalfootstart	1
\normalfootstartX	1
\normalvfootnote	1
\normalvfootnoteX	1
\notefontsizeX	40
\notenumfontX	39
\noteschanged@false	1
\noteschanged@true	1
\noteswidthliketwocolumnsX	44
\nulledindex	1
\nullsetzen	1
\num@lines	1
\numberedpar@false	1

<code>\numberedpar@true</code>	1
<code>\numberingfalse</code>	1
<code>\numberingtrue</code>	1
<code>\numberlinefalse</code>	19
<code>\numberlinetrue</code>	19
<code>\numberpstartfalse</code>	1, 18
<code>\numberpstarttrue</code>	1, 18
<code>\numberstanzafalse</code>	48
<code>\numberstanzatrue</code>	48
<code>\numlabfont</code>	1, 45

O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>optionauxdir</code>	15, 402
<code>optioncontinuousnumberingwithcolumns</code>	401
<code>optioninnnote</code>	400
<code>optioninnote</code>	400
<code>optionlinrangesep</code>	236
<code>optionnocritical</code>	400
<code>optionnoeledsec</code>	322, 402
<code>optionnoend</code>	400
<code>optionnopenalties</code>	65
<code>optionnotenumber</code>	400

P

<code>\page@action</code>	1
<code>\page@num</code>	1
<code>\pagelinesep</code>	1, 56
<code>\pageno</code>	1
<code>\pageref</code>	51
<code>\par@line</code>	1
<code>\para@footgroupX</code>	1
<code>\para@footsetup</code>	1
<code>\para@footsetupX</code>	1
<code>\para@vfootnoteX</code>	1
<code>\parafootfmt</code>	1
<code>\parafootfmtX</code>	1
<code>\parafootgroup</code>	1
<code>\parafootsepX</code>	42
<code>\parafootstart</code>	1
<code>\parafootstartX</code>	1
<code>\paravfootnote</code>	1
<code>\parindentX</code>	40
<code>\pausenumbering</code>	1, 18
<code>\pend</code>	1, 16
<code>Plato of Tivoli</code>	13
<code>\postbodyfootmark</code>	1
<code>\prebodyfootmark</code>	1
<code>\prenotesX</code>	43

<code>\prepare@edindex@fornote</code>	1
<code>\prepare@prenotesX</code>	1
<code>\prepare@Xprenotes</code>	1
<code>\prev@nopb</code>	1
<code>\prev@pb</code>	1
<code>\prevpage@num</code>	1
<code>\print@eledsection</code>	1
<code>\print@footnoteXrule</code>	1
<code>\print@leftmargin@eledsection</code>	1
<code>\print@lemma</code>	1
<code>\print@line</code>	1
<code>\print@notesX</code>	1
<code>\print@rightmargin@eledsection</code>	1
<code>\print@Xfootnoterule</code>	1
<code>\print@Xnotes</code>	1
<code>\printendlines</code>	1
<code>\printlineendnote</code>	1
<code>\printlineendnotearea</code>	1
<code>\printlinefootnote</code>	1
<code>\printlinefootnotearea</code>	1
<code>\printlinefootnotenumbers</code>	1
<code>\printlines</code>	1
<code>\printnpnum</code>	1
<code>\printpstart</code>	1
<code>\printsymlineendnotearea</code>	1
<code>\printsymlinefootnotearea</code>	1
<code>\printXafternumber</code>	1
<code>\printXbeforenumber</code>	1
<code>\pstart</code>	1, 16
<code>\pstarteref</code>	1
<code>\pstartnum</code>	1
<code>\pstartref</code>	49

Q

<code>\quotation</code>	1
<code>\quote</code>	1

R

<code>\raggedX</code>	42
<code>\raw@text</code>	1
<code>\rbracket</code>	1
<code>\read@linelist</code>	1
<code>\ref</code>	51
<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, 94
<code>\resetprevpage@</code>	1

\resetprevpage@num	95
\restore@familiarnotes	1
\restore@notes	1
\restore@sidenotes	1
\resumenumbering	1, 18
\rightctab	1
\rightlinenum	1, 20
\rightltab	1
\rightnoteupfalse	54
\rightrtab	1
\rightstartnum	1
\rigidbalance	1
\rigidbalanceX	1
\rtab	1
\rtabtext	1

S

Sacrobosco	13
\sameword	1, 27
\sameword@inedtext	1
Schöpf, Rainer	12
\section@num	1
\select@lemmafонт	1
\select@lemmafонт	1, 45
\SEref	1, 51
\SErefonlypage	51
\SErefwithpage	1, 51
\series	1
\seriesatbegin	1, 32
\seriesatend	1, 32
\set@line	1
\set@line@action	1
\set@Xtxtbeforenotes	1
\setapprefprefixmore	52
\setapprefprefixsingle	52
\setcommand@series	1
\sethangingsymbol	1, 47
\setistwofollowinglines	1
\setl@dlp@rbox	1
\setl@drpr@box	1
\setline	1, 21
\setlinenum	1, 22
\setmcellcenter	1
\setmcellleft	1
\setmcellright	1
\setmrowcenter	1
\setmrowleft	1
\setmrowright	1
\setmsdatalabel	1, 31
\setmsdataseries	1, 31

<code>\setnoteswidthliketwocolumnsX@</code>	1
<code>\setnotesXpositionliketwocolumns@</code>	1
<code>\setprintendlines</code>	1
<code>\setprintlines</code>	1
<code>\setSErefonlypageprefixmore</code>	52
<code>\setSErefonlypageprefixsingle</code>	52
<code>\setSErefprefixmore</code>	52
<code>\setSErefprefixsingle</code>	52
<code>\setsidenotessep</code>	55
<code>\setstanzaindents</code>	1, 45
<code>\setstanzapenalties</code>	1, 47
<code>\setstanzavalues</code>	1
<code>\settccllcenter</code>	1
<code>\settccllleft</code>	1
<code>\settccllright</code>	1
<code>\settoggle@series</code>	1
<code>\setthrowcenter</code>	1
<code>\setthrowleft</code>	1
<code>\setthrowright</code>	1
<code>\setXnotespositionliketwocolumns@</code>	1
<code>\setXnoteswidthliketwocolumns@</code>	1
<code>\showlemma</code>	1, 63
<code>\showwordrank</code>	1, 29
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 54
<code>\sidepstartnumtrue</code>	18
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 22
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 59
<code>\spreadtext</code>	1, 59
<code>\stanza</code>	1, 45
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 46
<code>\stanzaindent*</code>	1, 46
<code>\stanzaindentbase</code>	1, 45
<code>\stanzanumwrapper</code>	1, 48
<code>\startlock</code>	1, 21
<code>\startsub</code>	1, 21
<code>\stepl@dcclcount</code>	1
<code>\stopmsdata</code>	1, 30
<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

\sublinenumincrement	1, 19
\sublinenumr@p	1
\sublinenumrep	1
\sublineref	1, 49
\sublines@false	1
\sublines@true	1
\sublock@disp	1
\sublockdisp	1
Sullivan, Wayne	12, 13, 65, 78, 82, 161, 163, 242, 277
\sza@penalty	1

T

\tabHilfbox	1
\tabhilfbox	1
\theadcolcount	1
\theadtext	1
\theendpageline	1
\thefootnoteA	31
Theodosius	13
\thepageline	1
\thepstart	1, 18
\thestanza	1, 48
\thestartpageline	1
\this@line@list@version	1
\threecolfootfmt	1
\threecolfootfmtX	1
\threecolfootgroup	1
\threecolfootgroupX	1
\threecolfootsetup	1
\threecolfootsetupX	1
\threecolvfootnote	1
\threecolvfootnoteX	1
\toendnotes	25
\toendnotes*	1
\twocolfootfmt	1
\twocolfootfmtX	1
\twocolfootgroup	1
\twocolfootgroupX	1
\twocolfootsetup	1
\twocolfootsetupX	1
\twocolvfootnote	1
\twocolvfootnoteX	1

U

\unvxhX	1
---------	---

V

Vamana	13
\variab	1
\vbfnoteX	1

\vl@dbfnote	1
\vl@dcsnote	1
\vl@disnote	1
\vl@dlsnote	1
\vl@dosnote	1
\vl@drsnote	1
\vnumfootnoteX	1

W

Whitney, Ron	12
\widthX	44
\wrap@edcrossref	1
\wrapcontentX	40
\wrapped@bodyfootmarkX	1
\wrapped@footfootmarkX	1
Wujastyk, Dominik	12

X

\X@doreinfeet	1
\Xafterlemmaseparator	38
\Xafternote	42
\Xafternumber	36
\Xafterrule	43
\Xaftersymmlinenumber	36
\Xarrangement	1, 33
\Xarrangement@normal	1
\Xarrangement@paragraph	1
\Xarrangement@threecol	1
\Xarrangement@twocol	1
\Xbeforeinserting	41
\Xbeforelemmaseparator	38
\Xbeforenotes	43
\Xbeforenumber	34, 36
\Xbeforesymmlinenumber	36
\Xbhookgroup	42
\Xbhooknote	41
\Xboxlinenum	37
\Xboxlinenumalign	37
\Xboxsymmlinenumber	37
\Xcolalign	41
\Xdo@feet	1
\xedindex	1
\xedlabel	1
\xedtext	1
\Xendafterrenumber	36
\Xendafterlemmaseparator	39
\Xendafternote	44
\Xendafterpagenumber	38
\Xendaftersymmlinenumber	36
\Xendahookinplaceofnumber	38

\Xendahooklinenumber	38
\Xendbeforelemmaseparator	39
\Xendbeforenumber	36
\Xendbeforepagenumber	38
\Xendbeforesymlinenum	36
\Xendbhookinplaceofnumber	38
\Xendbhooklinenumber	38
\Xendbhooknote	41
\Xendboxendlinenumalign	37
\Xendboxlinenum	37
\Xendboxlinenumalign	37
\Xendboxstartlinenumalign	37
\Xendboxsymlinenum	37
\Xendhangindent	40
\Xendinplaceoflemmaseparator	39
\Xendinplaceofnumber	37
\Xendlemmadisablefontselection	39
\Xendlemmafont	39
\Xendlemmaseparator	39
\Xendlineprefixmore	38
\Xendlineprefixsingle	38
\Xendlinerrangeseparator	34
\Xendmorethantwolines	35
\Xendnonumber	35
\Xendnotefontsize	40
\Xendnotenumfont	39
\Xendnumberonlyfirstinline	34
\Xendnumberonlyfirstintwolines	34
\Xendparagraph	44
\Xendsep	44
\Xendsublinesep	36
\Xendsymlinenum	34
\Xendtwolines	35
\Xendtwolinesbutnotmore	35
\Xendwrapcontent	40
\xflagref	1
\Xhangindent	40
\Xsizethreecol	42
\Xsizetwocol	42
\Xinplaceoflemmaseparator	38
\Xinplaceofnumber	37
\Xinsertparafootsep	1
\Xledsetnormalparstuff	1
\xleft@appenditem	1
\Xlemmadisablefontselection	39
\Xlemmafont	39
\Xlemmaseparator	38
\Xlinerrangeseparator	34
\xlineref	1, 50
\Xmaxhnotes	43

\Xmorethantwolines	34
\Xnolemmaseparator	<u>1</u> , 38
\Xnonbreakableafternumber	36
\Xnonumber	35
\Xnotefontsize	40
\Xnotenumfont	39
\Xnoteswidthliketwocolumns	44
\Xnumberonlyfirstinline	34
\Xnumberonlyfirstintwolines	34
\Xonlypstart	36
\xpageref	<u>1</u> , 50
\Xparafootsep	42
\Xparindent	40
\Xprenotes	<u>1</u> , 43
\Xprenotes@	<u>1</u>
\Xpstart	35
\Xpstarteverytime	35
\xpstartref	<u>1</u> , 50
\XR@test	<u>1</u>
\XR@test@mac	<u>1</u>
\XR@test@mac@test	<u>1</u>
\Xragged	42
\xright@appenditem	<u>1</u>
\Xrigidbalance	<u>1</u>
\Xstanza	36
\Xstanzaseparator	36
\xsublineref	<u>1</u> , 50
\Xsublinesep	36
\Xsymlinenum	34
\Xtoendnotes	25
\Xtwolines	34
\Xtwolinesonlyinsamepage	35
\Xtxtbeforenotes	42
\Xunvxh	<u>1</u>
\Xwidth	44
\Xwrapcontent	40
\Xwrapendlemma	40
\Xwraplemma	40
\xxref	<u>1</u> , 51

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	70
\morenoexpands: Added \l@dtabnoexpands to \no@expands	117
\reledmac@error: Added \eledmac@error and replaced error messages	71
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	245
\doxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	237
\edlabel: Tweaked \edlabel to get correct page numbers	242
\l@ddodoreinxtrafeet: Renamed \dodoreinxtrafeet to \l@ddodoreinxtrafeet	238
\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	117
\zz@@@: Minor change to \zz@@@	242
General: Added text about normal labeling	51
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	240
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	160
\line@list@stuff: Added initial write of page number in \line@list@stuff	110
\para@footsetup: Added \footfudgefiddle to \para@footsetup	160
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	198
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	77
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	245
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	98
\ledlinenum: Added \linenumr@p and \sublinenum@rep to \leftlinenum and \rightlinenum	90
\linenumberlist: Added \linenumberlist mechanism	78
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	215
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	182
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	89
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	265
\Xarrangement@normal: Added minpage footnote setup to \footnormal	156
\edtext: Added \showlemma to \edtext	118
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	263

\mpnormalfootgroup: Added \mpnormalfootgroup	158
\mpnormalvfootnote: Added \mpnormalvfootnote	156
\showlemma: Added \showlemma	77
General: Added final/draft options	68
Added ledgroup environment	266
Added ledgroupsize environment	266
Added minipage, etc., support	1
v0.4.1.	
\Xdo@feet: Changed \Xdo@feet code for easier extensions	237
\edindex: Let eledmac take advantage of memoir's indexing	271
\print@Xnotes: Added \@opXfeet	237
General: Added code for changing \@doclearpage	240
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	184
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	184
\@xympar: Eliminated \marginpar disturbance	256
General: Added left and right side notes	256
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	142
\l@dotsidenote@margin: Added \sidenotemargin and \sidenote@margin	256
General: Added moveable side note	256
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	266
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	100
\@nl@reg: Added \fix@page to \@nl	98
Extended \@nl to include the page number	98
\fix@page: Added \last@page@num and \fix@page	100
\get@thisfootnote: Changed \l@dbfnote and \v1@dbfnote as originals could give incorrect markers in the footnotes	185
\new@line: Extended \new@line to output page numbers	111
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	98
\@ref@reg: Added \@ref@reg	108
\affixline@num: Added skipnumering to \affixline@num	142
\do@actions@fixedcode: Added \do@actions@fixedcode	141
\do@actions@next: Added number skipping to \do@actions	140
\do@insidelinehook: Added \do@linehook for use in \do@line	138
\endnumbering: Changed \endnumbering for eledpar	81
\fix@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fix@l@cks	145
\footplitskips: Added \footplitskips for use in many footnote styles	154
\get@linelistfile: Added \get@linelistfile	97
\initnumbering@reg: Added \initnumbering@reg	80

\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	265
\l@dcsnotetext@r: Added \l@emptyd@ta	138
\l@dgetline@margin: Added \l@dgetline@margin	86
\l@dgetlock@disp: Added \l@dgetlock@disp	88
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	256
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	78
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	139
\l@dzeropenalties: Added \l@dzeropenalties	133
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum ..	90
\line@list@stuff: Deleted \page@start from \line@list@stuff	110
\list@clearing@reg: Added \list@clearing@reg	97
\n@num: Added \n@num	105
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	190
\resumenumbering: Changed \resumenumbering for eledpar	82
\setprintendlines: Added \setprintendlines for use by \printendlines ...	213
\setprintlines: Added \setprintlines for use by \printlines	178
\skipnumbering: Added \skipnumbering and supports	114
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	87
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	89
Using \sublinenumrep instead of \sublinenumr@p	89
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	192
General: edmac having been available for 2 years, deleted the commented out original edmac texts	1
Maïeul Rouquette new maintainer	1
Made macros of all messages	71
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	78
General: For compatibility with eledpar, possibility to use \autopar on the right side. ...	1
Possibility to number \pstart.	18

Possibility to number the pstart with the commands <code>\numberpstarttrue</code> .	1
v0.12.1.	
General: Don't number <code>\pstarts</code> of stanza.	1
The numbering of <code>\pstarts</code> restarts on each <code>\beginnumbering</code> .	1
v0.13.0.	
<code>\managestanza@modulo</code> : New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	279
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	46
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.	242
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.	143
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	278
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code> .	181
General: Line numbering can be reset at each <code>pstart</code> .	84
Possibility to print <code>\pstart</code> number inside.	18
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	278
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	117
General: <code>\lemma</code> can contain commands.	26
Debug in lineation command	20
New generic commands to customize footnote display.	33, 228
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code> .	24
Options of <code>\Xfootnotes</code> .	152
Possibility to have commands in sidenotes.	54
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.	34
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code>	206
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	229
General: Add <code>\labelpstarttrue</code> .	18
Add <code>\Xnumberonlyfirstintwolines</code>	34
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	35
New hook to add arbitrary code at the beginning of the notes	41
New options for block of notes.	42
New package option: <code>parapparatus</code> .	1
New tools to change order of series	227
Sectioning commands.	61

v1.2.0.	
<code>\Xprenotes</code> :	Debug in familiar footnotes (bug introduced by v1.1). 206
<code>\endquote</code> :	Compatibility of <code>\ledchapter</code> with the <i>memoir</i> class. 314
v1.3.0.	
<code>\endquote</code> :	<i>Quotation</i> and quote environment inside numbered sections. 314
v1.4.0.	
<code>\edtext</code> :	Compatibility of <code>\edtext</code> with the right-to-left direction (with Polyglossia). 118
<code>\ledsetnormalparstuffX</code> :	Direction of footnotes with polyglossia. 202
<code>\newseries@</code> :	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. 219
<code>\rbracket</code> :	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). 173
General:	Compatibility with LuaTeX of RTL notes. 1
v1.4.1.	
<code>\affixside@note</code> :	Remove spurious spaces. 262
<code>\endquote</code> :	New option <i>noquotation</i> 314
<code>\get@thisfootnote</code> :	Compatibility of standard footnotes with <code>eledmac</code> when these footnotes contain any commands. 185
<code>\labelrefsparsesubline</code> :	Fix bug with <code>\edlabel</code> 244
v1.4.2.	
General:	Debug with some special classes. 1
v1.4.3.	
General:	Add <code>\Xnonbreakableafternumber</code> 36
Spurious space after familiar footnotes. 1
v1.4.4.	
General:	Label inside familiar footnotes. 1
v1.4.5.	
General:	Bug with <code>komasscript</code> + <code>eledpar</code> + <code>chapter</code> 1
v1.4.6.	
General:	Bug with <i>memoir</i> class introduced by 1.4.5. 1
v1.4.7.	
<code>\endquote</code> :	Compatibility of sectioning commands with <code>\autopar</code> 314
v1.4.8.	
General:	Corrects a bug with parallel texts introduced by 1.1. 1
v1.4.9.	
<code>\normalbfnoteX</code> :	Allow to redefine <code>\thefootnoteX</code> with <code>alph</code> when some packages are loaded. 190
v1.5.0.	
<code>\do@insidelinehook</code> :	Added <code>\do@insidelinehook</code> for use in <code>\do@line</code> 138
<code>\edindex</code> :	Compatibility with <code>imakeidx</code> package, and possibility to use multiple index with <code>\edindex</code> 271
General:	Correct indexing when the call is made in critical notes. 268
v1.5.1.	
<code>\managestanza@modulo</code> :	Correct <code>stanzaindentsrepetition</code> counter 279
<code>\normalvfootnoteX</code> :	Fix bug with normal familiar footnotes when mixing RTL and LTR text. 187
v1.6.0.	
<code>\newverse</code> :	Add <code>\falseverse</code> macro. 282
v1.6.1.	
<code>\AtEveryPstart</code> :	Spurious space in <code>\pstart</code> 131

\ifinserthangingsymbol: Hang verse is now not automatically flush right.	278
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill inside.	135
\pend: Spurious space in \pend.	132
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.	62
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts.	314
\get@index@command: Debug \get@index@command and compatibility with hyperref package.	270
\newhookcommand@series@reload: Debug \beforenotesX and \maxnotesX which did not work.	232
\prevpage@num: Correct \parafootsep when using with ledgroup.	166
General: Compatibility with parledgroup option ofeledpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	268
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	207
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.	268
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes.	237
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup.	263
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	314
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.	39
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.	98
\AtEveryPstart: New optional argument for \pstart, to execute code before it. . . .	131
\edindex: Use correctly default index when imakeidx is loaded.	271
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	314
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar. . .	80
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	256
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in eledmac, including needs for eledpar case.	258
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	78
\l@dunhbox@line: \do@line is split in more little commands.	136
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	232
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	232
\pend: New optional argument for \pend, to execute code after it.	132
\stanza: &can have an optional argument: content to be printed after.	282
\Stanza can have an optional argument: content to be printed before.	282
Add \newverse macro, \falseverse deprecated.	282
General: Add \ledinnernote and \ledouternote commands.	54
Add \Xendparagraph and related settings.	44
Add hyperlink to crossref (needs hyperref package).	49
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	268
New hooks: \Xafterrule and \afterruleX	43
New options for ragged-paragraph notes	42
New sectioning commands.	61
Optional arguments for \pstart and \pend.	18

v1.12.1.

\wrap@edcrossref: Fix spurious spaces.	247
--	-----

v1.12.2.

\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	135
--	-----

v1.12.3.

\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	112
\flag@start send a error message when a \edtext is done without insert (note)	112
\reledmac@error: Replaced error messages	71
General: Add macros for new messages since v0.7	71
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.	54
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	273
\@ssect: Debug \eledchapter in parallel with memoir	318
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	138
\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	81
\l@dgobblearg: \l@dgobblearg becomes \l@dgobbeloptarg	298
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	298
\resumenumbering: Debug \resumenumbering	82
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	247
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	112
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	231
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	198
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	229
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	44
Added widthliketwocolumns option	68
v1.13.1.	
\thepstart: Add \l@dzeropenalties in \pstart	131
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	78
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 \imakeidx even when memoir class is used	271
General: Debug spurious characters before endnotes.	207
Delete previous override of \l@d@@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	275
Move gobbling command	77
Provide \@gobblefour	77
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref	
package.	320
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	318
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	118
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	197
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	112
\if@lemmacommand@: New boolean \iflemmacommand@.	123

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	27
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.	110
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.	118
Compatibility of standard footnotes with some biblatex styles.	1
New <code>\stanzaindent</code> command.	1
v1.16.1.	
<code>\xlineref</code> : <code>\lineref</code> is not defined if defined by some other package, like <code>lineno</code> . Eledmac provides <code>\edlineref</code> instead.	247
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph. . . .	118
v1.18.0.	
<code>\@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools + hyperref</code>	273
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code> . .	242
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes.	208
<code>\l@dnumppstartsl</code> : Add <code>\ifl@dprintingpages</code> and <code>\dprintingcolumns</code> for <code>eledpar</code>	78
<code>\print@line</code> : Compatibility with Lua \TeX RTL languages.	136
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code>	174
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns (<code>eledpar</code>)	173
General: Add <code>\Xpstarteverytime</code>	35
Compatibility with Lua \TeX RTL languages.	1
Debug <code>\Xonlypstart</code> when using <code>\Xnumberonlyfirstinline</code> and the current line number differs from the previous.	35
v1.19.0.	
<code>\footssplitskips</code> : <code>\footssplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages.	154
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty. .	249
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>	182
<code>\stanza</code> : <code>&</code> and <code>&</code> can be preceded by spaces.	282
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (fix bug added in 1.19.0).	249
General: Add <code>\Xendboxlinenum</code>	37
Add <code>\Xtwolines</code> and <code>\Xmorethantwolines</code> hooks	34
Add series option.	1

Correct <code>\Xinplaceofnumber</code> hook.	1
Explicit error message when calling <code>\Xfootnote</code> outside of <code>\edtext</code>	1
Fix bug with line number typesetting direction when using <code>\eledsection</code> and similar commands for RTL texts with <code>Lua\TeX</code>	1
Fix issues with RTL text in notes when using <code>Lua\TeX</code>	1
Options <code>fulllines</code> in <code>\Xfootnote</code>	24
The <code>\newifs</code> are not followed by boolean values set to false, because it is the <code>\TeX</code> default setting.	1
v1.21.0.	
<code>\edindex@hyperref</code> : Look at the <code>hyperindex</code> option of <code>hyperref</code> before inserting <code>hyperref</code>	273
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	208
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code>	271
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code>	202
<code>\n@num</code> : <code>\n@num@ref</code> deleted	105
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	105
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	231
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	231
<code>\print@footnoterule</code> : Code refactoring: the spaces after the footnote rules are directly managed in <code>\print@Xfootnoterule</code> and <code>\print@footnoterule</code>	204
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	228
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	114
Correct <code>\skipnumbering</code> for stanza.	114
Delete <code>\skipnumbering@reg</code>	114
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	227
Add <code>\applabel</code> and related	52
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column.	1
Add <code>\hidenumbers</code>	22
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks	41
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code>	35
Add <code>\Xparindent</code> and <code>\hangindentX</code>	40
Add <code>\Xtwolinesbutnotmore</code> and <code>\Xtwolinesonlyinsamepage</code>	1
Add <code>nocritical</code> , <code>noend</code> , <code>nofamiliar</code> and <code>noledgroup</code> options.	1
Add <code>noeledsec</code> package option	1
Debug <code>\beforenotesX</code> <code>\maxhnotesX</code> <code>\noteswidthliketwocolumnsX</code> and <code>\afterruleX</code> with footnotes set in two and three columns.	1
Fix bug when a <code>\Xfootnote</code> follows a <code>\Xendnote</code> in the second argument of <code>\edtext</code> (bug added in <code>eledmac</code> 1.0.0).	1

Fix bug with <code>\maxnotesX</code> when using <code>\foottwocolX</code> or <code>\footthreecolX</code>	1
Fix bug with space between columns with notes in two columns (bug added in v1.13.0).	1
Fix spurious space after first page number in <code>\doendnotes</code> . <code>oldprintnpnumspace</code> option allows to come back to previous setting	1
<code>parapparatus</code> option works now with familiar footnotes.	1
Provide <code>\@gobblefive</code>	77
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	271
General: Add <code>\doendnotesbysection</code> command.	25
Add option for lemma separator inside endnotes	39
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>	166
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>	118
<code>\Serefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmoreethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code>	252
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	122
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	219
<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.	126
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code>	37
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code>	37
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.	27
Fix bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code>	27
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.	27
v1.23.1.	
General: Fix bug with <code>\lemma</code> command in the right side.	1
v1.23.2.	
General: Compatibility with \LaTeX 's release 2015.	1

v1.24.0.	
General: We can reinitialize <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> providing to it an empty argument.	1
v1.24.1.	
General: <code>\lemma</code> is disabled when using ‘ <code>nocritical</code> ’ option.	1
v1.24.2.	
General: Fix incompatibility between ‘ <code>nofamiliar</code> ’ option and ‘ <code>memoir</code> ’ package.	1
v1.24.3.	
General: Restore marginal numbers and notes with sectioning command (bug introduced in v1.21.0)	1
v1.24.4.	
General: Fix spurious space with <code>\edindex</code> when using <code>xindy+hyperref</code> option.	1
v1.24.5.	
General: Fix bug of indent, when a added in 1.1.0, when a <code>\beginnumbering</code> immediately follow a sectioning command.	1
v2.0.0.	
<code>\@iiiminipage</code> : Patch <code>\@iiiminipage</code> instead of redefining it.	265
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	256
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	265
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	314
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some code of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	265
<code>\newseries@</code> : One endnotes file by series.	224
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined	240
<code>\doxtrafeeti</code> becomes <code>\do@feetX</code> ; <code>\doxtrafeetii</code> becomes <code>\Xdo@feet</code> ; <code>\@opxtrafeeti</code> becomes <code>\@opfeetX</code> ; <code>\doreinxtrafeetii</code> becomes <code>\Xdoreinfeet</code> ; <code>\doreinxtrafeeti</code> becomes <code>\@doreinfeetX</code>	240
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	184
Suppress indexing command specific to <code>memoir</code>	271
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.	202
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>\Xendafterenumber</code> , <code>\Xendbeforemsymmlinenumber</code> , <code>\Xendaftersymmlinenumber</code> ,	
<code>\Xendboxsymmlinenumber</code> , <code>\Xendhangindent</code> , <code>\Xendbhooklinenumber</code> ,	
<code>\Xendahooklinenumber</code> , <code>\Xendbhookinplaceofnumber</code> ,	
<code>\Xendahookinplaceofnumber</code>	1
v2.2.1.	
General: Compatibility with TeX format 2015/10/01.	1
v2.2.2.	
General: Fix bug in <code>\sethangingsymbol</code>	1
Fix bug with old version of <code>etex</code>	1
v2.3.0.	
General: Disable empty lines as paragraph in stanza.	1
Fix compatibility of paragraphed footnotes with <code>bidif</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.	152
General: <code>\Xbhooknote</code> and <code>\bhooknoteX</code> work with notes in columns.	1
Fix bug of <code>\parindentX</code> and <code>\Xparindent</code> with two columns and three columns	
notes.	1
Fix bug with <code>\sameword</code> in right side.	1
Fix spurious space in two columns and three columns notes.	1
Fix spurious space when using optional argument of <code>\stanza</code> (introduced in v2.3.0).	1
New hooks: <code>\Xlinerrangeseparator</code> and <code>\Xendlinerrangeseparator</code>	34

Option linerangesep for critical footnotes and endnotes.	34
v2.4.1.	
General: Fix bug with \appref and \apprefwithpage (introduced in v2.4.0).	1
Fix bug with tabular environments when using babel or polyglossia languages	
that override \LaTeX \roman command, like Greek language.	1
Fix bug with tabular environments when using babel or polyglossia languages	
that override \LaTeX \roman command, like Greek.	1
v2.5.0.	
\SErefwithpage: Debug \setapprefprefixsingle	251
\edlabel: Fix bug when calling \edlabel in a footnotes of the rightside	242
\l@d@section: \endnotes take six arguments.	208
\printlines: \printlines takes an eighth argument: the line flag	181
\xlineref: \xlineref does not include anymore the side flag. Use \xflagref to get	
it. Not that \edlineref still contains the flag.	247
General: \apprefwithpage and \appref print double quotation mark when the label	
was not defined.	1
\apprefwithpage and \appref work with right side crossref.	1
\apprefwithpage works also when noend option is enabled.	1
\appref and \apprefwithpage can take linerangesep optional argument.	1
\edlabel works now in \Xfootnote.	1
\lemma can be used even when the nocritical is enabled.	1
Compatibility with new hook and tools of reledpar 2.6.0.	1
Fix spurious vertical space in astanza environment (reledpar)	1
Log now states ‘There were undefined references’ when using wrong references in	
\edlineref or edpageref.	1
New hooks to customize page and line number appearance in endnotes.	1
New hooks: \Xhookgroup and \hookgroupX.	1
New tools to easily make cross-reference to a passage defined by a start and an end line	
51	
v2.6.0.	
General: Adds compatibility with innnote and notenumber options of indextools	
package.	1
Fix bug with footnote counter in ledgroup (added in v2.5.0).	1
Fix bug, introduced in v2.5.0, with footnote numbering in parallel typesetting when	
using perpage package.	1
v2.7.0.	
\@k: \rigidbalance is split in \Xrigidbalance and \rigidbalanceX.	166
\l@d@section: \endnotes take seven arguments.	208
General: Add dash as default page range separator for \SEonlypage	1
Debug \SErefonlypage when referring to only one page.	1
Delete parenthesis after \SErefonlypage.	1
Fix (again) bugs with footnote numbering in parallel typesetting while using	
ledgroup environments (bug added in v2.5.0).	1
Fix bug with \SErefwithpage.	1
Fix bugs in compatibility with innote and notenumber options of indextools	
package, when indexing outside of a ledgroup.	1
New commands to make glossaries connected to page and linenumber with the	
glossaries package	1
New hooks: \Xhsize and \hsizeX	44
New hooks: \Xlemmafont and \Xendlemmafont	39

New setting commands: <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code>	1
Warning for duplicate and undefined labels are parsable by latexmk	1
Warning for duplicate labels does not send any more a false line and page number	1
When using <code>hyperref</code> package, add link in familiar footnotes between the footnote marks in the text and the footnote marks in the footnote	1
When using <code>hyperref</code> package, add links for <code>\SEref</code> and related, <code>\appref</code> and related.	1
When using <code>hyperref</code> package, add links from critical footnotes and critical endnotes to the line of text they refers	1
v2.7.1.	
General: Debug <code>\Xbhookgroup</code> hooks executed on columnar footnotes (moved to a larger group, to take effect).	1
v2.7.2.	
General: <code>\Xhsize</code> and <code>\hsizeX</code> become <code>\Xwidth</code> and <code>\widthX</code>	44
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.	208
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>	173

General: Add \AtEveryStanza and \AtEveryStopStanza.	1
Fix bug in \ledlsnotefontsetup and \ledrsnotefontsetup which could not handle \color command properly.	1
More specific error messages.	1
New hooks: \Xwrapcontent, \Xendwrapcontent and \wrapcontentX.	40
New hooks: \Xwraplemma and \Xendwraplemma	40
v2.10.1.	
General: Add ‘nopenalties’ option.	1
Fix bug introduced in v. 1.4: not paragraphed critical footnotes could prevent marginal line number from being displayed	1
v2.11.0.	
\do@actions@fixedcode: Add action 1010	141
General: Add new tools to produce an apparatus of manuscripts	1
Fix bug in \Xparafootsep in parallel typesetting	1
Make \parafootsepX work	1
Prevent \Xtxtbeforenotes hook from causing notes to go beyond the bottom margin	1
v2.12.0.	
General: \preXnotes becomes \Xprenotes (naming convention)	1
Add \hidenumberingonleftpage and \hidenumberingonrightpage	1
Add \toendnotes and related.	1
Add auxdir option.	1
Fix bug in critical and familiar footnotes when using uppercase letters with accent mark	1
Fix bug when using \chapter in optional argument of \pstart in parallel typesetting in combination with the noeledsec option.	1
Fix bug with \ledinnernote and \ledouternote in parallel typesetting	1
Fix bug with familiar footnote number in optional argument of \pstart or \pend in parallel typesetting	1
Fix spurious vertical space in \chapter when used as optional argument of \pstart in parallel typesetting.	1
Make endnote compatible with \sameword mechanism	1
More accurate message to control the position of \Xfootnote and \applabel in the L ^A T _E X code	1
v2.13.0.	
General: The version 2.13.0 never existed.	1
v2.13.1.	
General: In critical footnotes, the right side flag is printed only if asked explicitly with \Xlineflag (bug added in v. 2.5.0).	1
v2.13.2.	
General: Fix bug added in v. 11.2 which makes parallel typesetting could not work.	1