

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 v2.16.6, last revised 2016/11/14.

[†]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
5.5 Executing 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	30
6.4 Apparatus of Manuscripts	30
6.4.1 Marking sections of text	31
6.4.2 Layout of the apparatus of manuscripts	31
6.4.3 Settings	32
6.5 Familiar notes	32
6.5.1 Basic use	32
6.5.2 Customizing mark	32
6.5.3 Separator for multiple footnotes	32
6.6 Changing series	33
6.6.1 Create a new series	33
6.6.2 Delete series	33
6.6.3 Series order	33
6.7 Position of critical and familiar footnotes	33
7 Critical apparatus appearance	34
7.1 Notes arrangement in a series	34
7.2 Control line number printing	35
7.2.1 Print line number only at first time	35
7.2.2 Print page number only at first time	36
7.2.3 Arbitrary text before line number	36
7.2.4 Separator for line range	36
7.2.5 Abbreviate line range	37
7.2.6 Disable line number	37
7.2.7 Printing pstart number	37
7.2.8 Printing stanza number	38
7.2.9 Separator between line and subline numbers	38
7.2.10 Separator between page and line numbers	38
7.2.11 Space around number	38
7.2.12 Space around line symbol	39
7.2.13 Space in place of number	39
7.2.14 Boxing line number and line symbol	39
7.3 For endnotes	40
7.4 Arbitrary code around line number	40
7.5 Separator between the lemma and the note	40
7.5.1 For footnotes	40
7.5.2 For endnotes	41
7.6 Font style	41
7.6.1 For line number	41
7.6.2 For the lemma	41
7.6.3 For all notes	42
7.7 Wrapping notes	42
7.7.1 Wrapping lemmas	42
7.7.2 Wrapping contents	42

7.8 Indent of notes content	42
7.9 Arbitrary code at the beginning of notes	43
7.10 Arbitrary code before inserting note	43
7.11 Options for footnotes in columns	43
7.11.1 Alignment	43
7.11.2 Size of the columns	44
7.12 Options for paragraphed footnotes and notes grouped by line	44
7.12.1 Mark separation of notes	44
7.12.2 Ragged text	44
7.13 Options for block of notes	45
7.13.1 Grouping notes by line	45
7.13.2 Text before notes	45
7.13.3 Code before notes	45
7.13.4 Spacing	45
7.13.5 Rule	46
7.13.6 Maximum height	46
7.13.7 Width	46
7.14 Footnotes and the reledpar columns	47
7.15 Endnotes in one paragraph	47
8 Fonts	47
9 Verse	48
9.1 Basic	48
9.2 Define stanza indents	48
9.3 Repeating stanza indents	48
9.4 Manual stanza indent	49
9.5 Stanza breaking	49
9.6 Hanging symbol	50
9.7 Long verse and page break	50
9.8 Content before/after verses	50
9.9 Numbering stanza	50
9.10 Various tools	51
9.11 Notes on empty lines	51
10 Grouping	51
11 Cross referencing	52
11.1 Basic use	52
11.2 Cross-referencing to a critical note	52
11.3 Cross-referencing which return a number in any case	53
11.3.1 Cross-referencing in order to define line number of a critical note	53
11.4 Not automatic cross-referencing	53
11.5 Normal L ^A T _E X cross-referencing	54
11.6 References to start and end lines	54
11.6.1 Reference to main text lines	54

11.6.2 References to lines that are commented on in the apparatus	54
11.6.3 Settings	54
11.7 Compatibility with <code>xr</code> package	56
12 Side notes	56
12.1 Basics	56
12.2 Setting	57
12.2.1 Width	57
12.2.2 Vertical position	57
12.2.3 Distance to the main text	57
12.2.4 Font	57
12.2.5 Separator between notes	57
13 Indexing	57
13.1 Basics	57
13.2 Referring to critical notes	58
13.3 Separator between page and line numbers	58
13.4 Using <code>xindy</code>	59
13.5 Advanced setting	59
14 Glossary	60
14.1 Preamble setting	60
14.2 Commands	60
15 Tabular material	60
16 Sectioning commands	63
16.1 Sectioning commands without line numbers or critical notes	63
16.2 Sectioning commands with line numbering and critical notes	64
16.3 Optimization	65
17 Quotation environments	65
18 Page breaks	65
18.1 Control page breaking	65
18.2 Prevent page break in a long verses	65
19 Miscellaneous	66
19.1 Known and suspected limitations	66
19.1.1 Non-standard geometry	66
19.1.2 <code>floatrow</code> package compatibility	66
19.1.3 ‘No room for a new’	66
19.1.4 Marginal notes	67
19.1.5 Paragraph shape	67
19.1.6 Paragraphed footnotes	67
19.1.7 Use with other packages	68
19.1.8 Parallel typesetting	69

I Implementation overview	70
II Preliminaries	70
II.1 Links with original edmac	70
II.2 Package declaration	70
II.3 Package options	71
II.4 Loading packages	73
II.5 Compatibility with LuaTeX	73
II.6 Boolean flags	73
II.7 Messages	74
II.8 Gobbling	81
II.9 Miscellaneous commands	81
II.10 Prepare reledpar	81
II.11 Booleans provided by other optional packages which are required in any case	82
III Sectioning commands	82
IV List macros	86
V Line counting	88
V.1 Choosing the system of lineation	88
V.2 Line number margin	89
V.3 Line number initialization and increment	90
V.4 Line number locking	91
V.5 Line number style	92
V.6 Line number printing	93
V.7 Line number counters and lists	94
V.8 Line number locking counter	95
V.9 Line number associated to lemma	95
V.10 Reading the line-list file	99
V.11 Commands within the line-list file	101
V.12 Writing to the line-list file	113
VI Marking text for notes	119
VI.1 \edtext itself	120
VI.2 Substitute lemma	127
VI.3 Substitute line numbers	128
VI.4 Lemma disambiguation	128
VII Paragraph decomposition and reassembly	134
VII.1 Boxes, counters, \pstart and \pend	135
VII.2 Processing one line	140
VII.2.1 General process	140
VII.2.2 Process for “normal” line	140
VII.2.3 Process for line containing \eledsection command	142
VII.2.4 Hooks	143

VII.2.5 Sidenotes and marginal line number initialization	143
VIII Line and page number computation	144
IX Line number printing	147
X Pstart number printing in side	151
XI Restoring footnotes and penalties	152
XI.1 Add insertions to the vertical list	152
XI.2 Penalties	154
XI.3 Printing leftover notes	155
XI.4 Text before notes	156
XII Critical footnotes	157
XII.1 Fonts	157
XII.2 Individual note options	157
XII.3 Notes language	158
XII.4 General survey of the way we manage notes	159
XII.5 General setup	159
XII.6 Footnotes arrangement	160
XII.6.1 User level macro	160
XII.6.2 Normal footnote	161
XII.6.3 Paragraphed footnotes	166
XII.6.4 Columnar footnotes	173
XII.7 Critical notes presentation	180
XII.7.1 Font tools	180
XII.7.2 Pstart number in footnote	181
XII.7.3 Lemma printing	181
XII.7.4 Line number printing	182
XII.7.5 Footnote grouped by line	190
XIII Familiar footnotes	191
XIII.1 Adjacent footnotes	191
XIII.2 Regular footnotes for numbered texts	193
XIII.3 Footnote formats	195
XIII.4 Footnote arrangement	195
XIII.4.1 User level macro	195
XIII.4.2 Normal footnotes	196
XIII.4.3 Two columns footnotes	202
XIII.4.4 Three columns footnotes	204
XIII.4.5 Paragraphed footnotes	206
XIII.5 Wrapping footnote marks in hyperlink	210
XIV Code common to both critical and familiar footnote in normal arrangement	211

XV Footnotes' width for two columns	212
XVI Footnotes' order	213
XVII Footnotes' rule	213
XVIII Specific skip for first series of footnotes	214
XVIII.0.1 Overview	214
XVIII.0.2 User level command	214
XVIII.0.3 Internal commands	215
XIX Endnotes	216
XIX.1 Internal commands	216
XIX.2 User level commands	220
XIX.2.1 Inserting contents to endnotes	220
XIX.2.2 Printing endnotes	221
XX Generate series of notes	227
XX.1 Test if series is still existing	228
XX.2 Init specific to reledpar	228
XX.3 For critical footnotes	228
XX.3.1 Options	228
XX.3.2 Create inserts, needed to add notes in foot	229
XX.3.3 Create commands for critical apparatus, \Afootnote, \Bfootnote etc.	230
XX.3.4 Set standard display	232
XX.4 For familiar footnotes	233
XX.4.1 Options	233
XX.4.2 Create tools for familiar footnotes (\footnoteX)	233
XX.5 The endnotes	235
XX.5.1 The auxiliary file	235
XX.5.2 The main macro	236
XX.5.3 Tools	236
XX.5.4 Internal commands	237
XX.5.5 The options	237
XX.6 Init standards series (A,B,C,D,E)	239
XXI Setting series display	239
XXI.1 Change series order	239
XXI.2 Test series order	239
XXI.2.1 Get the first series	239
XXI.3 Series setting	240
XXI.3.1 General way of working	240
XXI.3.2 Tools to set options	240
XXI.3.3 Tools to generate options commands	242
XXI.3.4 Options for critical notes	243
XXI.3.5 Options for familiar notes	245

XXI.3.6 Options for endnotes	246
XXI.4 Hooks for a particular footnote	247
XXI.5 Alias	248
XXII Output routine	248
XXII.1 Extra footnotes output	248
XXII.2 Patching standard output's commands	252
XXIII Cross referencing	254
XXIII.1 Compatibility with xref	268
XXIV Side notes	269
XXV Minipages and such	276
XXVI Indexing	281
XXVI.1 Looking on package order	281
XXVI.2 Auxiliary macros for \edindex	282
XXVI.3 Code specific to \edindexin critical footnotes	283
XXVI.4 Analysis of command in indexed text	284
XXVI.5 Code for the formatted index	284
XXVI.6 Main code	285
XXVI.7 Hyperlink	286
XXVI.8 'innote' and 'notenumber' option of indextols package	289
XXVII Glossaries	290
XXVIII Verse	291
XXVIII.1 Hanging symbol management	292
XXVIII.2 Using & character	292
XXVIII.3 Code category setting	292
XXVIII.4 Stanza count and indent	293
XXVIII.5 Numbering stanza	294
XXVIII.6 Stanza number in note	295
XXVIII.7 Main work	296
XXVIII.8 Restore catcode and penalties	298
XXIX Apparatus of Manuscripts	298
XXIX.1 User level macro	298
XXIX.2 Setting macro	299
XXIX.3 Counters and lists	300
XXIX.4 Auxiliary file macros	300
XXIX.5 Action macro	301
XXIX.6 Inserting footnote	306
XXIX.7 Other	306

XXX Arrays and tables	306
XXX.1 Preamble: macro as environment	306
XXX.2 Tabular environments	310
XXX.2.1 Disabling and restoring commands	310
XXX.2.2 Counters, boxes and lengths	313
XXX.2.3 Tabular typesetting	317
XXX.2.4 Environments	328
XXXI Quotation's commands	329
XXXII Section's title commands	330
XXXII.1 Commands to disable some feature	330
XXXII.2 General overview	330
XXXII.3 \beforeeledchapter command	331
XXXII.4 Auxiliary commands	331
XXXII.5 Patching standard commands	332
XXXII.6 Main code of \eledxxx commands	337
XXXII.7 Macros written in the auxiliary file	340
XXXIII Page breaking or no page breaking depending of specific lines	342
XXXIV Long verse: prevents being separated by a page break	343
XXXV Tools for hyperref package	344
XXXVI Compatibility with eledmac	345
Appendix A Things to do when changing versions	348
Appendix A.1 Migrating from edmac to ledmac	348
Appendix A.2 Migration from ledmac to eledmac	349
Appendix A.3 Migration to eledmac 1.5.1	350
Appendix A.4 Migration to eledmac 1.12.0	350
Appendix A.5 Migration to eledmac 17.1	351
Appendix A.6 Migration to eledmac 1.21.0	351
Appendix A.6.1 \Xledsetnormalparstuffand\ledsetnormalparstuffX351	
Appendix A.6.2 Endnotes	351
Appendix A.7 Migration to eledmac 1.22.0	351
Appendix A.8 Migration to eledmac 1.23.0	351
Appendix A.9 Migration from eledmac to reledmac	352
Appendix A.9.1 Risk of 'no room for a new'	352
Appendix A.9.2 Multiple indices with memoir	352
Appendix A.9.3 Deprecated commands and options	352
Appendix A.9.4 \renewcommandreplaced by command	353
Appendix A.9.5 Commands the names of which have been changed	353
Appendix A.9.6 Endnotes	355
Appendix A.9.7 Z Series	355
Appendix A.9.8 Internal commands	355

Appendix A.10 Migration to <code>reledmac</code> 2.1.0	355
Appendix A.11 Migration to <code>reledmac</code> 2.1.3	355
Appendix A.12 Migration to <code>reledmac</code> 2.3.0	355
Appendix A.13 Migration to <code>reledmac</code> 2.4.0	356
Appendix A.14 Migration to <code>reledmac</code> 2.5.0	356
Appendix A.15 Migration to <code>reledmac</code> 2.7.0	356
Appendix A.16 Migration to <code>reledmac</code> 2.7.2	356
Appendix A.17 Migration to <code>reledmac</code> 2.8.0	356
Appendix A.18 Migration to <code>reledmac</code> 2.13.1	356
References	358
Index	358
Change History	405

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. 34). 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. 349.

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. 352.

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 7.13.6 p. 46.

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. 353).

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. 65).

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. 59).

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. 64).

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. 51). 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. 48) or `reledpar` (see 19.1.8 p. 69).

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`
`\numberpstartfalse`
`\thepstart`

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

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

`\sidepstartnumtrue`
`\labelpstarttrue`

On each `\beginnumbering` the numbering restarts.

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

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`
`\resumenumbering`

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

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:

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

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

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

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

5.3.3 Setting lineation reset

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

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

5.3.4 Setting line number margin

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

```
\linenummargin{left}
```

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

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

5.3.5 Other settings

`\leftlinenum` `\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 (7.2.9 p. 38 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. 48.

`\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.

5.5 Executing code at each line

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

6 Apparatus commands

6.1 Terminology

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

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

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

So :

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

6.2 Critical notes

6.2.1 The lemma

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

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

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

For example:

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

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

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

I am happy : <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>		1 Smith] Jones C, D.
<code>July 16, 1954.}</code>		
}		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. 33.

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

- `fulllines` to disable `\Xtwolines` and `\Xmorethantwolines` features for this note (cf. 7.2.5 p. 37).
- `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. 41).
- `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. 7.2.5 p. 37).
- `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. 41).

- `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. 41).

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

\linenum You can use `\linenum{ $\langle arg \rangle$ }` 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 $\langle lemma \rangle$ argument to `\edtext` is extremely long, you may run out of memory; here again you can specify a note with an abbreviated lemma using `\lemma` and `\linenum`. The numbers used in `\linenum` need not be entered manually; you can use the ‘x-’ symbolic cross-referencing commands below (11 p. 52) 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`.

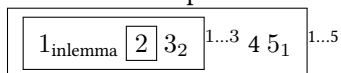
Note that only words that are actually referenced in a `\lemma` need the optional argument. Therefore, the first `\sameword` in the example above should have “1” as its optional argument, to be referenced correctly in the lemma.

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

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

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

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

Here, the corresponding code:

```

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

```

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

6.3.4 Customizing

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

```

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

```

6.4 Apparatus of Manuscripts

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

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

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

6.4.1 Marking sections of text

$\backslash\text{msdata}$ $\backslash\text{msdata}\{\langle\text{text}\rangle\}$ must be inserted at the point where a section for which only part of the manuscripts are relevant starts. $\langle\text{text}\rangle$ 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:

```
 $\backslash\text{msdata}\{\text{ABC}\}\text{Lorem ipsum}$ 
```

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

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

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

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.

$\backslash\text{stopmsdata}$ is also automatically inserted by $\backslash\text{msdata}$.

Note that in most cases, any $\backslash\text{stopmsdata}$ is followed by $\backslash\text{msdata}$. 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 $\backslash\text{stopmsdata}$, even if $\backslash\text{msdata}$ 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 $\backslash\text{msdata}$ is called at the very beginning of the page and $\backslash\text{endmsdata}$ 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. 34). 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}{\textsuperscript{\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` 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` 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, `\mpfnpos` it is possible to change the relative position of both types of footnotes. If you want to have familiar footnotes after critical footnotes in a normal page, use:

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

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

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

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

`<series><type>`

`<series>` is a series letter (A,B,C etc.), while `<type>` must be either “critical” or “familiar”.

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

```

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

```

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

7 Critical apparatus appearance

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

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

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

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

There is also name convention:

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

7.1 Notes arrangement in a series

`\Xarrangement`
`\arrangementX`

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

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

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

The value of `⟨a⟩` can be one of the following

- `paragraph` formats all of the footnotes of a series as a single paragraph; if you use this arrangement, you are strongly encouraged to read 19.1.6 p. 67.
- `twocol` formats them as separate paragraphs, but in two columns;
- `threecol`, in three columns.
- `normal`, restore normal arrangement.

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

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

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

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

7.2 Control line number printing

7.2.1 Print line number only at first time

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

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

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

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

7.2.2 Print page number only at first time

For endnotes, `reledmac` provides tools to print the page number only the first time it is seen. However, when a lemma spans over two pages, the line numbers are normally printed following this pattern: start page number - start line number - end page number - end line number. Consequently, what is the ‘page number’ is not obvious. So: `\Xendpagenumberonlyfirst` can be called to tell that the start page number of a lemma is not printed if it is equal to the end page number of the previous lemma. You can use *as complement* one (and only one) of the following commands:

- `\Xendpagenumberonlyfirstifsingle`[$\langle s \rangle$]: the first page number of the lemma will not be printed only if the following conditions are true:

1. The start page number of the lemma is equal to the end page number of the previous lemma.
2. The end page number of the lemma is equal to the start page number of the lemma.

In this case the end page number will always be printed if differing from the start page number.

- `\Xendpagenumberonlyfirstintwo`[$\langle s \rangle$]: both the start page number and the end page number of a lemma are not printed if they are both equal to respectively the start page number and the end page number of the previous lemma.

In any case, you can use:

`\Xendsympagenum`

- `\Xendsympagenum`[$\langle series \rangle$]{ $\langle c \rangle$ } to tell to print $\langle c \rangle$ when the page number is not printed.

`\Xendinplaceofpagenumber`

- `\Xendinplaceofpagenumber`[$\langle series \rangle$]{ $\langle l \rangle$ } to tell to print a horizontal space of $\langle l \rangle$ if no symbol is printed instead of page number.

7.2.3 Arbitrary text before line number

`\Xbeforenumber`

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

7.2.4 Separator for line range

`\Xlinerangeseparator`

`\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`[$\langle s \rangle$]{ $\langle text \rangle$ }, and with `\Xendlinerangeseparator`[$\langle s \rangle$]{ $\langle text \rangle$ } for critical endnotes.

7.2.5 Abbreviate line range

`\Xtwolines`
`\Xmorethantwolines`

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

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

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

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

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

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

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

`\Xtwolinesonlyinsamepage`

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

You can disable `\Xtwolines` and related for a specific note by using the ‘[fulllines]’ argument in the note macro cf. 6.2.2 p. 24.

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

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

7.2.6 Disable line number

`\Xnonumber`
`\Xendnonumber`

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

7.2.7 Printing pstart number

`\Xpstart`

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

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

`\Xpstarteverytime`

By default, the pstart number is printed only in the part of text where you have called `\numberpstarttrue`. We don’t know why you would like to print the pstart

number in the notes and not in the main text. However, if you want to do it, you can call `\Xpstarteverytime[⟨s⟩]`. In this case, the pstart number will be printed every time in footnote.

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

7.2.8 Printing stanza number

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

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

`\Xstanzaseparator`

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

7.2.9 Separator between line and subline numbers

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

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

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

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

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

7.2.10 Separator between page and line numbers

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

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

7.2.11 Space around number

`\Xbeforenumber` With `\Xbeforenumber[⟨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` and `\Xendafternumber` are the equivalents of `\Xbeforenumber` and `\afternumber` for endnotes.

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

7.2.12 Space around line symbol

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

7.2.13 Space in place of number

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

7.2.14 Boxing line number and line symbol

<code>\Xboxlinenum</code>	It could be useful to put the line number inside a fixed box: the content of the note will be printed after this box. You can use <code>\Xboxlinenum[⟨s⟩]{⟨l⟩}</code> to do that. To subsequently disable this feature, use <code>\Xboxlinenum</code> with length equal to 0 pt. One use of this feature is to print line number in a column, and the note in an other column: <pre>\Xhangindent{1em} \Xafternumber{0em} \Xboxlinenum{1em}</pre>
<code>\Xboxsymmlinenumber</code>	<code>\Xboxsymmlinenumber[⟨s⟩]{⟨l⟩}</code> is the same as <code>\Xboxlinenum</code> but for the line number symbol.
<code>\Xendboxsymmlinenumber</code>	<code>\Xendboxsymmlinenumber[⟨s⟩]{⟨l⟩}</code> is the same as <code>\Xboxsymmlinenumber</code> but for endnotes.
<code>\Xboxlinenumalign</code>	If you put line number in box, it will be aligned left inside the box. However, you can change it using <code>\Xboxlinenumalign[⟨s⟩]{⟨text⟩}</code> where <code>⟨text⟩</code> can be the following: L to align left (default value); R to align right; C to center.

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

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

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

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

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

7.3 For endnotes

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

7.4 Arbitrary code around line number

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

7.5 Separator between the lemma and the note

7.5.1 For footnotes

<code>\Xlemmaseparator</code> <code>\Xbeforelemmaseparator</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. Using <code>\Xbeforelemmaseparator[⟨s⟩]{⟨l⟩}</code> you can add some space between
--	---

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

lemma and separator. If your lemma separator is empty, this space won't be printed. The default value is 0 em.

`\Xafterlemmaseparator`

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

`\Xnolemmaseparator`

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

`\Xinplaceoflemmaseparator`

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

7.5.2 For endnotes

`\Xendlemmaseparator`

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

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

`\Xendbeforelemmaseparator`

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

`\Xendafterlemmaseparator`

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

`\Xinplaceoflemmaseparator`

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

7.6 Font style

7.6.1 For line number

`\Xnotenumfont`

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

`\Xendnotenumfont`

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

`\notenumfontX`

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

7.6.2 For the lemma

`\lemmadisablefontselection`

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

`\Xendlemmadisablefontselection`

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

`\Xlemmafont`
`\Xendlemmafont`

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

`\Xlemmafont{\bfseries}`

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

7.6.3 For all notes

`\Xnotefontsize`

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

`\notefontsizeX`

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

`\Xendnotefontsize`

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

7.7 Wrapping notes

7.7.1 Wrapping lemmas

`\Xwraplemma`

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

`\Xwrapendlemma`

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

7.7.2 Wrapping contents

`\Xwrapcontent`

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

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

`\Xendwrapcontent`

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

`\wrapcontentX`

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

7.8 Indent of notes content

`\Xparindent`

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

`\parindentX`

By default, `reledmac` does not add indentation before the paragraphs inside familiar

footnotes. Use `\parindentX[⟨s⟩]` to enable indentation.

`\Xhangindent` For critical notes NOT paragraphed you can define an indent with `\Xhangindent[⟨s⟩]{⟨l⟩}`, which will be applied in the second line of notes. It can help to make distinction between a new note and a break in a note. The default value is 0 pt.

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

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

7.9 Arbitrary code at the beginning of notes

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

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

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

7.10 Arbitrary code before inserting note

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

They allow one to add any arbitrary code just before the footnotes are added in the list of footnotes. The main use is to insert text direction code. For example, if you edit right-to-left text with `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. 42)).

Note that the changes are local to the footnote.

7.11 Options for footnotes in columns

7.11.1 Alignment

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

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

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

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

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

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

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

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

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

7.11.2 Size of the columns

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

<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 and notes grouped by line

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 Grouping notes by line

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

In many cases, you might like to use it in combination with `\Xnumberonlyfirstinline` (7.2.1 p. 35).

`\Xgroupbylineseparetwolines` Note that the `\Xafternote` and `\Xparafootsep` settings are used to determine space and content between footnotes (7.12 p. 44). Suppose you have two notes on line 1 which overlap lines 1 and 2. This last note will be printed, if you use `\Xgroupbyline` in the same group as the previous one. In the case you want that note to be distinct, you must use both `\Xgroupbyline` and `\Xgroupbylineseparetwolines[⟨s⟩]`.

In many cases, you might like to use it in combination with `\Xnumberonlyfirstintwolines` (7.2.1 p. 35)

7.13.2 Text before notes

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

7.13.3 Code before notes

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

7.13.4 Spacing

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

Be careful, the standard 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.

7.13.5 Rule

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

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

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

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

7.13.6 Maximum height

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

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

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

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

7.13.7 Width

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

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

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

Note that changes the with of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.11.2 p. 44.

7.14 Footnotes and the `reledpar` columns

`\Xnoteswidthliketwocolumns`
`\noteswidthliketwocolumnsX`

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

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

7.15 Endnotes in one paragraph

`\Xendparagraph`

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

`\Xendafternote`

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

`\Xendsep`

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

8 Fonts

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

`\numlabfont`

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

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

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

`\select@lemmafont`

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

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

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

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

9 Verse

9.1 Basic

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

9.2 Define stanza indents

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

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

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

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

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

If you want the hanging verse to be flush right, you can use `\sethanginsymbol`: see p. 9.6 p. 50.

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. 350.

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. 65 for further details.

9.8 Content before/after verses

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

- The `\stanza` command can take an optional argument (in brackets). Its content will be printed before the stanza.
Use `\AtEveryStanza{<arg>}` to automatically add content at the beginning of stanza.
- `&` can be replaced by `\newverse` with two optional arguments (in brackets). The first will be printed after the current verse, the second before the next verse.
Use `\AtEveryPend{<arg>}` to automatically add content after verses (including the final one) and `\AtEveryPstart{<arg>}` to automatically add content before verses (including the first one).
- `\&` can take an optional argument (in brackets). Its content will be printed after the stanza.
Use `\AtEveryStopStanza` to automatically add content at the end of stanzas.

9.9 Numbering stanza

`\numberstanzatrue`
`\numberstanzafalse`

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` (7.2.8 p. 38).

`\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. 50).

10 Grouping

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

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

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

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

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

```
\begin{ledgroupsize}[⟨pos⟩]{⟨width⟩}.
```

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

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

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

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

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

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

11 Cross referencing

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

11.1 Basic use

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

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

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

`\pstartref`

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

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

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

11.2 Cross-referencing to a critical note

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

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

If you want to refer to starting and ending lines, you should use `\appref` and related tools (11.6.2 p. 54).

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

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 \TeX is looking for a number, such a warning will lead to a complaint that the number is missing. This is the case for references used within the argument to `\linenum`, for example (see 6.2.5 p. 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 automatically 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 `\xlineref{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, and operate in the familiar fashion.

`\ref`

`\pageref`

11.6 References to start and end lines

11.6.1 Reference to main text lines

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

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

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

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

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

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

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

11.6.2 References to lines that are commented on in the apparatus

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

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

 If you use `\linenum` (6.2.5 p. 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.

`\apprefwithpage`

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.

`\setapprefprefixmore`

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 `\XendbeforepagenumberSerefonly` (see 11.6.3 p. 55). So if you use `\Xendbeforepagenumber` without any optional argument, it will override this setting.

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

These commands are the following:

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

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

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

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

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

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

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

12.2 Setting

12.2.1 Width

`\ledlsnotewidth` The left sidenote text is put into a box of width `\ledlsnotewidth` and the right 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 them 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`.

12.2.4 Font

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

These can of course be changed to suit.

12.2.5 Separator between notes

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

13 Indexing

13.1 Basics

`\edindex` \LaTeX provides the `\index{<item>}` command for specifying that `<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) \LaTeX three times, then run `makeindex`, and then finally run (Xe/Lua) \LaTeX again, in order to get an index with the right page numbers.

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

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

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

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

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

13.2 Referring to critical notes

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

```
The \edtext{creature\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-linenumber`

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

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

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

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

1. Use `xindy+hyperref` option when loading the `reledmac` package. When you run (Xe/Lua)TeX with this option, a `.xdy` configuration file will be generated with all the settings needed to allow internal hyperlinking in each index entry which is created by `\edindex`.
2. Use `hyperindex=false` option when loading `hyperref`.
3. Uncomment — by removing the semicolons at the beginning of the relevant lines — some lines in the `<code>.xdy</code>` file provided in the “examples” folder in order to restore internal links in the index to be used by the standard `index` command.²²

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:

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

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

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

`\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. 58).

14.2 Commands

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

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

15 Tabular material

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

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

`edarrayl`
`edarrayc`
`edarrayr`
`edtabularl`
`edtabularc`
`edtabularr`

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

1	2	3
a	bb	ccc
AAA	BB	C

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

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

For example:

```
\beginnumbering
\pstart
\begin{edtabularl}
\textbf{\Large I} & wish I was a little bug\edindex{bug} &
\textbf{\Large I} & eat my peas with honey\edindex{honey} \\
& With whiskers \edtext{round}{\Afootnote{around}} my tummy &
& I've done it all my life. \\
& I'd climb into a honey\edindex{honey} pot &
& It makes the peas taste funny \\
& And get my tummy gummy.\edindex{gummy} &
& But it keeps them on the knife.
\end{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{$}$` typesets `{$}$` but the `{$}$` has no effect on the
`\spreadtext` 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}
```

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

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

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

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

```
\begin{edtabularr}
```

```

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

```

1	2	3	4	5
Q		fd	h	qwertziohg
v	wptz	x	y	vb
g	nnn	$\overbrace{\hspace{10em}}$		
k	$\underbrace{\hspace{10em}}$		pq	dgh
1	2	3	co	ghweropjklmnbvcxys
			$\rule{10em}{0.4pt}$	

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	\lrule		d
A	B	C	D

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

```

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

```

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

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

For example:

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

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

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

`\edvertdots`

```
\begin{edarrayr}
a & & b & & C & & d & & & \\
v & & w & & x & & y & & & \\
m & & n & & o & & p & & & \\
k & & & & L & & cvb & & \edvertline{4pc} \\
\end{edarrayr}
```

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

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

16 Sectioning commands

16.1 Sectioning commands without line numbers or critical notes

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

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

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

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

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

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

16.3 Optimization

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

17 Quotation environments

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

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

18 Page breaks

18.1 Control page breaking

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

`\ledpb`
`\lednopb`

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

These commands have effect only at the second run.

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

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

18.2 Prevent page break in a long verses

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

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

19 Miscellaneous

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

19.1 Known and suspected limitations

19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.13.6 p. 46 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. 65 in order to know which are the available options.

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

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

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

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

19.1.4 Marginal notes

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

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

19.1.5 Paragraph shape

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

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

19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.1 p. 35, and described in more detail on XII.6.3 p. 168, really is a nuisance if that is something you need to do. There are some possible solutions, described by Michael Downes, but this area remains unsatisfactory.

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

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

`\footfudgefiddle`

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

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

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

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

Finally, in many cases you should use `\Xmaxhnotes` and / or `\maxhnotesX` (7.13.6 p. 46), in order to define the maximum height relative to `\textheight` and not to `\vsize`, because the `\vsize` value is not the same inside and outside of the preamble.

19.1.7 Use with other packages

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

`\morenoexpands`

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

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

If you actually try this²³ you will find \TeX whinging ‘Missing { inserted’, and then things start to fall apart. The trick in this case is to specify either:

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

or

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

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

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

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

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

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

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

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

19.1.8 Parallel typesetting

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

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

I Implementation overview

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

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

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

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

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

II Preliminaries

II.1 Links with original `edmac`

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

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

II.2 Package declaration

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

```

1 %<code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledmac}[2016/11/14 v2.16.6 typesetting critical editions]
4 %

```

II.3 Package options

```

\ifledfinal Use this to remember which option is used, set and execute the options with final as the
\ifnocritical@ default. We use xkeyval in order to manage options with argument.
\if@noeled@sec \RequirePackage{xkeyval}
\ifnoend@ %
\ifnofamiliar@
\ifnoledgroup@ The parledgroup option is for reledpar. However, it has consequence on reledmac
\ifparapparatus@ internal command. So we need to define the boolean now.
\ifnoquotation@ \newif\ifparledgroup
\iflednopbinverse %
\ifparledgroup
\ifwidthliketwocolumns And now, the options of reledmac.
\ifxindy@ \DeclareOptionX{series}[A,B,C,D,E]{\xdef\default@series{#1}}
\ifxindyhyperref@ \ExecuteOptionsX{series}%
\ifeledmaccompat@
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}
39
40 \newif\ifparapparatus@
41 \DeclareOptionX{parapparatus}{\parapparatus@true}
42
43 \newif\iflednopbinverse
44 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
45
46 \newif\ifwidthliketwocolumns%
47 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
48
49 \newif\ifcontinuousnumberingwithcolumns
50 \DeclareOptionX{continuousnumberingwithcolumns}{\
continuousnumberingwithcolumnstrue}%
51
52 \newif\ifxindy@
53 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
54   \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
55   \newwrite\eledmac@xindy@out%
56   \xindy@true%
57   \gdef\eledmacmarkuplocdepth{:depth 1}%
58   \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
59 }%
60
61 \newif\ifxindyhyperref@
62 \DeclareOptionX{xindy+hyperref}{%
63   \xindyhyperref@true%
64 }%
65
66 \newif\ifeledmaccompat@%
67 \DeclareOptionX{eledmac-compat}{%
68   \eledmaccompat@true%
69 }%
70 \DeclareOptionX{nopenalties}{%
71   \AtBeginDocument{\let\add@penalties\relax}%
72 }
73 \def\l@auxdir{}%
74 \DeclareOptionX{auxdir}{%
75   \xdef\l@auxdir{#1}/}%
76 }%
77 %

```

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

```

78 \ProcessOptionsX*\relax
79
80 %

```


II.4 Loading packages

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

```

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

```

II.5 Compatibility with Lua_{TeX}

Here, we enable some primitives for Lua_{TeX}.

```

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

```

II.6 Boolean flags

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

```

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

```

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

```

106 \newif\if@ledgroup%
107 %

```

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

```

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

```

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

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

II.7 Messages

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

\reledmac@warning Write a warning message.

```

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

```

`\reledmac@error` Write an error message.

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

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

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

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

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

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

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

```

\led@warn@NoLineFile 65 \newcommand*{\led@warn@NoLineFile}[1]{%
166 \reledmac@warning{Can't find line-list file #1}}
167 %

\led@warn@LineFileObsolete 68 \newcommand*{\led@warn@Obsolete}[1]{%
169 \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
170 %

\led@warn@BadAdvancelineSubline 71 \newcommand*{\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine 72 \reledmac@warning{\string\advanceline\space produced a sub-line
173 number less than zero.}}
174 \newcommand*{\led@warn@BadAdvancelineLine}{%
175 \reledmac@warning{\string\advanceline\space produced a line
176 number less than zero.}}
177 %

\led@warn@BadSetline 78 \newcommand*{\led@warn@BadSetline}{%
\led@warn@BadSetlinenum 79 \reledmac@warning{Bad \string\setline\space argument}}
180 \newcommand*{\led@warn@BadSetlinenum}{%
181 \reledmac@warning{Bad \string\setlinenum\space argument}}
182 %

\led@err@PstartNotNumbered 83 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart 84 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered 85 numbered section %
\led@err@PendNoPstart 86 (\string\...beginnumbering\string\endnumbering)}{\@ehc}}%
\led@err@AutoparNotNumbered 87 \newcommand*{\led@err@PstartInPstart}{%
\led@err@NumberingWithoutPstart 88 \reledmac@error{\string\pstart\space encountered while another
189 \string\pstart\space was in effect}{\@ehc}}
190 \newcommand*{\led@err@PendNotNumbered}{%
191 \reledmac@error{\string\pend\space must be used within a
192 numbered section}{\@ehc}}
193 \newcommand*{\led@err@PendNoPstart}{%
194 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
195 \newcommand*{\led@err@AutoparNotNumbered}{%
196 \reledmac@error{\string\autopar\space must be used within a
197 numbered section}{\@ehc}}
198 \newcommand*{\led@err@NumberingWithoutPstart}{%
199 \reledmac@error{\string\beginnumbering...\string\endnumbering\space
without \string\pstart}{\@ehc}}%
200 %

```

```

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

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

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

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

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

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

```

```

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

```

```

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

```

```

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

```

```

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

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

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

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

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

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

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

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

```

```

\led@error@fail@patch@endminipage305 \newcommand{\led@error@fail@patch@endminipage}{%
306 \reledmac@error{Fail to patch \string\endminipage\space command.}{\@ehc}%
307 }%
308 %

```

```

\led@error@fail@patch@endminipage309 \newcommand{\led@error@fail@patch@makeindex}{%
310 \reledmac@error{Fail to patch \string\makeindex\space command.}{\@ehc}%
311 }%
312 %

```

```

\led@warn@edinde@outsidenumbering313 \newcommand{\led@warn@edinde@outsidenumbering}{%
314 \reledmac@warning{\string\edindex\space called outside of \string\
...beginnumbering\string\endnumbering. \MessageBreak Automatically switched
to \string\index.}%
315 }%
316 %

```

```

\led@warning@hsizeX@deprecated317 \newcommand{\led@warning@hsizeX@deprecated}{%
318 \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
319 }%
320 %

```

```

\led@warning@Xhsize@deprecated321 \newcommand{\led@warning@Xhsize@deprecated}{%
322 \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
323 }%
324 %

```

```

\led@warning@msdatawithoutstop325 \newcommand{\led@warning@msdatawithoutstop}{%
326 \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
327 }%
328 %

```

```

\led@warning@preXnotes@deprecated329 \newcommand{\led@warning@preXnotes@deprecated}{%
330 \reledmac@warning@preXnotes@deprecated%
331 }%
332 %

```


II.8 Gobbling

Here, we define some commands which gobble their arguments.

```
\@gobblethree33 \providecommand*\@gobblethree}[3]{}
\@gobblefour34 \providecommand*\@gobblefour}[4]{}
\@gobblefive35 \providecommand*\@gobblefive}[5]{}
336 %
```

II.9 Miscellaneous commands

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

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

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

Initialize it as `\empty`.

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

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

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

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

II.10 Prepare reledpar

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

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

```

354 \newcount\l@dnumpstartsL
355 %

```

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

```

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

```

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

```

359 \newif\ifnumberingR
360 %

```

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

```

361 \newif\ifXnote@%
362 %

```

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

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

```

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

```

III Sectioning commands

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

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

```

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

```

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

```

370 \newif\ifnumbering
371 %

```

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

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

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

```

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

```

The tools for section's title commands are called:

- Define an empty list of pstart number where sectioning commands are called.

- Input auxiliary file with the description of section titles.
- Open the same auxiliary file to write in.

```

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

```

`\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.

```

423 \def\endnumbering{%
424   \ifnumbering
425     \global\numberingfalse

```

```

426 \normal@pars
427 \ifnum\l@dnumpstartsL=0%
428   \led@err@NumberingWithoutPstart%
429 \fi%
430 \ifl@dpairing
431   \global\pst@rtedLfalse
432 \else
433   \ifx\insertlines@list\empty\else
434     \global\noteschanged@true
435   \fi
436   \ifx\line@list\empty\else
437     \global\noteschanged@true
438   \fi
439 \fi
440 \ifnoteschanged@
441   \led@mess@NotesChanged
442 \fi
443 \else
444   \led@err@NumberingNotStarted
445 \fi
446 \autoparfalse
447 \if@noeled@sec\else%
448   \immediate\closeout\eled@sectioning@out%
449 \fi%
450 \ifl@dpairing\else
451   \global\l@dnumpstartsL=\z@%
452 \endgroup
453 \fi
454 }
455 %

```

\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

```

456 \newcommand{\pausenumbering}{%
457   \ifautopar\global\autopar@pausetrue\fi%
458   \endnumbering\global\numberingtrue}
459 %

```

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.

```

460 \newcommand*{\resumenumbering}{%
461   \ifnumbering
462     \ifautopar@pause\autopar\fi
463     \global\pst@rtedLtrue
464     \global\advance\section@num \@ne

```

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

```

465 \led@mess@SectionContinued{\the\section@num}%
466 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
467 \l@dend@stuff
468 \ifl@dpairing\else%
469   \begingroup%
470   \initnumbering@quote%
471   \ifwidthliketwocolumns%
472     \csuse{setwidthliketwocolumns@\columns@position}%
473     \csuse{setpositionliketwocolumns@\columns@position}%
474   \fi%
475 \fi%
476 \ifcontinuousnumberingwithcolumns%
477   \ifdefined\line@numR%
478     \ifnum\line@numR>\line@num%
479       \expandafter\setlinenum\expandafter{\the\line@numR}%
480     \fi%
481     \ifnum\last@page@numR>\last@page@num%
482       \global\last@page@num=\last@page@numR%
483     \fi%
484   \fi%
485 \fi%
486 \else
487   \led@err@NumberingShouldHaveStarted
488   \endnumbering
489   \beginnumbering
490 \fi}
491
492
493 %

```

IV List macros

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

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

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

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

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

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

initializing an empty list macro.

```

494 \newcommand*\list@create}[1]{%
495   \global\let#1=\empty%
496 }%
497 %

```

\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 .

```

498 \newcommand*\list@clear}[1]{%
499   \global\let#1=\empty%
500 }
501 %

```

\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 of a counter. **\led@toksb** `\xright@appenditem` creates global control sequences, like `\xdef`, and uses two temporary token-list registers, `\@toksa` and `\@toksb`.

```

502 \newtoks\led@toksa \newtoks\led@toksb
503 \global\led@toksa={\}
504 \long\def\xright@appenditem#1\to#2{%
505   \global\led@toksb=\expandafter{#2}%
506   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
507   \global\led@toksb={}}
508 %

```

\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`.

```

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

```

\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.

```

514 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
515 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
516
517 %

```

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

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

`\ifbypage@` • line-of-pstart: `bypstart@ = true` and `bypage@ = false`.

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

`\bypage@false`

```
518 \newif\ifbypage@
519 \newif\ifbypstart@
520 %
```

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@R21 \newif\ifbypage@R
\ifbypstart@R22 \newif\ifbypstart@R
523 %
```

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

```
524 \newcommand*{\lineation}[1]{%
525 %
```

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

```
526 \ifnumbering
527 \led@err@LineationInNumbered
528 \else
529 %
```

If the argument is page.

```
530 \def\@tempa{#1}\def\@tempb{page}%
531 \ifx\@tempa\@tempb
532 \global\bypage@true
533 \global\bypstart@false
534 \unless\ifnocritical@%
535 \Xpstart[] [false]%
536 \fi%
537 %
```

If the argument is pstart.


```

538 \else
539 \def\@tempb{pstart}%
540 \ifx\@tempa\@tempb
541 \global\bypage@false
542 \global\bystart@true
543 \unless\ifnocritical@%
544 \Xpstart%
545 \fi%
546 %

```

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

```

547 \else
548 \def\@tempb{section}
549 \ifx\@tempa\@tempb
550 \global\bypage@false
551 \global\bystart@false
552 \unless\ifnocritical@%
553 \Xpstart[] [false]%
554 \fi%
555 %

```

In other case, it is an error.

```

556 \else
557 \led@warn@BadLineation
558 \fi
559 \fi
560 \fi
561 \fi}}
562 %

```

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`

```

563 \newcount\line@margin%
564 \newcount\line@margin@columns%Only for parallel typesetting
565 \line@margin@columns=\m@ne%
566
567 \newcommand*{\linenummargin}[1]{%
568 \l@dgetline@margin{#1}%
569 \ifnum\@l@dttempcntb>\m@ne
570 \ifledRcol
571 \global\line@marginR=\@l@dttempcntb
572 \led@warn@setting@in@rightside{\linenummargin}%
573 \else
574 \global\line@margin=\@l@dttempcntb

```

```

575 \fi
576 \fi}}
577
578 \newcommand*{\l@dgetline@margin}[1]{%
579 \def\@tempa{#1}\def\@tempb{left}%
580 \ifx\@tempa\@tempb
581 \l@dtempcntb \z@
582 \else
583 \def\@tempb{right}%
584 \ifx\@tempa\@tempb
585 \l@dtempcntb \@ne
586 \else
587 \def\@tempb{outer}%
588 \ifx\@tempa\@tempb
589 \l@dtempcntb \tw@
590 \else
591 \def\@tempb{inner}%
592 \ifx\@tempa\@tempb
593 \l@dtempcntb \thr@@
594 \else
595 \led@warn@BadLinenummargin
596 \l@dtempcntb \m@ne
597 \fi
598 \fi
599 \fi
600 \fi}
601
602 %

```

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.

```

603 \newcounter{firstlinenum}
604 \setcounter{firstlinenum}{5}
605 \newcounter{linenumincrement}
606 \setcounter{linenumincrement}{5}
607 %

```

`\c@firstsublinenum`
`\c@sublinenumincrement`

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

```

608 \newcounter{firstsublinenum}
609 \setcounter{firstsublinenum}{5}
610 \newcounter{sublinenumincrement}

```

```

611 \setcounter{sublinenumincrement}{5}
612
613 %

```

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

```

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

```

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.

```

649 \newcount\lock@disp
650 \newcommand{\lockdisp}[1]{%
651   \l@getlock@disp{#1}%
652   \ifnum\@l@tempcntb>\m@ne
653     \global\lock@disp=\@l@tempcntb
654   \else
655     \led@warn@BadLockdisp
656   \fi}}
657 \newcommand*{\l@getlock@disp}[1]{
658   \def\@tempa{#1}\def\@tempb{first}%
659   \ifx\@tempa\@tempb
660     \@l@tempcntb \z@
661   \else
662     \def\@tempb{last}%
663     \ifx\@tempa\@tempb
664       \@l@tempcntb \@ne
665     \else
666       \def\@tempb{all}%
667       \ifx\@tempa\@tempb
668         \@l@tempcntb \tw@
669       \else
670         \@l@tempcntb \m@ne
671       \fi
672     \fi
673   \fi}
674
675 %

```

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

```

676 \newcount\sublock@disp
677 \newcommand{\sublockdisp}[1]{%
678   \l@getlock@disp{#1}%
679   \ifnum\@l@tempcntb>\m@ne
680     \global\sublock@disp=\@l@tempcntb
681   \else
682     \led@warn@BadSublockdisp
683   \fi}}
684
685 %

```

V.5 Line number style

`\linenumberstyle` We provide a mechanism for using different representations of the line numbers, not
`\linenumrep` just the normal arabic.
`\linenumr@p` NOTE: In v0.7 `\linenumrep` and `\sublinenumrep` replaced the internal `\linenumr@p`
`\sublinenumberstyle` and `\sublinenumr@p`.
`\sublinenumrep`
`\sublinenumr@p`

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

```

686 \newcommand*{\linenumberstyle}[1]{%
687   \def\linenumrep##1{\@nameuse{@#1}{##1}}
688 \newcommand*{\sublinenumberstyle}[1]{%
689   \def\sublinenumrep##1{\@nameuse{@#1}{##1}}
690 %

```

Initialise the number styles to arabic.

```

691 \linenumberstyle{arabic}
692 \let\linenumr@p\linenumrep
693 \sublinenumberstyle{arabic}
694 \let\sublinumr@p\sublinenumrep
695 %
696 %

```

V.6 Line number printing

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

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

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

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

```

697 \newlength{\linenumsep}
698 \setlength{\linenumsep}{1pc}
699 \newcommand*{\numlabfont}{\normalfont\scriptsize}
700 \newcommand*{\ledlinenum}{%
701   \bgroup%
702   \ifluatex%
703     \texdir TL%
704   \fi%
705   \numlabfont\linenumrep{\line@num}%
706   \ifsublines@
707     \ifnum\subline@num>0\relax
708       \unskip%
709       \Xsublinesep@side%
710       \sublinenumrep{\subline@num}%

```

```

711 \fi
712 \fi%
713 \egroup%
714 }%
715
716 \newcommand*{\leftlinenum}{%
717 \ledlinenum
718 \kern\linenumsep}
719 \newcommand*{\rightlinenum}{%
720 \kern\linenumsep
721 \ledlinenum}
722
723 %

```

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`.

```

724 \newcount\line@num
725 %

```

`\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.

```

726 \newcount\subline@num
727 %

```

`\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.

```
728 \newif\ifsublines@
729 %
```

\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.

```
730 \newcount\absline@num
731 %
```

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.

```
732 \newcount\@lock
733 \newcount\sub@lock
734 %
```

V.9 Line number associated to lemma

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

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

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

```
23|35|0|24|3|0|0T1/cmr/m/n.
```

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

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

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

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

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

The action code -1002 specifies the end of sub-lineation. At the next start-of-line, we should clear the sub-line counter and start advancing the line number.

The action codes for starting and ending sub-lineation are added to the list by the `\sub@action` macro, as called to implement the `\startsub` and `\endsub` macros.

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

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

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

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

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

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

Here are the commands to create these lists:

```

735 \list@create{\line@list}
736 \list@create{\insertlines@list}
737 \list@create{\actionlines@list}
738 \list@create{\actions@list}
739
740 %

```

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

```

741 \newcount\page@num
742 \newcount\endpage@num
743 \newcount\endline@num
744 \newcount\endsubline@num
745 %

```

`\ifnoteschanged@` If the number of the footnotes in a section is different from what it was during the last
`\noteschanged@true` run, or if this is the very first time you've run \LaTeX , on this file, the information from
`\noteschanged@false`

the line-list used to place the notes will be wrong, and some notes will probably be misplaced. When this happens, we prefer to give a single error message for the whole section rather than messages at every point where we notice the problem, because we do not really know where in the section notes were added or removed, and the solution in any case is simply to run \LaTeX two more times; there is no fix needed to the document. The `\ifnoteschanged@` flag is set if such a change in the number of notes is discovered at any point.

```
746 \newif\ifnoteschanged@
747 %
```

`\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@48 \newcommand*\resetprevline@{%
749   \def\do##1{\global\csundef{prevline##1}}%
750   \dolistloop{\@series}%
751 }
752 %
```

`\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@53 \newcommand*\resetprevpage@num{%
754   \def\do##1{%
755     \ifcsdef{prevpage##1@num}{%
756       \global\csname prevpage##1@num\endcsname=\z@%
757       \global\csname prevpage##1@numR\endcsname=\z@%
758     }%
759     {}%
760     \ifcsdef{##1prevpage@num}{%
761       \global\csname ##1prevpage@num\endcsname=\z@%
762       \global\csname ##1prevpage@numR\endcsname=\z@%
763     }%
764     {}%
765   }%
766   \dolistloop{\@series}%
767 }
768 %
```

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.

```

769 \newread\@inputcheck
770 \newcommand*{\read@linelist}[1]{%
771   \ifledRcol%
772     \list@clearing@regR%
773   \else%
774     \list@clearing@reg%
775   \fi%
776 %

```

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).

```

777 \list@clear{\maxlinesinpar@list}
778 %

```

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

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

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

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

```

779 \get@linelistfile{#1}%
780 \ifcontinuousnumberingwithcolumns
781   \global\page@numR=\page@numR\relax
782   \global\last@page@numR=\last@page@numR\relax
783   \global\page@num=\page@num\relax
784   \global\last@page@num=\last@page@num\relax
785 \fi
786 \@stopmsd%Security if last \endms{} is forgotten
787 \unless\ifledRcol%Get the last line of the last page
788   \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
789   \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
790 \else%
791   \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
792   \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%

```

```

793 \fi%
794 \endgroup
795 %

```

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.

```

796 \ifledRcol
797   \global\page@numR=\m@ne
798   \ifx\actionlines@listR\empty
799     \gdef\next@actionlineR{1000000}%
800   \else
801     \gl@p\actionlines@listR\to\next@actionlineR
802     \gl@p\actions@listR\to\next@actionR
803   \fi
804 \else
805   \global\page@num=\m@ne
806   \ifx\actionlines@list\empty
807     \gdef\next@actionline{1000000}%
808   \else
809     \gl@p\actionlines@list\to\next@actionline
810     \gl@p\actions@list\to\next@action
811   \fi
812 \fi
813 }
814 %

```

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

```

815 \newcommand*\list@clearing@reg{%
816   \list@clear{\line@list}%
817   \list@clear{\insertlines@list}%
818   \list@clear{\actionlines@list}%
819   \list@clear{\actions@list}%
820   \list@clear{\linesinpar@listL}%
821   \list@clear{\linesonpage@listL}%
822 }%
823 %

```

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

```

824 \newcommand*\get@linelistfile}[1]{%
825   \InputIfFileExists{\l@auxdir#1}{%
826     \global\noteschanged@false
827     \begingroup
828       \catcode`\[=1 \catcode`\]=2
829       \makeatletter \catcode`\^M=9}{%

```

```

830 \led@warn@NoLineFile{\l@auxdir#1}%
831 \global\noteschanged@true
832 \begingroup}%
833 }
834
835 %

```

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

V.11 Commands within the line-list file

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

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

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

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

```

836 \newcommand{\line@list@version}[1]{%
837   \IfStrEq{#1}{\this@line@list@version}%
838   {}%
839   {\ifledRcol%
840     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
841     \else%
842     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
843     \fi%
844     \endinput%
845   }%
846 }%
847 %

```

`\@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`.

```

848
849 \newcommand*{\@nl}[2]{%
850   \fix@page{#1}%
851   \ifledRcol%
852     \@nl@regR%
853   \else%
854     \@nl@reg%
855   \fi%
856 }
857 \newcommand*{\@nl@reg}{%
858   \ifx\l@dchset@num\relax \else
859     \advance\absline@num \@ne
860     \set@line@action
861     \let\l@dchset@num=\relax
862     \advance\absline@num \m@ne
863     \advance\line@num \m@ne
864   \fi
865   %

```

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

```

866   \advance\absline@num \@ne
867   \ifx\next@page@num\relax \else
868     \page@action
869     \let\next@page@num=\relax
870   \fi
871   \ifx\sub@change\relax \else
872     \ifnum\sub@change>\z@
873       \sublines@true
874     \else
875       \sublines@false
876     \fi
877   \sub@action
878   \let\sub@change=\relax

```

```

879 \fi
880 %

```

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

```

881 \ifcase\@lock
882 \or
883 \@lock \tw@
884 \or \or
885 \@lock \z@
886 \fi
887 \ifcase\sub@lock
888 \or
889 \sub@lock \tw@
890 \or \or
891 \sub@lock \z@
892 \fi
893 %

```

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

```

894 \ifsublines@
895 \ifnum\sub@lock<\tw@
896 \advance\subline@num \@ne
897 \fi
898 \else
899 \ifnum\@lock<\tw@
900 \advance\line@num \@ne \subline@num \z@
901 \fi
902 \fi}
903
904 %

```

`\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`.

```

905 \newcount\last@page@num
906 \last@page@num=-10000
907
908 \newcommand*{\fix@page}[1]{%
909 \ifledRcol
910 \ifnum #1=\last@page@numR
911 \else
912 \csnumgdef{\lastabsline@forpageR\the\page@numR}{\the\absline@numR}%
913 \csnumgdef{\lastline@forpageR\the\page@numR}{\the\line@numR}%
914 \ifbypage@R
915 \line@numR \z@ \subline@numR \z@
916 \fi
917 \page@numR=#1\relax
918 \last@page@numR=#1\relax
919 \def\next@page@numR{#1}%

```

```

920 \fi
921 \else
922   \ifnum #1=\last@page@num
923   \else
924     \csnumgdef{\lastabsline@forpage@the\page@num}{\the\absline@num}%
925     \csnumgdef{\lastline@forpage@the\page@num}{\the\line@num}%
926     \ifbypage@
927       \line@num \z@ \subline@num \z@
928     \fi
929     \page@num=#1\relax
930     \last@page@num=#1\relax
931     \def\next@page@num{#1}%
932     \listxadd{\normal@page@break}{\the\absline@num}
933   \fi
934 \fi}
935 %

```

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

```

936 \newcommand*{\@pend}[1]{}
937 \newcommand*{\@pendR}[1]{}
938 \newcommand*{\@lopL}[1]{}
939 \newcommand*{\@lopR}[1]{}
940
941 %

```

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

```

942 \newcommand*{\sub@on}{\ifsublines@
943   \let\sub@change=\relax
944   \else
945     \def\sub@change{1}%
946   \fi}
947 \newcommand*{\sub@off}{\ifsublines@
948   \def\sub@change{-1}%
949   \else
950     \let\sub@change=\relax
951   \fi}
952
953 %

```

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

```

954
955 \newcommand*{\@adv}[1]{}

```



```

956 \ifsublines@
957   \ifledRcol
958     \advance\subline@numR by #1\relax
959     \ifnum\subline@numR<\z@
960       \led@warn@BadAdvancelineSubline
961       \subline@numR \z@
962     \fi
963   \else
964     \advance\subline@num by #1\relax
965     \ifnum\subline@num<\z@
966       \led@warn@BadAdvancelineSubline
967       \subline@num \z@
968     \fi
969   \fi
970 \else
971   \ifledRcol
972     \advance\line@numR by #1\relax
973     \ifnum\line@numR<\z@
974       \led@warn@BadAdvancelineLine
975       \line@numR \z@
976     \fi
977   \else
978     \advance\line@num by #1\relax
979     \ifnum\line@num<\z@
980       \led@warn@BadAdvancelineLine
981       \line@num \z@
982     \fi
983   \fi
984 \fi
985 \set@line@action}
986
987 %

```

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

```

988
989 \newcommand*{\@set}[1]{%
990   \ifledRcol
991     \ifsublines@
992       \subline@numR=#1\relax
993     \else
994       \line@numR=#1\relax
995     \fi
996   \set@line@action
997 \else
998   \ifsublines@
999     \subline@num=#1\relax
1000   \else
1001     \line@num=#1\relax

```

```

1002 \fi
1003 \set@line@action
1004 \fi}
1005
1006 %

```

\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 `\@n1?` macro. If it is not `\relax` then a linenumber change is to be done.

```

1007
1008 \newcommand*{\l@d@set}[1]{%
1009 \ifledRcol
1010 \line@numR=#1\relax
1011 \advance\line@numR \@ne
1012 \def\l@dchset@num{#1}
1013 \else
1014 \line@num=#1\relax
1015 \advance\line@num \@ne
1016 \def\l@dchset@num{#1}
1017 \fi}
1018 \let\l@dchset@num\relax
1019
1020 %

```

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

```

1021
1022 \newcommand*{\page@action}{%
1023 \ifledRcol
1024 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1025 \xright@appenditem{\next@page@numR}\to\actions@listR
1026 \else
1027 \xright@appenditem{\the\absline@num}\to\actionlines@list
1028 \xright@appenditem{\next@page@num}\to\actions@list
1029 \fi}
1030 %

```

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

```

1031
1032 \newcommand*{\set@line@action}{%
1033 \ifledRcol
1034 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1035 \ifsublines@
1036 \@l@dttempcnta=-\subline@numR
1037 \else
1038 \@l@dttempcnta=-\line@numR

```

```

1039 \fi
1040 \advance\@l@tempcnta by -5000\relax
1041 \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1042 \else
1043 \xright@appenditem{\the\absline@num}\to\actionlines@list
1044 \ifsublines@
1045 \@l@tempcnta=-\subline@num
1046 \else
1047 \@l@tempcnta=-\line@num
1048 \fi
1049 \advance\@l@tempcnta by -5000\relax
1050 \xright@appenditem{\the\@l@tempcnta}\to\actions@list
1051 \fi}
1052 %

```

\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.

```

1053 \newcommand*{\sub@action}{%
1054 \ifledRcol
1055 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1056 \ifsublines@
1057 \xright@appenditem{-1001}\to\actions@listR
1058 \else
1059 \xright@appenditem{-1002}\to\actions@listR
1060 \fi
1061 \else
1062 \xright@appenditem{\the\absline@num}\to\actionlines@list
1063 \ifsublines@
1064 \xright@appenditem{-1001}\to\actions@list
1065 \else
1066 \xright@appenditem{-1002}\to\actions@list
1067 \fi
1068 \fi}
1069 %
1070

```

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

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

```

1071 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1072
1073 \newcommand*{\do@lockon}{%
1074 \ifx\next\lock@off

```

```

1075 \global\let\lock@off=\skip@lockoff
1076 \else
1077 \ifledRcol
1078 \do@lockonR
1079 \else
1080 \do@lockonL
1081 \fi
1082 \fi}
1083
1084
1085 \newcommand*{\do@lockonL}{%
1086 \xright@appenditem{\the\absline@num}\to\actionlines@list
1087 \ifsublines@
1088 \xright@appenditem{-1005}\to\actions@list
1089 \ifnum\sub@lock=\z@
1090 \sub@lock \@ne
1091 \else
1092 \ifnum\sub@lock=\thr@@
1093 \sub@lock \@ne
1094 \fi
1095 \fi
1096 \else
1097 \xright@appenditem{-1003}\to\actions@list
1098 \ifnum\@lock=\z@
1099 \@lock \@ne
1100 \else
1101 \ifnum\@lock=\thr@@
1102 \@lock \@ne
1103 \fi
1104 \fi
1105 \fi}
1106
1107 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1108 \newcommand*{\do@lockoffL}{%
1109 \xright@appenditem{\the\absline@num}\to\actionlines@list
1110 \ifsublines@
1111 \xright@appenditem{-1006}\to\actions@list
1112 \ifnum\sub@lock=\tw@
1113 \sub@lock \thr@@
1114 \else
1115 \sub@lock \z@
1116 \fi
1117 \else
1118 \xright@appenditem{-1004}\to\actions@list
1119 \ifnum\@lock=\tw@
1120 \@lock \thr@@
1121 \else

```

```

1122 \lock \z@
1123 \fi
1124 \fi}
1125
1126 \newcommand*{\do@lockoff}{%
1127 \ifledRcol
1128 \do@lockoffR
1129 \else
1130 \do@lockoffL
1131 \fi}
1132 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1133 \global\let\lock@off=\do@lockoff
1134
1135 %

```

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

```

1136 \newcommand*{\n@num}{%
1137 \ifledRcol%
1138 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1139 \xright@appenditem{-1007}\to\actions@listR
1140 \else%
1141 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1142 \xright@appenditem{-1007}\to\actions@list%
1143 \fi%
1144 }%
1145
1146 %

```

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

```

1147 \newcommand*{\n@num@stanza}{%
1148 \ifledRcol%
1149 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1150 \xright@appenditem{-1008}\to\actions@listR%
1151 \else%
1152 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1153 \xright@appenditem{-1008}\to\actions@list%
1154 \fi%
1155 }
1156 %

```

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

```

\hidenumberingonleftpage
\hidenumberingonrightpage
1157 \newif\ifl@dhidenumber
1158 \newcommand*{\hidenumbering}{

```

```

1159 \ifledRcol%
1160 \write\linenum@outR{\string\hide@num}%
1161 \else%
1162 \write\linenum@out{\string\hide@num}%
1163 \fi%
1164 }%
1165 \newcommand*{\hide@num}{%
1166 \ifledRcol%
1167 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1168 \xright@appenditem{-1009}\to\actions@listR%
1169 \else%
1170 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1171 \xright@appenditem{-1009}\to\actions@list%
1172 \fi%
1173 }
1174 \newcommand*{\hidenumberingonleftpage}{%
1175 \ifledRcol%
1176 \write\linenum@outR{\string\hide@num@left}%
1177 \else%
1178 \write\linenum@out{\string\hide@num@left}%
1179 \fi%
1180 }%
1181
1182 \newcommand*{\hide@num@left}{%
1183 \ifledRcol%
1184 \ifodd\page@numR\else%
1185 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1186 \xright@appenditem{-1009}\to\actions@listR%
1187 \fi%
1188 \else%
1189 \ifodd\page@num\else%
1190 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1191 \xright@appenditem{-1009}\to\actions@list%
1192 \fi%
1193 \fi%
1194 }%
1195
1196 \newcommand*{\hidenumberingonrightpage}{%
1197 \ifledRcol%
1198 \write\linenum@outR{\string\hide@num@right}%
1199 \else%
1200 \write\linenum@out{\string\hide@num@right}%
1201 \fi%
1202 }%
1203
1204 \newcommand*{\hide@num@right}{%
1205 \ifledRcol%
1206 \ifodd\page@numR%
1207 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1208 \xright@appenditem{-1009}\to\actions@listR%

```

```

1209 \fi%
1210 \else%
1211 \ifodd\page@num%
1212 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1213 \xright@appenditem{-1009}\to\actions@list%
1214 \fi%
1215 \fi%
1216 }%
1217
1218 %

```

\@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.

```

1219 \newcount\insert@count
1220 %

```

- #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.

```

1221 \newcommand*\@dummy@ref}[2]{#2}
1222 %

```

\@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.

```

1223 \newcommand*\@ref}[2]{%
1224 \ifledRcol%
1225 \@ref@regR{#1}{#2}%
1226 \else%
1227 \@ref@reg{#1}{#2}%
1228 \fi%
1229 }%
1230 \newcommand*\@ref@reg}[2]{%
1231 \global\insert@count=#1\relax
1232 \global\advance\@edtext@level by 1%
1233 \loop\ifnum\insert@count>\z@
1234 \xright@appenditem{\the\absline@num}\to\insertlines@list
1235 \global\advance\insert@count \m@ne
1236 \repeat
1237 %

```

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.

```

1238 \begingroup
1239   \let\@ref=\dummy@ref
1240   \let\@lopL\@gobble
1241   \let\page@action=\relax
1242   \let\sub@action=\relax
1243   \let\set@line@action=\relax
1244   \let\@lab=\relax
1245   \let\@lemma=\relax%
1246   \let\@sw\@gobblethree%
1247   #2
1248   \global\endpage@num=\page@num
1249   \global\endline@num=\line@num
1250   \global\endsubline@num=\subline@num
1251 \endgroup
1252 %

```

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

```

1253 \xright@appenditem%
1254   {\the\page@num|\the\line@num|%
1255    \ifsublines@ \the\subline@num \else 0\fi}%
1256   \the\endpage@num|\the\endline@num|%
1257   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1258 %

```

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

```

1259   \@ref@reg@parse{#2}%
1260 %

```

Decrease edtext level counter.

```

1261 \global\advance\@edtext@level by -1%
1262 }
1263 %

```

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

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

```

1264 \newcommand{\@ref@reg@parse}[1]{%
1265   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
1266   \@edtext@level\endcsname}%
1267   \providebool{lemmacommand@\the\@edtext@level}%
1268   \boolfalse{lemmacommand@\the\@edtext@level}%
1269 }

```


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

```
1269 #1%
1270 %
```

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.

```
1271 \ifnum\@edtext@level>0%
1272 \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1273 \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level
}{}%
1274 \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
1275 \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}%
1276 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1277 \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
1278 \fi%
1279 %

1280 }
1281
1282 %
```

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

```
1283 \newcommand{\@ref@later}[1]{%
1284 \global\advance\@edtext@level by \@ne%
1285 \ifledRcol%
1286 \@ref@reg@parseR{#1}%
1287 \else%
1288 \@ref@reg@parse{#1}%
1289 \fi%
1290 \global\advance\@edtext@level by -\@ne%
1291 }%
1292 %
```

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`.

```
1293 \newwrite\linenum@out
1294 %
```

```
\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.

```
1295 \newif\iffirst@linenum@out@
1296 \first@linenum@out@true
1297 %
```

```
\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.

```
1298 \newcommand{\this@line@list@version}{5}%
1299 %
```

```
\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.

```
1300 \newcommand*{\line@list@stuff}[1]{%
1301 %
```

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

```
1302 \read@linelist{#1}%
1303 %
```

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.

```
1304 \iffirst@linenum@out@
1305 \immediate\closeout\linenum@out%
1306 \global\first@linenum@out@false%
1307 \immediate\openout\linenum@out=\l@auxdir#1\relax%
1308 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1309 \ifl@dpaging%
1310 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
```

```

1311 \fi%
1312 \else
1313 %

```

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.

```

1314 \if@minipage%
1315 \leavevmode%
1316 \fi%
1317 \closeout\linenum@out%
1318 \openout\linenum@out=\l@auxdir#1\relax%
1319 \write\linenum@out{\string\line@list@version{\this@line@list@version}}%
1320 %
1321 \ifl@dpaging%
1322 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}%
1323 %
1324 \fi%
1325 \fi}
1326 %

```

\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.

```

1326 \newcommand*{\new@line}{%
1327 \IfStrEq{\led@pb@setting}{after}%
1328 {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1329 {\xifinlist{\the\absline@num}{\normal@page@break}%
1330 {\numgdef{\@next@page}{\c@page+\@ne}%
1331 \write\linenum@out{\string\@nl[\@next@page][\@next@page}}%
1332 }%
1333 {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1334 }%
1335 {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1336 }%
1337 \IfStrEq{\led@pb@setting}{before}%
1338 {\numgdef{\next@absline}{\the\absline@num+\@ne}%
1339 \xifinlist{\next@absline}{\l@prev@nopb}%
1340 {\xifinlist{\the\absline@num}{\normal@page@break}%
1341 {\numgdef{\nc@page}{\c@page+\@ne}%
1342 \write\linenum@out{\string\@nl[\nc@page][\nc@page}}%
1343 }%
1344 {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1345 }%
1346 {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1347 }%
1348 {}%
1349 \IfStrEqCase{\led@pb@setting}{\before}{\relax}{after}{\relax}}{\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1350 }

```

```
1351
1352 %
```

\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

```
1353 \newif\if@noneed@Footnote%
1354
1355 \newcommand*{\flag@start}{%
1356   \ifledRcol%
1357     \edef\next{\write\linenum@outR{%
1358       \string\@ref[\the\insert@countR] []}%
1359     \next%
1360     \ifnum\insert@countR<1%
1361       \if@noneed@Footnote\else%
1362         \led@err@EdtextWithoutFootnote%
1363       \fi%
1364     \fi%
1365   \else%
1366     \edef\next{\write\linenum@out{%
1367       \string\@ref[\the\insert@count] []}%
1368     \next%
1369     \ifnum\insert@count<1%
1370       \if@noneed@Footnote\else%
1371         \led@err@EdtextWithoutFootnote%
1372       \fi%
1373     \fi%
1374   \fi}%
1375
1376 \newcommand*{\flag@end}{%
1377   \ifledRcol%
1378     \write\linenum@outR{[]}%
1379   \else%
1380     \write\linenum@out{[]}%
1381   \fi}%
1382
1383
1384 %
```

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

\flag@end@later

```
1385 \newcommand*{\flag@start@later}{%
1386   \ifledRcol%
1387     \write\linenum@outR{\string\@ref@later []}%
1388   \else%
```

```

1389 \write\linenum@out{\string\@ref@later{}}%
1390 \fi%
1391 }%
1392 \newcommand{\flag@end@later}{%
1393 \ifledRcol%
1394 \write\linenum@outR{}}%
1395 \else%
1396 \write\linenum@out{}}%
1397 \fi%
1398 }
1399 %

```

\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.

```

1400
1401
1402 \newcommand*{\startsub}{\dimen0\lastskip
1403 \ifdim\dimen0>0pt \unskip \fi
1404 \ifledRcol \write\linenum@outR{\string\sub@on}%
1405 \else \write\linenum@out{\string\sub@on}%
1406 \fi
1407 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1408 \def\endsub{\dimen0\lastskip
1409 \ifdim\dimen0>0pt \unskip \fi
1410 \ifledRcol \write\linenum@outR{\string\sub@off}%
1411 \else \write\linenum@out{\string\sub@off}%
1412 \fi
1413 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1414
1415 %

```

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

```

1416 \newcommand*{\advanceline}[1]{\leavevmode%
1417 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1418 \else \write\linenum@out{\string\@adv[#1]}%
1419 \fi%

```

```
1420 }
1421 %
```

\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.

```
1422
1423 \newcommand*{\setline}[1]{%
1424   \leavevmode%
1425   \ifnum#1<\z@
1426     \led@warn@BadSetline
1427   \else
1428     \ifledRcol \write\linenum@outR{\string\@set[#1]}%
1429     \else      \write\linenum@out{\string\@set[#1]}%
1430     \fi
1431   \fi}
1432
1433 %
```

\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.

```
1434
1435 \newcommand*{\setlinenum}[1]{%
1436   \ifnum#1<\z@
1437     \led@warn@BadSetlinenum
1438   \else
1439     \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1440     \else      \write\linenum@out{\string\l@d@set[#1]} \fi
1441   \fi}
1442
1443 %
```

\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.

```
1444
1445 \newcommand*{\startlock}{%
1446   \ifledRcol \write\linenum@outR{\string\lock@on}%
1447   \else      \write\linenum@out{\string\lock@on}%
1448   \fi}
1449 \def\endlock{%
1450   \ifledRcol \write\linenum@outR{\string\lock@off}%
1451   \else      \write\linenum@out{\string\lock@off}%
1452   \fi}
1453 %
```

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

```

1454 \newif\ifl@dskipnumber
1455 \newif\ifl@dskipversenumber%
1456 \newcommand*{\skipnumbering}{%
1457   \leavevmode%
1458   \ifledRcol%
1459     \ifinstanza%
1460       \write\linenum@outR{\string\n@num@stanza}%
1461     \else%
1462       \write\linenum@outR{\string\n@num}%
1463     \fi%
1464     \advanceline{-1}%
1465   \else%
1466     \ifinstanza%
1467       \write\linenum@out{\string\n@num@stanza}%
1468     \else%
1469       \write\linenum@out{\string\n@num}%
1470     \fi%
1471     \advanceline{-1}%
1472   \fi%
1473 }%
1474
1475 %

```

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

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

VI.1 `\edtext` itself

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

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

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

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

```
1476 \list@create{\end@lemmas}
1477 %
```


`\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.

```
1478 \newcommand{\dummy@edtext}[2]{#1}
1479 %
```

`\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`.

```
1480 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1481 %
```

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

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

`\morenoexpands`

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

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

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

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

```

1482 \newcommand*{\no@expands}{%
1483   \let\select@lemmafnt=0%
1484   \let\startsub=\relax \let\endsub=\relax
1485   \let\startlock=\relax \let\endlock=\relax
1486   \let\edlabel=\@gobble
1487   \let\setline=\@gobble \let\advanceline=\@gobble
1488   \let\sameword\sameword@inedtext%
1489   \let\edtext=\dummy@edtext
1490   \l@dtabnoexpands
1491   \morenoexpands}
1492 \let\morenoexpands=\relax
1493
1494 %

```

\@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.

```

1495 \newcommand{\@tag}{}
1496 %

```

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

```

1497 \newcount\@edtext@level%
1498 \@edtext@level=0%
1499 %

```

\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.

```

1500 \newif\if@edtext@secondarg%
1501 %

```

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

```

1502 \newcounter{edtext}
1503 \renewcommand{\theedtext}{edtxt@{\arabic{edtext}}}%
1504 %

```

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

```
1505 \newcommand{\edtext}[2]{\leavevmode%
1506 %
```

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

```
1507 \ifnumberedpar%
1508 %
```

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

```
1509 \global\advance\@edtext@level by 1%
1510 %
```

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

```
1511 \stepcounter{edtext}%
1512 %
```

By default, we do not use `\lemma`

```
1513 \global\@lemmacommand@false%
1514 %
```

```
1515 \begingroup%
1516 %
```

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`.

```
1517 \ifledRcol%
1518 \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1519 {\global\let\sw@inthisedtext\empty}%
1520 {\expandafter\glp\cswname sw@list@edtextR@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1521 \else%
1522 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1523 {\global\let\sw@inthisedtext\empty}%
1524 {\expandafter\glp\cswname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1525 \fi%
1526 %
```

\@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.

```

1527 \global\renewcommand{\@tag}{%
1528 \no@expands #1%
1529 }%
1530 %

```

`\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`.

```

1531 \set@line%
1532 %

```

`\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`.

```

1533 \ifledRcol \global\insert@countR \z@%
1534 \else \global\insert@count \z@ \fi%
1535 %

```

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.

```

1536 \@edtext@secondarg@true%
1537 \ignorespaces #2\relax%
1538 \@edtext@secondarg@false%
1539 %

```

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

```

1540 \ifundefined{xpg@main@language}{%if not polyglossia
1541 \flag@start}%
1542 {\if@RTL\flag@end\else\flag@start\fi%
1543 }%
1544 %

```

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

```

1545 \if@lemmacommand%
1546 \ifledRcol%
1547 \write\linenum@outR{\string\@lemma}%
1548 \else%
1549 \write\linenum@out{\string\@lemma}%
1550 \fi%
1551 \fi%
1552 %

```

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.

```

1553     \endgroup%
1554     \ifdef{\hypertarget}%
1555         {%
1556             \csedef{thisedtext@the\@edtext@level}{\theedtext}%We need one
macro by level, as #1 can contain new \edtext
1557             \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}{}}%
1558             \showlemma{#1}%
1559             \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}%
1560         }%
1561         {%
1562             \showlemma{#1}%
1563         }%
1564     %

```

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.

```

1565     \ifx\end@lemmas\empty \else%
1566         \gl@p\end@lemmas\to\x@lemma%
1567         \x@lemma%
1568         \global\let\x@lemma=\relax%
1569     \fi%
1570     \ifundefined{xpg@main@language}{%if not polyglossia
1571         \flag@end}%
1572         {%ifRTL\flag@start\else\flag@end\fi% With polyglossia, you must
track whether the language reads left to right (English) or right to left
(Arabic).
1573         }%
1574     %

```

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`.

```

1575     \global\@noneed@Footnotefalse%
1576     \global\advance\@edtext@level by -1%

```

```
1577 \global\@lemmacommand@false%
1578 %
```

We also reset `\@beforeinsertofthistext`

```
1579 \global\let\@beforeinsertofthistext\relax%
1580 %
```

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

```
1581 \else%
1582 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidestart%
1583 \fi%
1584 }%
1585
1586
1587 %
```

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

```
1588 \let\@beforeinsertofthistext\relax
1589 %
```

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

```
1590 \newif\ifnumberline
1591 \numberlinetrue
1592 %
```

`\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.

```
1593 \newcommand*{\set@line}{%
1594 \iflabeledRcol
1595 \ifx\line@listR\empty
1596 \global\noteschanged@true
1597 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1598 \else
1599 \glp\line@listR\to\tempb
1600 \xdef\l@d@nums{\@tempb|\edfont@info}%
```

```

1601 \global\let\@tempb=\undefined
1602 \fi
1603 \else
1604 \ifx\line@list\empty
1605 \global\noteschanged@true
1606 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1607 \else
1608 \gl@p\line@list\to\@tempb
1609 \xdef\l@d@nums{\@tempb|\edfont@info}%
1610 \global\let\@tempb=\undefined
1611 \fi
1612 \fi}
1613
1614 %

```

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

```

1615 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1616
1617 %

```

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. 123).

```

1618 \newcommand*{\lemma}[1]{%
1619 \global\@lemmacommand@true%
1620 \global\renewcommand{\@tag}{%
1621 \no@expands #1%
1622 }%
1623 \ignorespaces%
1624 }%
1625 %

```

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

```

1626 \newcommand{\@lemma}{%
1627 \booltrue{lemmacommand@the\@edtext@level}%
1628 }%
1629 %

```

\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.

```

1630 \newif\if@lemmacommand@%
1631 %

```

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. 95): 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.

```
1632 \newcommand*{\linenum}[1]{%
1633   \xdef\@tempa{#1|}|}|}|}|}|noexpand\\l@d@nums}%
1634   \global\let\l@d@nums=\empty
1635   \expandafter\line@set\@tempa|\\ignorespaces}
1636 %
```

\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.

```
1637 \def\line@set#1|#2|#3|#4|\\{%
1638   \gdef\@tempb{#1}%
1639   \ifx\@tempb\empty
1640     \l@d@add{#3}%
1641   \else
1642     \l@d@add{#1}%
1643   \fi
1644   \gdef\@tempb{#4}%
1645   \ifx\@tempb\empty\else
1646     \l@d@add{|}\line@set#2|#4|\\%
1647   \fi}
1648 %
```

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

```
1649 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1650
1651 %
```

VI.4 Lemma disambiguation

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

The general mechanism is the following:

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

1. Get the rank of each `\sameword` in a line (relative rank) from the rank of each `\sameword` in all the numbered section (absolute rank):

- For each paired `\sameword` argument and absolute line number, a counter is defined. Its value corresponds to the number of times `\sameword{<argument>}` is called from the beginning of the lineation to the end of the current line. We also store the same data for the preceding absolute line number, if it does not have `\sameword{<argument>}`.
- For each `\sameword` having the same argument, we subtract from its absolute rank the number stored for the paired `\sameword` argument and previous absolute line number. Consequently, we obtain the relative rank.
- See the following example which explains how, for same `\sameword`, absolute ranks are transformed to relative ranks.

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

2. Create lists of lists of `\sameword` by depth of `\edtext`. That is: create a list for `\edtexts` of level 1, a list for `\edtexts` of level 2, a list for `\edtexts` of level 3 etc. For each `\edtext` in these lists, we store all of the relative ranks of `\sameword` which are called as lemma information. That is: 1) either called in the first argument of `\sameword`, or, 2) called in the `\lemma` macro of the second argument of `\sameword` AND marked by the optional argument of `\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 its `\edtext` level. So, in the previous example:
 - Critical notes (A) and (B) are associated with $\{1, 2, 2, 3\}$.
 - Critical note (C) is associated with $\{1, 2, 2, 3, 4, 3\}$.
 - Critical note (D) is associated with $\{5\}$.
 - Critical note (E) is associated with $\{5, 4\}$.
- At the second run, when a critical note is printed:
 - The `\sameword` command is let `\sameword@inedtext`.
 - At each call of this `\sameword@inedtext`, we step to the next element of the list associated to the note. Let it be r .
 - For the word marked by `\sameword`, we calculate how many time it is called in its line. To do it:
 - * We get the absolute line number of the current `\sameword`. This absolute line number was stored with a list of relative ranks for the current `\edtext`. That means, in the previous example, that if the absolute line number of `\edtext` was 1, that critical notes (A) and (B) were not associated with $\{1, 2, 2, 3\}$ but with $\{(1, 1), (2, 1), (2, 1), (3, 1)\}$. Such a method of knowing the absolute line number associated to a `\sameword` is required because a `\edtext` can overlap many lines, but `\sameword` can't get it.
 - * When reading the auxiliary file, we get the value associated to the pair composed by the current marked word and the current absolute line number. To this value, we subtract the value associated to the pair composed by the current marked word and the previous absolute line number. Let the result be n .
 - If $n > 1$, that means the current word appears more than once in its line. In this case, we call `\showwordrank` with the word as the first argument and r as the second argument. If the word is called only once, we just print it.

After theory, implementation.

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

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

```

1652 \newcommand{\get@sw@txt}[1]{%
1653   \ifxetex%
1654     \xdef\sw@txt{#1}%
1655   \else%
1656     \expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize{#1}}%
1657   \fi%
1658 }%
1659 %

```

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

```

1660 \newcommandx{\sameword}[2][1,usedefault]{%
1661   \leavevmode%
1662   \get@sw@txt{#2}%
1663 %

```

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

```

1664 \unless\ifledRcol%
1665   \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1666 %

```

Then, write its value to the numbered file.

```

1667   \protected@write\linenum@out{\string\sw@\sw@txt}{\csuse{sw@\sw@txt}
1668 }{#1}}%
1669 %

```

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

```

1669 \else%
1670   \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1671   \protected@write\linenum@outR{\string\sw@\sw@txt}{\csuse{sw@\sw@txt}
1672 }{#1}}%
1673   \fi%
1674 %

```

And print the word.

```

1674   #2%
1675 }%
1676 %

```

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

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

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

```
\if@addsw77 \newif\if@addsw%
1678 %
```

`\@sw` The command printed in the auxiliary files.

```
1679 \newcommand{\@sw}[3]{%
1680   \get@sw@txt{#1}%
1681   \unless\ifledRcol%
1682 %
```

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

```
1683   \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
1684 %
```

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

```
1685   \numdef{\prev@line}{\the\absline@num-1}%
1686   \ifcsundef{sw@\sw@txt @\prev@line @\the\section@num}{%
1687     \csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
1688   }{}%
1689 %
```

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

```
1690   \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1691 %
```

And do the same thing for the right side.

```
1692   \else%
1693     \csxdef{sw@\sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1694     \numdef{\prev@line}{\the\absline@numR-1}%
1695     \ifcsundef{sw@\sw@txt @\prev@line @\the\section@numR @R}{%
1696       \csnumgdef{sw@\sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1697     }{}%
1698     \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
1699   \fi%
1700 %
```

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

```
1701   \@tempcnta=\@edtext@level
1702   \@whilenum{\@tempcnta>0}\do{%
1703     \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1704     {%
1705       \@addswfalse%
1706       \notbool{lemmacommand@\the\@tempcnta}%
1707       {\@addswtrue}%

```

```

1708     {\IfStrEq{#3}{inlemma}%
1709      {\@addswtrue}%
1710      {%
1711       \def\do##1{%
1712         \ifnumequal{##1}{\the\@tempcnta}%
1713         {\@addswtrue\listbreak}%
1714         {}%
1715       }%
1716       \docsvlist{#3}%
1717     }%
1718   }%
1719   \if@addsw%
1720     \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1721     \ifledRcol%
1722       \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
1723     \else%
1724       \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
1725     \fi%
1726     \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
1727   \fi%
1728 }%
1729 {}%
1730 \advance\@tempcnta by -1%
1731 }%
1732 }%
1733 %

```

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

```

1734 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1735   \get@sw@txt{#2}%
1736   \unless\ifledRcol%
1737 %

```

Just a precaution.

```

1738   \ifx\sw@list@inedtext\empty%
1739     \def\the@sw{999}%
1740     \def\this@absline{-99}%
1741   \else%
1742 %

```

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

```

1743   \gl@p\sw@list@inedtext\to\@tmp%
1744   \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1745   \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1746   \fi%
1747 %

```

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

```

1748 \ifcsdef{sw@sw@txt @\this@absline @\the\section@num}{%
1749 \numdef{\prev@line}{\this@absline-1}%
1750 \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@num}-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1751 }%
1752 {\numdef{\sw@atthisline}{0}}%
1753 %

```

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

```

1754 \ifnumgreater{\sw@atthisline}{1}%
1755 {\showwordrank{#2}{\the@sw}}%
1756 {#2}%
1757 %

```

And the same for right side.

```

1758 \else%
1759 \ifx\sw@list@inedtext\empty%
1760 \def\the@sw{999}%
1761 \def\this@absline{-99}%
1762 \else%
1763 \gl@p\sw@list@inedtext\to\@tmp%
1764 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1765 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1766 \fi%
1767 \ifcsdef{sw@sw@txt @\this@absline @\the\section@numR @R}{%
1768 \numdef{\prev@line}{\this@absline-1}%
1769 \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
1770 }%
1771 {\numdef{\sw@atthisline}{0}}%
1772 \ifnumgreater{\sw@atthisline}{1}%
1773 {\showwordrank{#2}{\the@sw}}%
1774 {#2}%
1775 \fi%
1776 }%
1777 %

```

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

```

1778 \newcommand{\showwordrank}[2]{%
1779 #1\textsuperscript{#2}%
1780 }%
1781 %

```

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.

`\num@lines`

`\one@line`

`\par@line`

```
1782 \newbox\raw@text
1783 \newif\ifnumberedpar@
1784 \newcount\num@lines
1785 \newbox\one@line
1786 \newcount\par@line
1787 %
```

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

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

`\numberpstarttrue`

`\numberpstartfalse`

`\labelpstarttrue` Beware: everything that occurs between `\pstart` and `\pend` is happening within a group; definitions must be global if you want them to survive past the end of the paragraph.

`\labelpstartfalse`

`\thepstart`

```
1788
1789 \newcommand{\AtEveryPstart}[1]{%
1790   \ifstrempy{#1}%
1791     {\xdef\at@every@pstart{}}%
1792     {\gdef\at@every@pstart{\noindent#1}}%
1793 }%
1794 \xdef\at@every@pstart{}%
1795
1796 \newcounter{pstart}
1797 \renewcommand{\thepstart}{\bfseries\@arabic@c@pstart}. }
1798 \newif\ifnumberpstart
1799 \numberpstartfalse
1800 \newif\iflabelpstart
1801 \labelpstartfalse
1802 \newcommandx*\pstart[1][1]{%
1803   \normal@pars%
1804   \ifstrempy{#1}{\at@every@pstart}{\noindent#1}%
1805   \ifautopar%
```

```

1806 \autopar%
1807 \fi%
1808 \ifluatex%
1809 \edef\l@luatexttextdir@L{\the\textdir}%
1810 \fi%
1811 \@nbreaktrue%
1812 \ifnumbering \else%
1813 \led@err@PstartNotNumbered%
1814 \beginnumbering%
1815 \fi%
1816 \ifnumberedpar@%
1817 \led@err@PstartInPstart%
1818 \pend%
1819 \fi%
1820 \list@clear{\inserts@list}%
1821 \global\let\next@insert=\empty%
1822 \begingroup\normal@pars%
1823 \global\advance \l@dnumstartL\@ne
1824 \global\setbox\raw@text=\vbox\bgroup%
1825 \if@nbreak%
1826 \if@afterindent\else%
1827 \noindent%
1828 \fi%
1829 \fi%
1830 \ifautopar\else%
1831 \ifnumberpstart%
1832 \ifinstanza\else%
1833 \ifsidepstartnum\else%
1834 \thepstart%
1835 \fi%
1836 \fi%
1837 \fi%
1838 \fi%
1839 \numberedpar@true%
1840 \iflabelpstart\protected@edef\@currentlabel%
1841 {\p@pstart\thepstart}
1842 \fi%
1843 \l@dzeropenalties%
1844 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
1845 }
1846 %

```

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

```

1847 \newcommandx*\pend}[1][1]{\ifnumbering \else%
1848 \led@err@PendNotNumbered%
1849 \fi%
1850 \global\l@dskipversenumberfalse%
1851 \ifnumberedpar@ \else%

```



```

1852 \led@err@PendNoPstart%
1853 \fi%
1854 %

```

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.

```

1855 \l@dzeropenalties%
1856 \endgraf\global\num@lines=\prevgraf\egroup%
1857 \global\par@line=0%
1858 %

```

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`.

```

1859 \cnumdef{pstartline}{0}%
1860 \loop\ifvbox\raw@text%
1861   \cnumdef{pstartline}{\pstartline+\@ne}%
1862   \do@line%
1863   \ifbypstart@%
1864     \ifnumequal{pstartline}{1}{%
1865       \bgroup%
1866       \let\leavevmode\relax%
1867       \setline{1}%
1868       \egroup%
1869       \resetprevline@}{}%
1870     \fi%
1871   \repeat%
1872 %

```

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

```

1873 \flush@notes%
1874 \endgroup%
1875 \ignorespaces%
1876 %

```

Increase `pstart` counter.

```

1877 \ifnumberpstart%
1878   \pstartnumtrue%
1879 \fi%
1880 \addtocounter{pstart}{1}%
1881 %

```

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

```

1882 \normal@pars%
1883 \ifstrempy{#1}{\at@every@pend}{\noindent#1}%
1884 %

```

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.

```

1885 \@nobreakfalse%
1886 \ifautopar%
1887   \autopar%
1888 \fi%
1889 }
1890 %

```

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.

```

\AtEveryPend91
\at@every@pend92 \newcommand{\AtEveryPend}[1]{%
1893   \ifstrempy{#1}%
1894   {\xdef\at@every@pend{}}%
1895   {\gdef\at@every@pend{\noindent#1}}%
1896 }%
1897 \xdef\at@every@pend{}%
1898
1899 %

```

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

```

1900 \newcommand*\l@dzeroopenalties{%
1901   \brokenpenalty \z@ \clubpenalty \z@
1902   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
1903   \postdisplaypenalty \z@ \widowpenalty \z@}
1904
1905 %

```

`\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.

```

1906 \newif\ifautopar
1907 \autoparfalse
1908 \newcommand*{\autopar}{
1909   \ifledRcol
1910     \ifnumberingR \else
1911       \led@err@AutoparNotNumbered
1912       \beginnumberingR
1913       \fi
1914     \else
1915       \ifnumbering \else
1916         \led@err@AutoparNotNumbered
1917         \beginnumbering
1918         \fi
1919       \fi
1920       \autopartrue
1921       \everypar{\setbox0=\lastbox
1922         \endgraf \vskip-\parskip
1923         \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\fi\fi
1924         \let\par=\pend}%
1925       \ignorespaces}
1926 %

```

`\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.

```

1927 \newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}
1928
1929 %

```

`\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.

```

1930 \newif\ifaautopar@pause
1931 %

```

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`

```

1932 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
1933 \newcommand*{\do@line}{%
1934   {\vbadness=10000
1935     \splittopskip=\z@
1936     \do@linehook
1937   \l@emptyd@ta
1938     \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
1939   \unvbox\one@line \global\setbox\one@line=\lastbox
1940   \getline@num
1941   \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{%
1942     \ifnum\@lock>\@ne
1943       \inserthangingsymboltrue
1944     \else
1945       \inserthangingsymbolfalse
1946     \fi
1947     \check@pb@in@verse
1948     \ifl@dhiddenumber%
1949       \global\l@dhiddenumberfalse%
1950       \f@x@l@cks%
1951     \else%
1952       \affixline@num%
1953     \fi%
1954   %

```

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

```

1955 \xifinlist{\the\l@dnumstartsL}{\eled@sections@}%
1956   {\print@eledsection}%
1957   {\print@line}%
1958   \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{%
1959     }%
1960 %

```

VII.2.2 Process for “normal” line

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

```

1961 \def\print@line{
1962 %

```

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

```
1963 \affixpstart@num%
1964 %
```

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

```
1965 \hb@xt@ \linewidth{%
1966 %
```

User hook.

```
1967 \do@insidelinehook%
1968 %
```

Left line number

```
1969 \l@dld@ta%
1970 %
```

Prepare text to be inserted before notes.

```
1971 \if@firstlineofpage%
1972 \set@Xtxtbeforenotes%
1973 \global\@firstlineofpagefalse%
1974 \fi%
1975 %
```

Insert footnotes made of manuscripts data.

```
1976 \insert@msdata%
1977 %
```

Restore marginal and footnotes.

```
1978 \add@inserts%
1979 \add@Xgroupbyline%
1980 \affixside@note%
1981 %
```

Print left notes.

```
1982 \l@dlsn@te
1983 %
```

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

```
1984 {\ledllfill\hb@xt@ \wd\one@line{\new@line%
1985 %
```

If we use Lua_{TEX} then restore the direction.

```
1986 \ifluatex%
1987 \texdir\l@luatextextdir@L%
1988 \fi%
1989 %
```

Insert, if needed, the hanging symbol.

```

1990 \inserthangingsymbol%
1991 %

```

And so, print the line.

```

1992 \l@dunhbox@line{\one@line}}%
1993 %

```

Right line number

```

1994 \ledrlfill\l@drd@ta%
1995 %

```

Print right notes.

```

1996 \l@drsn@te
1997 }}%
1998 %

```

And reinsert penalties (for page breaking)...

```

1999 \add@penalties%
2000 }
2001 %

```

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.

```

2002 \def\print@eledsection{%
2003   \if@firstlineofpage%
2004     \set@Xtxtbeforenotes%
2005     \global\@firstlineofpagefalse%
2006   \fi%
2007   \insert@msdata%
2008   \add@inserts%
2009   \add@Xgroupbyline%
2010   \affixside@note%
2011   \numdef{\temp@}{\l@dnumstartsL-1}%
2012   \xifinlist{\temp@}{\eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
2013   \@eled@sectioningtrue%
2014   \csuse{eled@sectioning@the\l@dnumstartsL}%
2015   \@eled@sectioningfalse%
2016   \global\csundef{eled@sectioning@the\l@dnumstartsL}%
2017   \if@RTL%
2018     \hspace{-3\paperwidth}%
2019     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2020   \else%
2021     \hspace{3\paperwidth}%
2022     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%

```

```

2023 \fi%
2024 \vskip-\baselineskip%
2025 }
2026 %

```

VII.2.4 Hooks

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

\do@insidelinehook

```

2027 \newcommand*\do@linehook{}
2028 \newcommand*\do@insidelinehook{}
2029 %

```

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

\doinsidelinehook

```

2030 \newcommand*\dolinehook[1]{\gdef\do@linehook{#1}}%
2031 \newcommand*\doinsidelinehook[1]{\gdef\do@insidelinehook{#1}}%
2032 %
2033 %

```

VII.2.5 Sidenotes and marginal line number initialization

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

\l@dcsnotetext

\l@dcsnotetext@l

\l@dcsnotetext@r

```

2034 \newcommand*\l@emptyd@ta{}%
2035 \gdef\l@dld@ta{}%
2036 \gdef\l@drd@ta{}%
2037 \gdef\l@dcsnotetext@l{}%
2038 \gdef\l@dcsnotetext@r{}%
2039 \gdef\l@dcsnotetext{}%
2040 %
2041 %

```

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

\l@drsn@te

```

2042 \newcommand*\l@dlsn@te{}%
2043 \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep}}
2044 \newcommand*\l@drsn@te{}%
2045 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
2046 %
2047 %

```

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

\ledrfill

```

2048 \newcommand*{\ledllfill}{\hfil}
2049 \newcommand*{\ledrlfill}{}
2050
2051 %

```

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.

```

2052 \newcommand*{\getline@num}{%
2053   \global\advance\absline@num \@ne%
2054   \do@actions
2055   \do@ballast
2056   \ifnumberline
2057     \ifsublines@
2058       \ifnum\sub@lock<\tw@
2059         \global\advance\subline@num \@ne
2060       \fi
2061     \else
2062       \ifnum\@lock<\tw@
2063         \global\advance\line@num \@ne
2064         \global\subline@num \z@
2065       \fi
2066     \fi
2067   }
2068 }
2069 %

```

\do@ballast The real work in the macro above is done in `\do@actions`, but before we plunge into that, let's get `\do@ballast` out of the way. This macro looks to see if there is an action to be performed on the *next* line, and if it is going to be a page break action, `\do@ballast` decreases the count `\ballast@count` counter by the amount of ballast. This means, in practice, that when `\add@penalties` assigns penalties at this point, \TeX will be given extra encouragement to break the page here (see XI.2 p. 154).

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

\c@ballast

```

2070 \newcount\ballast@count
2071 \newcounter{ballast}
2072 \setcounter{ballast}{0}
2073 %

```

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.


```

2074 \newcommand*{\do@ballast}{\global\ballast@count \z@
2075 \beginingroup
2076 \advance\absline@num \@ne
2077 \ifnum\next@actionline=\absline@num
2078 \ifnum\next@action>-1001\relax
2079 \global\advance\ballast@count by -\c@ballast
2080 \fi
2081 \fi
2082 \endgroup}
2083 %

```

`\do@actions` The `\do@actions` macro looks at the list of actions to take at particular absolute line numbers, and does everything that is specified for the current line.

`\do@actions@next`

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`.

```

2084 \newcommand*{\do@actions}{%
2085 \global\let\do@actions@next=\relax
2086 \ifnum\absline@num<\next@actionline\else
2087 %

```

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.

```

2088 \ifnum\next@action>-1001
2089 \global\page@num=\next@action
2090 \global\@firstlineofpagetrue%
2091 \ifbypage@
2092 \global\line@num=\z@ \global\subline@num=\z@
2093 \resetprevline@
2094 \fi
2095 \add@msdata@firstlineofpage%
2096 %

```

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`.)

```

2097 \else
2098 \ifnum\next@action<-4999
2099 \@l@dttempcnta=-\next@action
2100 \advance\@l@dttempcnta by -5001
2101 \ifsublines@
2102 \global\subline@num=\@l@dttempcnta
2103 \else
2104 \global\line@num=\@l@dttempcnta
2105 \fi
2106 %

```

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

```

2107     \else
2108         \@l@dttempcnta=-\next@action
2109         \advance\@l@dttempcnta by -1000
2110         \do@actions@fixedcode
2111     \fi
2112 \fi
2113 %

```

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

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

```

2114     \ifx\actionlines@list\empty
2115         \gdef\next@actionline{1000000}%
2116     \else
2117         \gl@p\actionlines@list\to\next@actionline
2118         \gl@p\actions@list\to\next@action
2119         \global\let\do@actions@next=\do@actions
2120     \fi
2121 \fi
2122 %

```

Make the recursive call, if necessary.

```

2123 \do@actions@next}
2124
2125 %

```

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

```

2126 \newcommand*{\do@actions@fixedcode}{%
2127     \ifcase\@l@dttempcnta
2128     \or% % 1001 = starting sublineation
2129         \global\sublines@true
2130     \or% % 1002 = ending sublineation
2131         \global\sublines@false
2132     \or% % 1003 = starting locking number
2133         \global\@lock=\@ne
2134     \or% % 1004 = ending locking number
2135         \ifnum\@lock=\tw@
2136             \global\@lock=\thr@@
2137         \else
2138             \global\@lock=\z@
2139         \fi
2140     \or% % 1005 = starting locking subnumber
2141         \global\sub@lock=\@ne
2142     \or% % 1006 = ending locking subnumber
2143         \ifnum\sub@lock=\tw@
2144             \global\sub@lock=\thr@@
2145         \else

```

```

2146 \global\sub@lock=\z@
2147 \fi
2148 \or% % 1007 = skipping numbering
2149 \l@dskipnumbertrue
2150 \or% % 1008 = skipping numbering in stanza
2151 \l@dskipversenumbertrue%
2152 \or% % 1009 = hiding number
2153 \l@dhidenumbertrue
2154 \or% % 1010 = inserting msdata
2155 \add@msdata%
2156 \else
2157 \led@warn@BadAction
2158 \fi}
2159
2160
2161 %

```

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` \leq `\firstlinenum`, we compare the two directly instead of making these calculations.

We compute, in the scratch counter `\@l@tempcnta`, the number of the next line that should be printed with a number (*m* in the above discussion), and move the current line number into the counter `\@l@tempcntb` for comparison.

First, the case when we are within a sub-line range.

```

2162 \newcommand*{\affixline@num}{%
2163 %

```

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).

```

2164 \ifl@groupnotesL@else
2165 \ifnumberline
2166 \ifl@dskipnumber
2167 \global\l@dskipnumberfalse
2168 \else
2169 \ifsublines@

```

```

2170 \l@dttempcntb=\subline@num
2171 \ifnum\subline@num>\c@firstsublinenum
2172 \l@dttempcnta=\subline@num
2173 \advance\l@dttempcnta by-\c@firstsublinenum
2174 \divide\l@dttempcnta by\c@sublinenumincrement
2175 \multiply\l@dttempcnta by\c@sublinenumincrement
2176 \advance\l@dttempcnta by\c@firstsublinenum
2177 \else
2178 \l@dttempcnta=\c@firstsublinenum
2179 \fi
2180 %

```

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.

```

2181 \ch@cksub@l@ck
2182 %

```

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

```

2183 \else
2184 \l@dttempcntb=\line@num
2185 %

```

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

```

2186 \ifx\linenumberlist\empty
2187 \ifnum\line@num>\c@firstlinenum
2188 \l@dttempcnta=\line@num
2189 \advance\l@dttempcnta by-\c@firstlinenum
2190 \divide\l@dttempcnta by\c@linenumincrement
2191 \multiply\l@dttempcnta by\c@linenumincrement
2192 \advance\l@dttempcnta by\c@firstlinenum
2193 \else
2194 \l@dttempcnta=\c@firstlinenum
2195 \fi
2196 \else
2197 %

```

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

```

2198 \l@dttempcnta=\line@num
2199 \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2200 \edef\sc@n@list{\def\noexpand\sc@n@list
2201 ###1,\number\l@dttempcnta,###2|{\def\noexpand\rem@inder
2202 {###2}}}%
2203 \sc@n@list\expandafter\sc@n@list\rem@inder|
2204 \ifx\rem@inder\empty%
2205 \advance\l@dttempcnta\@ne
2206 \fi
2207 %

```

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

```
2208      \ch@ck@l@ck
2209      \fi
2210  %
```

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

```
2211      \ifnum\@l@tempcnta=\@l@tempcntb
2212      \ifl@dskipversenumber\else
2213  %
```

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

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

```
\l@drd@ta
2214      \if@twocolumn
2215      \if@firstcolumn
2216      \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2217      \else
2218      \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2219      \fi
2220      \else
2221      \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}}%
2222      {\@l@tempcntb=\line@margin@columns}%
2223      {\@l@tempcntb=\line@margin}%
2224      \ifnum\@l@tempcntb>\@ne
2225      \advance\@l@tempcntb \page@num
2226      \fi
2227      \ifodd\@l@tempcntb
2228      \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2229      \else
2230      \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2231      \fi
2232      \fi
2233      \fi
2234      \fi
2235  %
```

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

```

2236     \f@x@l@cks
2237     \fi
2238   \fi
2239 \fi
2240 }
2241
2242 %

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck`
`\ch@ck@l@ck` checks subline locking. If it fails, then we disable the line-number display by setting the
`\f@x@l@cks` counters to arbitrary but unequal values.

```

2243 \newcommand*{\ch@cksub@l@ck}{%
2244   \ifcase\sub@lock
2245   \or
2246     \ifnum\sublock@disp=\@ne
2247       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2248     \fi
2249   \or
2250     \ifnum\sublock@disp=\tw@ \else
2251       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2252     \fi
2253   \or
2254     \ifnum\sublock@disp=\z@
2255       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2256     \fi
2257   \fi}
2258 %

```

Similarly for line numbers.

```

2259 \newcommand*{\ch@ck@l@ck}{%
2260   \ifcase\@lock
2261   \or
2262     \ifnum\lock@disp=\@ne
2263       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2264     \fi
2265   \or
2266     \ifnum\lock@disp=\tw@ \else
2267       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2268     \fi
2269   \or
2270     \ifnum\lock@disp=\z@
2271       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2272     \fi
2273   \fi}
2274 %

```

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

```

2275 \newcommand*{\f@x@l@cks}{%
2276   \ifcase\@lock
2277   \or
2278     \global\@lock=\tw@
2279   \or \or
2280     \global\@lock=\z@
2281   \fi
2282   \ifcase\sub@lock
2283   \or
2284     \global\sub@lock=\tw@
2285   \or \or
2286     \global\sub@lock=\z@
2287   \fi}
2288
2289 %

```

X Pstart number printing in side

In side, the printing of pstart number is running like the printing of line number. There is only some differences:

- The pstarts counter is upgrade in the \pend command. Consequently, the \affixpstart@num command has not to upgrade it, unlike the \affixline@num which upgrades the lines counter.
- To print the pstart number only at the beginning of a pstart, and not in every line, a boolean test is made. The \pstartnum boolean is set to TRUE at every \pend. It is tried in the \leftpstartnum and \rightpstartnum commands. After the try, it is set to FALSE.

```

\leftpstartnum90
\rightpstartnum91 \newif\ifsidepstartnum
\ifsidepstartnum92 \newcommand*{\affixpstart@num}{%
2293   \ifsidepstartnum
2294     \if@twocolumn
2295       \if@firstcolumn
2296         \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2297       \else
2298         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2299       \fi
2300     \else
2301       \@l@dttempcntb=\line@margin
2302       \ifnum\@l@dttempcntb>\@ne
2303         \advance\@l@dttempcntb \page@num
2304       \fi
2305       \ifodd\@l@dttempcntb
2306         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2307       \else

```

```

2308 \gdef\l@dld@ta{\llap{{\leftstartnum}}}%
2309 \fi
2310 \fi
2311 \fi
2312 }
2313 %
2314 %
2315 \newif\ifpstartnum
2316 \pstartnumtrue
2317 \newcommand*{\leftpstartnum}{
2318 \ifpstartnum\thepstart
2319 \kern\linenumsep\fi
2320 \global\pstartnumfalse
2321 }
2322 \newcommand*{\rightpstartnum}{
2323 \ifpstartnum
2324 \kern\linenumsep
2325 \thepstart
2326 \fi
2327 \global\pstartnumfalse
2328 }
2329 %
2330 %

```

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.

```

2331 \list@create{\inserts@list}
2332 %

```

`\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`.

```
2333 \newcommand*{\add@inserts}{%
2334   \global\let\add@inserts@next=\relax
2335 }
```

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

```
2336 \ifx\inserts@list\empty \else
2337 %
```

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.

```
2338 \ifx\next@insert\empty
2339   \ifx\insertlines@list\empty
2340     \global\noteschanged@true
2341     \gdef\next@insert{100000}%
2342   \else
2343     \gl@p\insertlines@list\to\next@insert
2344   \fi
2345 \fi
2346 %
```

If the next insert's for this line, tack it on (and then erase the contents of the insert macro, as it could be quite large). In that case, we also set `\add@inserts@next` so that we will call ourself recursively: there might be another insert for this same line.

```
2347 \ifnum\next@insert=\absline@num
2348   \gl@p\inserts@list\to\@insert
2349   \@insert
2350   \global\let\@insert=\undefined
2351   \global\let\next@insert=\empty
2352   \global\let\add@inserts@next=\add@inserts
2353 \fi
2354 \fi
2355 %
```

Make the recursive call, if necessary.

```
2356 \add@inserts@next}
2357
2358 %
```

`\add@Xgroupbyline` If you use `\Xgroupbyline`, the insertion of the critical footnotes are not made immediately in `\add@inserts`, but the content to be inserted is stored, to be inserted in one block. This insertion in one block is made by `\add@Xgroupbyline`.

```
2359 \newcommand{\add@Xgroupbyline}{%
2360   \unless\ifnocritical%
```

```

2361 \def\do##1{%Looping on the series
2362 \def\do####1{%Looping on the ##1@forinserting command
2363 \ifcsdef{##1@forinserting@####1}{%
2364 \X@beforeinsertion{##1}%
2365 \if@ledgroup%
2366 \global\setbox\@nameuse{mp##1footins}=\vbox%
2367 \else%
2368 \insert\csname ##1footins\endcsname%
2369 \fi%
2370 {%
2371 \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
2372 {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
2373 }%
2374 \if@ledgroup%
2375 \unvbox\@nameuse{mp##1footins}%
2376 \fi%
2377 \X@atbegininsertion{##1}%
2378 \ifcsstring{series@display##1}%
2379 {%
2380 \Xledsetnormalparstuff{##1}%
2381 \rule{z@}{\splittopskip}%
2382 }%
2383 }%
2384 \csuse{##1@forinserting@####1}%
2385 \strut\par%
2386 }%
2387 \global\csundef{##1@forinserting@####1}%
2388 }%
2389 }%
2390 }%
2391 \ifcsdef{##1@forinserting}{%
2392 \dolistcslloop{##1@forinserting}%
2393 }{%
2394 \global\csundef{##1@forinserting}%
2395 }%
2396 \dolistloop{\@series}%
2397 \fi%
2398 }%
2399
2400
2401 %

```

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. 144). Finally, the penalty is checked to see that it does not go below -10000 .

```

2402 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
2403 \ifnum\num@lines>\@ne
2404   \global\advance\par@line \@ne
2405   \ifnum\par@line=\@ne
2406     \advance\@l@tempcnta \clubpenalty
2407   \fi
2408   \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
2409   \ifnum\@l@tempcntb=\num@lines
2410     \advance\@l@tempcnta \widowpenalty
2411   \fi
2412   \ifnum\par@line<\num@lines
2413     \advance\@l@tempcnta \interlinepenalty
2414   \fi
2415 \fi
2416 \ifnum\@l@tempcnta=\z@
2417   \relax
2418 \else
2419   \ifnum\@l@tempcnta>-10000
2420     \penalty\@l@tempcnta
2421   \else
2422     \penalty -10000
2423   \fi
2424 \fi}
2425
2426 %

```

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.

```

2427 \newcommand*{\flush@notes}{%
2428   \@xloop
2429   \ifx\inserts@list\empty \else
2430     \gl@p\inserts@list\to\@insert
2431     \@insert
2432     \global\let\@insert=\undefined

```

```

2433 \repeat}
2434
2435 %

```

\@xloop `\@xloop` is a variant of the PLAIN T_EX `\loop` macro, useful when it's hard to construct a positive test using the T_EX `\if` commands—as in `\flush@notes` above. One types `\@xloop ... \if ... \else ... \repeat`, and the action following `\else` is repeated as long as the `\if` test fails. (This macro will work wherever the PLAIN T_EX `\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.

```

2436 \def\@xloop#1\repeat{%
2437   \def\body{#1\expandafter\body\fi}%
2438   \body}
2439
2440 %

```

XI.4 Text before notes

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

```

2441 \newcommand{\set@Xtxtbeforenotes}{%
2442   \unless\ifnocritical@%
2443     \def\do##1{%
2444       \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2445     }%
2446     \dolistloop{\@series}%
2447   \fi%
2448 }%
2449 %

```

\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.

```

2450 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2451   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2452     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2453     \ifcvoid{Xtxtbeforenotes@#1}{%
2454       \ifcsstring{series@display#1}{paragraph}%
2455       {\noindent\cuse{Xtxtbeforenotes@#1}}%
2456       {\expandafter\insert\csname#1footins\endcsname%
2457         \bgroup%
2458         \noindent%

```

```

2459         \csuse{\csuse{series@display#1}@begin@insert}{#1}%
2460         \strut\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}%
2461         \egroup%
2462     }%
2463 }%
2464 }%
2465 {}%
2466 }%
2467 %

```

XII Critical footnotes

The footnote macros are adapted from those in PLAIN \TeX , but they differ in these respects: the outer-level commands must add other commands to a list macro rather than doing insertions immediately; there are many separate levels of the footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

XII.1 Fonts

Before getting into the details of formatting the notes, we set up some font macros. It is the notes that present the greatest challenge for our font-handling mechanism, because we need to be able to take fragments of our main text and print them in different forms: it is common to reduce the size, for example, without otherwise changing the fonts used.

`\select@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.

```

2468 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@@lemmafont#7|}
2469 \def\select@@lemmafont#1/#2/#3/#4|{%
2470     {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2471     \selectfont}
2472
2473 %

```

XII.2 Individual note options

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

```

2474 \newcommand*{\footnoteoptions@}[3]{%
2475     \def\do##1{%
2476         \ifstrequal{#1}{L}{% In Leftside
2477             \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@list%
2478             \global\advance\insert@count \@ne% Increment the left insert
counter.

```

```

2479     }%
2480     {%
2481         \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}\to\inserts@listR%
2482         \global\advance\insert@countR \@ne% Increment the right insert
counter insert.
2483     }%
2484 }%
2485 \notblank{#2}{\docsvlist{#2}}{}}% Parsing all options
2486 }
2487 %

```

XII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua⁴TeX is used.

```

2488 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
2489     \ifstrequal{#1}{L}{%
2490         \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@list%Know the dir of lemma
2491         \global\advance\insert@count \@ne%
2492         \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@list%Know the dir of lemma
2493         \global\advance\insert@count \@ne%
2494     }%
2495     {%
2496         \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@listR%Know the dir of lemma
2497         \global\advance\insert@countR \@ne%
2498         \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@listR%Know the dir of lemma
2499         \global\advance\insert@countR \@ne%
2500     }%
2501 }
2502 %

```

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

```

2503 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2504     \ifstrequal{#1}{L}{%
2505         \if@RTL%
2506             \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@list%Know the language used in the lemma
2507             \global\advance\insert@count \@ne%
2508         \else
2509             \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@list%Know the language of lemma

```

```

2510 \global\advance\insert@count \@ne%
2511 \fi%
2512 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}}}\
to\inserts@listR%Know the language of lemma
2513 \global\advance\insert@count \@ne%
2514 }%
2515 {%
2516 \if@RTL
2517 \xright@appenditem{{\csxdef{footnote@dir}{@RTLtrue}}}\to\
inserts@listR%Know the language of lemma
2518 \global\advance\insert@countR \@ne%
2519 \else
2520 \xright@appenditem{{\csxdef{footnote@dir}{@RTLfalse}}}\to\
inserts@listR%Know the language of lemma
2521 \global\advance\insert@countR \@ne%
2522 \fi
2523 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}}}\
to\inserts@listR%Know the language of lemma
2524 \global\advance\insert@countR \@ne%
2525 }%
2526 }
2527 %

```

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`.

```

2528 \newcommand*{\footplitskips}{%
2529   \interlinepenalty=\interfootnotelinepenalty
2530   \unless\ifl@dprintingpages%
2531     \floatingpenalty=\@MM%
2532   \fi%
2533   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2534   \leftskip=\z@skip \rightskip=\z@skip}
2535
2536 %

```

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

```

2537 \let\normalfootnoterule=\footnoterule
2538 %

```

XII.6 Footnotes arrangement

XII.6.1 User level macro

`\Xarrangement` `\Xarrangement[$\langle s \rangle$]{ $\langle arrangement \rangle$ }` The command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

2539 \newcommandx{\Xarrangement}[2][1,usedefault]{%
2540   \def\do##1{%
2541     \csname Xarrangement@#2\endcsname{##1}%
2542   }%
2543   \ifstrempy{#1}%
2544     {%
2545       \dolistloop{\@series}%
2546     }%
2547   {
2548     \docsvlist{#1}%
2549   }%
2550 }%
2551 %

```


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.

```
2552 \newcommand*{\Xarrangement@normal}[1]{%
2553   \csdef{series@display#1}{normal}
2554   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
2555   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
2556   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
2557   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
2558   \expandafter\let\csname #1footnoterule\endcsname=%
2559                                     \normalfootnoterule
2560   \count\csname #1footins\endcsname=1000
2561   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2562   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2563   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2564   %
```

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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ’s inserts.

```
2565   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
2566   side only
2567   %
```

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).

```
2567 \ifnoledgroup@else%
2568   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2569   \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2570   \count\csname mp#1footins\endcsname=1000
```

```

2571 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2572 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2573 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2574 \fi
2575 }
2576
2577 %

```

\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.

```

2578 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2579 \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
2580 \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
2581 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
2582 \X@beforeinsertion{#1}%
2583 \insert\csname #1footins\endcsname{%
2584 \X@atbegininsertion{#1}%
2585 \normalvfootnote@inserted{#1}{#2}%
2586 }%
2587 }%
2588 }%
2589 %

```

\normalvfootnote@inserted The \normalvfootnote@inserted macro is expanded to the content to be add to a \insert for normal critical footnote.

```

2590 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote@inserted}[2]{%
2591 \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
2592 \csuse{Xnotefontsize@#1}%
2593 \footsplitskips
2594 \ifl@dpairing\ifl@dpaging\else%
2595 \setXnoteswidthliketwocolumns@{#1}%
2596 \fi\fi%
2597 \setXnotespositionliketwocolumns@{#1}%
2598 \spaceskip=\z@skip \xspaceskip=\z@skip%
2599 \csname #1footfmt\endcsname #2{#1}%
2600 }%
2601 %

```

```

\X@beforeinsertion02 \newcommand{\X@beforeinsertion}[1]{%
2603 \if@ledgroup\else%
2604 \insert@Xtxtbeforenotes{#1}%

```

```

2605 \fi%
2606 \csuse{Xbeforeinserting@#1}%
2607 }%
2608 %

```

```

\X@atbegininsertion 2609 \newcommand{\X@atbegininsertion}[1]{%
2610 \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2611 }%
2612 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote 2613 \notbool{parapparatus@}\newcommand*{\newcommand}\mpnormalvfootnote}[2]{%
2614 \iftoggle{Xgroupbyline@#1}{%
2615 \prepare@Xgroupbyline{#1}{#2}\mpnormalvfootnote@inserted}%
2616 }%
2617 {%
2618 \global\setbox\@nameuse{mp#1footins}%
2619 \vbox{%
2620 \unvbox\@nameuse{mp#1footins}%
2621 \mpnormalvfootnote@inserted{#1}{#2}%
2622 }%
2623 }%
2624 }%
2625 %
2626 %

```

```

\normalvfootnote@inserted 2627 \newcommand{\mpnormalvfootnote@inserted}[2]{%
2628 \noindent\csuse{Xbhooknote@#1}%
2629 \csuse{Xnotefontsize@#1}%
2630 \hspace\columnwidth%
2631 \@parboxrestore%
2632 \color@begingroup%
2633 \csname #1footfmt\endcsname #2{#1}\color@endgroup%
2634 }%
2635 %

```

`\normalfootfmt` `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 95), 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.

```

2636
2637
2638 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
2639 \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}}%
2640 \hangindent=\csuse{Xhangindent@#4}%
2641 \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2642 \nottoggle{Xgroupbyline@#4}{\rule{z@}{\splittopskip}}}%
2643 {\printlinefootnote{#1}{#4}}%
2644 \print@lemma{#1}{#2}{#4}%
2645 \csuse{Xwrapcontent@#4}{#3}%
2646 \nottoggle{Xgroupbyline@#4}{\strut\par}}}%
2647 }%
2648 %

```

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

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

But if the skip \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.

```

2649 \newcommand*{\normalfootstart}[1]{%
2650 %

```

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

```

2651 \ifdimequal{0pt}{\Xprenotes@}{}%
2652 {%
2653 \iftoggle{Xprenotes@}{%
2654 \togglefalse{Xprenotes@}%
2655 \skip\csname #1footins\endcsname=%
2656 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
2657 }%
2658 }%
2659 }%
2660 \vskip\skip\csname #1footins\endcsname%
2661 %

```

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. 212 for the general description of the problem.

```

2662 \leftskip0pt \rightskip0pt
2663 \ifl@dpairing\else%
2664   \hsize=\old@hsize%
2665 \fi%
2666 \setXnoteswidthliketwocolumns@{#1}%
2667 \setXnotespositionliketwocolumns@{#1}%
2668 %

```

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

```

2669 \print@Xfootnoterule{#1}%
2670 }%
2671 %

```

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

```

2672 \newcommand*{\normalfootgroup}[1]{%
2673   \csuse{Xbhookgroup@#1}%
2674   \unvbox\csname #1footins\endcsname%
2675   \hsize=\old@hsize%
2676 }%
2677
2678 %

```

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

```

2679 \unless\ifnoledgroup@
2680 \newcommand*{\mpnormalfootgroup}[1]{%
2681   \vskip\skip\@nameuse{mp#1footins}
2682   \ifl@dpairing\ifparledgroup%
2683     \leavevmode\marks\parledgroup@{begin}%
2684     \marks\parledgroup@series{#1}%
2685     \marks\parledgroup@type{Xfootnote}%
2686   \fi\fi\normalcolor%
2687   \ifparledgroup%
2688     \ifl@dpairing%
2689     \else%
2690       \setXnoteswidthliketwocolumns@{#1}%
2691       \setXnotespositionliketwocolumns@{#1}%
2692       \print@Xfootnoterule{#1}%%
2693     \fi%
2694   \else%
2695     \setXnoteswidthliketwocolumns@{#1}%
2696     \setXnotespositionliketwocolumns@{#1}%
2697     \print@Xfootnoterule{#1}%%

```

```

2698 \fi%
2699 \setlength{\parindent}{Opt}
2700 \csuse{Xbhookgroup@#1}%
2701 \unvbox\csname mp#1footins\endcsname}}
2702 \fi
2703 %

```

XII.6.3 Paragraphed footnotes

The paragraphed-footnote option reformats all the footnotes of one series for a page into a single paragraph; this is especially appropriate when the notes are numerous and brief. The code is based on *The TeXbook*, pp. 398–400, with alterations for our environment. This algorithm uses a considerable amount of save-stack space: a T_EX of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

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

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

```

2704 \newcommand*{\Xarrangement@paragraph}[1]{%
2705   \csgdef{series@display#1}{paragraph}
2706   \expandafter\let\csname #1footstart\endcsname=\parafootstart
2707   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
2708   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
2709   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
2710   \count\csname #1footins\endcsname=1000
2711   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2712   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2713   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2714   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2715   \para@footsetup{#1}
2716 %

```

And the extra setup for minipages.

```

2717 \ifnoledgroup@else
2718   \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
2719   \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
2720   \count\csname mp#1footins\endcsname=1000
2721   \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2722   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2723   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2724 \fi
2725 }
2726 %

```

\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.

```
2727 \providecommand{\footfudgefiddle}{64}
2728 %
```

\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`.

```
2729 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}
2730 \setXnoteswidthliketwocolumns@{#1}%
2731 \ifcempty{Xwidth@#1}%
2732 {}%
2733 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
2734 \dimen0=\baselineskip
2735 \multiply\dimen0 by 1024
2736 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
2737 \csxdef{#1footfudgefactor}{%
2738 \expandafter\strip@pt\dimen0 }}
2739
2740 %
```

`\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.

```
2741 \newcommand*{\parafootstart}[1]{%
2742 \rightskip=0pt \leftskip=0pt%
2743 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2744 \ifdimequal{0pt}{\Xprenotes@}{}%
2745 {%
2746 \iftoggle{Xprenotes@}{%
2747 \togglefalse{Xprenotes@}%
2748 \skip\csname #1footins\endcsname=%
2749 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
2750 }%
```

```

2751     {}%
2752   }%
2753   \vskip\skip\csname #1footins\endcsname%
2754   \setXnoteswidthliketwocolumns@{#1}%
2755   \setXnotespositionliketwocolumns@{#1}%
2756   \printXfootnoterule{#1}%
2757   \let\bidirTL@everypar\@empty%
2758   \noindent\leavevmode}
2759 %

```

\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.

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

²⁸Michael Downes, 'Line Breaking in \unhboxed Text', *TUGboat* 11 (1990), pp. 605–612.

²⁹See *The TeXbook*, p. 455 (editions after January 1990).

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.

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. 164 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.

```

2760 \newcommand*{\paravfootnote}[2]{%
2761   \csuse{Xbeforeinserting@#1}%
2762   \insert\csname #1footins\endcsname
2763   \bgroup
2764     \csuse{Xnotefontsize@#1}
2765     \footsplitskips
2766     \setbox0=\vbox{\hsize=\maxdimen%
2767       \let\bidir@RTL@everypar\@empty%
2768       \insert@Xtxtbeforenotes{#1}%
2769       \noindent\csuse{Xbhooknote@#1}%
2770       \csname #1footfmt\endcsname #2{#1}}%
2771     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2772     \dp0=0pt
2773     \ht0=\csname #1footfudgefactor\endcsname\wd0
2774   }%

```

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

```

2775   \if@RTL\noindent \leavevmode\fi\box0%
2776   \penalty0
2777   \egroup}
2778
2779   %

```

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.

```

2780 \newcommand*\mpparavfootnote}[2]{%
2781   \global\setbox\@nameuse{mp#1footins}\vbox{%
2782     \unvbox\@nameuse{mp#1footins}%
2783     \csuse{Xnotefontsize@#1}
2784     \footsplitskips
2785     \setbox0=\vbox{\hsize=\maxdimen%
2786       \let\bidirTL@everypar\@empty%
2787       \insert@Xtxtbeforenotes{#1}%
2788       \noindent\color@begingroup%
2789       \csuse{Xhooknote@#1}%
2790       \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
2791     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2792     \dp0=\z@
2793     \ht0=\csname #1footfudgefactor\endcsname\wd0
2794     \box0
2795     \penalty0
2796   }}
2797
2798 %

```

\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`.

```

2799 \newcommand*\Xunvxh}[2]{%
2800   \setbox0=\vbox{\unvbox#1%
2801     \global\setbox1=\lastbox}%
2802   \unhbox1
2803   \unskip           % remove \rightskip,
2804   \unskip           % remove \parfillskip,
2805   \unpenalty        % remove \penalty of 10000,
2806   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
2807
2808 %

```

\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).

```

2809 \newcommand*\parafootfmt}[4]{%
2810   \Xinsertparafootsep{#4}%
2811   \ledsetnormalparstuff@common%
2812   \printlinefootnote{#1}{#4}%
2813   \print@lemma{#1}{#2}{#4}%

```

```

2814 \csuse{Xwrapcontent@#4}{#3}%
2815 \penalty-10 }
2816 %

```

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.

```

2817 \newcommand*{\parafootgroup}[1]{%
2818   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2819   \unvbox\curname #1footins\endcurname
2820   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2821   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2822   \makehboxofhboxes
2823   \setbox0=\hbox{\unhbox0 \removehboxes}%
2824   \csuse{Xbhookgroup@#1}%
2825   \csuse{Xnotefontsize@#1}%
2826   \unhbox0\par%
2827   \global\hsize=\old@hsize%
2828   }%
2829
2830 %

```

`\mpparafootgroup` The minipage version.

```

2831 \newcommand*{\mpparafootgroup}[1]{%
2832   \setXnoteswidthliketwocolumns@{#1}%
2833   \vskip\skip\@nameuse{mp#1footins}
2834   \ifl@dpairing\ifparledgroup%
2835     \leavevmode\marks\parledgroup@{begin}%
2836     \marks\parledgroup@series{#1}%
2837     \marks\parledgroup@type{Xfootnote}%
2838   \fi\fi\normalcolor
2839   \ifparledgroup%
2840     \ifl@dpairing%
2841     \else%
2842       \setXnoteswidthliketwocolumns@{#1}%
2843       \setXnotespositionliketwocolumns@{#1}%
2844       \print@Xfootnoterule{#1}%
2845     \fi%
2846   \else%
2847     \setXnoteswidthliketwocolumns@{#1}%
2848     \setXnotespositionliketwocolumns@{#1}%

```

```

2849 \print@Xfootnoterule{#1}%
2850 \fi%
2851 \unvbox\csname mp#1footins\endcsname
2852 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2853 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2854 \makeboxofhboxes
2855 \setbox0=\hbox{\unhbox0 \removehboxes}%
2856 \csuse{Xbhookgroup@#1}%
2857 \csuse{Xnotefontsize@#1}%
2858 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2859 \unhbox0\par}}
2860
2861 %

```

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

```

\makeboxofhboxes 2862 \newcommand*\makeboxofhboxes{\setbox0=\hbox{}}%
\removehboxes 2863 \loop
2864 \unpenalty
2865 \setbox2=\lastbox
2866 \ifhbox2
2867 \setbox0=\hbox{\box2\unhbox0}%
2868 \repeat}
2869
2870 \newcommand*\removehboxes{\setbox0=\lastbox
2871 \ifhbox0{\removehboxes}\unhbox0 \fi}
2872
2873 %

```

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

```

\prevpage@num 2874 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep 2875 \ifledRcol{%
2876 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
2877 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2878 {\ifnumequal{\csuse{prevline#1}}{\line@numR}%
2879 {\ifcempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
2880 {\csuse{Xparafootsep@#1}}%
2881 }%
2882 {\csuse{Xparafootsep@#1}}%
2883 }%
2884 }%
2885 \global\csname #1prevpage@numR\endcsname=\page@numR%
2886 \else%
2887 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
2888 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.

```

```

2889         {\ifnumequal{\csuse{prevline#1}}{\line@num}%
2890          {\ifcempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}}%
2891         {\csuse{Xparafootsep@#1}}}%
2892     }%
2893     {\csuse{Xparafootsep@#1}}}%
2894 }%
2895 {}%
2896 \global\csname #1prevpage@num\endcsname=\page@num%
2897 \fi%
2898 }
2899 %

```

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 a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\Xrigidbalance` `\dosplits` `\splitoff` `\@h` `\@k` `\rigidbalance` macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they do not depend on white space. Note also the extra unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it goes into the alignment.

The \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`.

```

2900 \newcount\@k \newdimen\@h
2901 \newcommand*{\Xrigidbalance}[3]{%
2902     \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
2903     \rigidbalance{#1}{#2}{#3}%
2904 }%
2905
2906 \newcommand*{\rigidbalanceX}[3]{%
2907     \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
2908     \rigidbalance{#1}{#2}{#3}%
2909 }%
2910
2911 \newcommand*{\rigidbalance}[3]{%
2912     \setbox0=\box#1 \@k=#2 \@h=#3%
2913     \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
2914     \valign{##\vfil\cr\dosplits}}}%
2915
2916 \newcommand*{\dosplits}{\ifnum\@k>0 \noalign{\hfil}\splitoff
2917     \global\advance\@k-1\cr\dosplits\fi}
2918
2919 \newcommand*{\splitoff}{\dimen0=\ht0
2920     \divide\dimen0 by\@k \advance\dimen0 by\@h

```

```

2921 \setbox2 \vsplit0 to \dimen0
2922 \unvbox2 }
2923
2924 %

```

Three columns

```

\Xarrangement@threecol 2925 \newcommand*{\Xarrangement@threecol}[1]{%
2926 \csgdef{series@display#1}{threecol}
2927 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
2928 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
2929 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
2930 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2931 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2932 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2933 \threecolfootsetup{#1}
2934 %

```

The additional setup for minipages.

```

2935 \ifnoledgroup@else
2936 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2937 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
2938 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2939 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2940 \mpthreecolfootsetup{#1}
2941 \fi
2942 }
2943
2944 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XII.6.2 p. 164 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.

```

2945 \newcommand*{\threecolfootsetup}[1]{%

```

```

2946 \count\csname #1footins\endcsname 333
2947 \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
2948 \multiply\dimen\csname #1footins\endcsname \thr@@}
2949 %

```

\mpthreecolfootsetup The setup for minipages.

```

2950 \newcommand*\mpthreecolfootsetup}[1]{%
2951 \count\csname mp#1footins\endcsname 333
2952 \multiply\dimen\csname mp#1footins\endcsname \thr@@}
2953 %
2954 %

```

\threecolvfootnote \threecolvfootnote This is the \vfootnote command for three-column notes. However, most of the code is deported on \threecolvfootnote@inserted. The call to \Xnotefontsize@<s> ensures that the \splittopskip and \splitmaxdepth take their values from the right \strutbox: the one used in a footnotes. Note especially the importance of temporarily reducing the \hsize to 0.3 of its normal value. This determines the widths of the individual columns. So if the normal \hsize is (say) 10 cm, then each column will be $0.3 \times 10 = 3$ cm wide, leaving a gap of 1 cm spread equally between columns (i.e., .5 cm between each).

The arguments are #1 the note series letter and #1 the full text of the note (including numbers, lemma and text).

```

2955 \notbool{parapparatus@}\newcommand*\newcommand*\threecolvfootnote}[2]{%
2956 \iftoggle{Xgroupbyline@#1}{%
2957 \prepare@Xgroupbyline{#1}{#2}\threecolvfootnote@inserted}%
2958 }%
2959 {%
2960 \X@beforeinsertion{#1}%
2961 \insert\csname #1footins\endcsname{%
2962 \threecolvfootnote@inserted{#1}{#2}%
2963 }%
2964 }%
2965 }%
2966 %

```

```

\threecolvfootnote@inserted 2967 \notbool{parapparatus@}\newcommand*\newcommand*\threecolvfootnote@inserted}[2]{%
2968 \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2969 \noindent\csuse{Xhooknote@#1}%
2970 \csuse{Xnotefontsize@#1}%
2971 \footplitskips%
2972 \csname #1footfmt\endcsname #2{#1}%
2973 }%
2974 %

```

`\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.

```

2975 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
2976   \threecol@begin@insert{#4}%
2977   \hspace{\parindent}%
2978   \printlinefootnote{#1}{#4}%
2979   \print@lemma{#1}{#2}{#4}%
2980   \csuse{Xwrapcontent@#4}{#3}%
2981   \nottoggle{Xgroupbyline@#4}%
2982     {\strut\par\allowbreak}%
2983   }%
2984 }%
2985 %

```

`\threecol@begin@insert` The `\threecol@begin@insert` contains code used at the beginning of any `\insert` for critical footnotes in three columns. It is used both by `\threecolfootfmt` and by `\insert@Xtxtbeforenotes`.

```

2986 \newcommand{\threecol@begin@insert}[1]{%
2987   \normal@pars%
2988   \nottoggle{Xgroupbyline@#1}%
2989     {\hsize \csuse{Xhsizethreecol@#1}}%
2990   }%
2991   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2992   \tolerance=5000%
2993   \hangindent=\csuse{Xhangindent@#1}%
2994   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
2995   \@tempdima=\parindent%
2996   \csuse{Xcolalign@#1}%
2997   \parindent=\@tempdima%
2998   \strut%
2999 }%
3000 %

```

`\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.

```

3001 \newcommand*{\threecolfootgroup}[1]{%
3002   \csuse{Xbhookgroup@#1}\par%
3003   \splittopskip=\ht\strutbox

```



```

3004 \expandafter
3005 \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
3006 %

```

\mpthreecolfootgroup The setup for minipages.

```

3007 \newcommand*{\mpthreecolfootgroup}[1]{\{
3008 \vskip\skip\@nameuse{mp#1footins}
3009 \ifl@dpairing\ifparledgroup%
3010 \leavevmode\marks\parledgroup@{begin}%
3011 \marks\parledgroup@series{#1}%
3012 \marks\parledgroup@type{Xfootnote}%
3013 \fi\fi\normalcolor
3014 \ifparledgroup%
3015 \ifl@dpairing%
3016 \else%
3017 \setXnoteswidthliketwocolumns@{#1}%
3018 \setXnotespositionliketwocolumns@{#1}%
3019 \print@Xfootnoterule{#1}%
3020 \fi%
3021 \else%
3022 \setXnoteswidthliketwocolumns@{#1}%
3023 \setXnotespositionliketwocolumns@{#1}%
3024 \print@Xfootnoterule{#1}%
3025 \fi%
3026 \csuse{Xhookgroup@#1}\par%
3027 \splittopskip=\ht\strutbox
3028 \expandafter
3029 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3030 %
3031 %

```

Two columns

```

\Xarrangement@twocol 3032 \newcommand*{\Xarrangement@twocol}[1]{%
3033 \csgdef{series@display#1}{twocol}
3034 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3035 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3036 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3037 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3038 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3039 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3040 \twocolfootsetup{#1}
3041 %

```

The additional setup for minipages.

```

3042 \ifnoledgroup\else
3043 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3044 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup

```

```

3045 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3046 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3047 \mptwocolfootsetup{#1}
3048 \fi
3049 }
3050
3051 %

```

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

```

\begin{macrocode}
\newcommand*{\twocolfootnote}[2]{%
  \count\csname #1footins\endcsname 500
  \csxdef{default@#1footins}{500}%
  \multiply\dimen\csname #1footins\endcsname \tw@}
\end{macrocode}

\begin{macrocode}
\newcommand*{\twocolfootnote@inserted}[2]{%
  \iftoggle{Xgroupbyline@#1}{%
    \prepare@Xgroupbyline{#1}{#2}{\twocolfootnote@inserted}%
  }{%
    \X@beforeinsertion{#1}%
    \insert\csname #1footins\endcsname{%
      \twocolfootnote@inserted{#1}{#2}%
    }%
  }%
}
\end{macrocode}

\begin{macrocode}
\newcommand*{\twocolfootnote@inserted}[2]{%
  \hsz=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
  \noindent\csuse{Xhooknote@#1}%
  \csuse{Xnotefontsize@#1}%
  \footssplitsskip%
  \csname #1footfmt\endcsname #2{#1}%
}
\end{macrocode}

\begin{macrocode}
\newcommand*{\twocolfootfmt}[4]{% 4th
  arg is optional, for backward compatibility
  \twocol@begin@insert{#4}%
  \hspace{\parindent}%
  \printlinefootnote{#1}{#4}%
  \print@lemma{#1}{#2}{#4}%
  \csuse{Xwrapcontent@#4}{#3}%
  \nottoggle{Xgroupbyline@#4}{%
    {\strut\par\allowbreak}%
  }%
}
\end{macrocode}

```

```

3086 \newcommand{\twocol@begin@insert}[1]{%
3087   \normal@pars%
3088   \hsize \csuse{Xhsizetwocol@#1}%
3089   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3090   \tolerance=5000%
3091   \hangindent=\csuse{Xhangindent@#1}%
3092   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3093   \@tempdima=\parindent%
3094   \csuse{Xcolalign@#1}%
3095   \parindent=\@tempdima%
3096   \strut%
3097 }%
3098
3099 \newcommand*{\twocolfootgroup}[1]{%
3100   \csuse{Xbhookgroup@#1}\par%
3101   \splittopskip=\ht\strutbox
3102   \expandafter
3103   \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
3104
3105 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3106 \newcommand*{\mptwocolfootsetup}[1]{%
3107   \count\csname mp#1footins\endcsname 500
3108   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3109 %
3110
3111 \newcommand*{\mptwocolfootgroup}[1]{%
3112   \vskip\skip\@nameuse{mp#1footins}
3113   \ifl@dpairing\ifparledgroup%
3114     \leavevmode\marks\parledgroup@{begin}%
3115     \marks\parledgroup@series{#1}%
3116     \marks\parledgroup@type{Xfootnote}%
3117   \fi\fi\normalcolor
3118   \ifparledgroup%
3119     \ifl@dpairing%
3120       \else%
3121         \setXnoteswidthliketwocolumns@{#1}%
3122         \setXnotespositionliketwocolumns@{#1}%
3123         \print@Xfootnoterule{#1}%
3124       \fi%
3125     \else%
3126       \setXnoteswidthliketwocolumns@{#1}%
3127       \setXnotespositionliketwocolumns@{#1}%
3128       \print@Xfootnoterule{#1}%
3129     \fi%
3130     \csuse{Xbhookgroup@#1}\par%
3131     \splittopskip=\ht\strutbox
3132     \expandafter

```

```

3132 \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3133
3134 %

```

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.

Note that these commands are not directly called by `reledmac`, but are enclosed as default value of specific hooks. Consequently, people should not redefine them, but use instead the `\Xlinrangeseparator`, `\Xendlinrangeseparator`, `\Xsublinesep`, `\Xendsublinesep` and `\Xlemmaseparator` macros.

With `polyglossia`, each critical note has a `\footnote@lang` which shows the language of the lemma, and which can be used to switch the bracket from right to left.

```

3135 \def\endashchar{\textnormal{--}}
3136
3137 \newcommand*\fullstop{\textnormal{.}}
3138 \def\Xsublinesep@side{\fullstop}
3139
3140 \newcommand*\rbracket{\textnormal{%
3141   \csuse{text\csuse{footnote@lang}}{%
3142     \ifluatex%
3143       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]{\thinspace
3144         \else%
3145         \thinspace}%
3146         \fi}%
3147   }}%
3148 }
3149
3150 %

```

XII.7.2 Pstart number in footnote

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

```

3151 \newcommand{\printpstart}[0]{%
3152   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3153     \ifledRcol%
3154       \thepstartR%
3155     \else%
3156       \thepstartL%
3157     \fi%
3158   }{%
3159     \thepstart%
3160   }%
3161 }
3162 %

```

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).

```

3163 %
3164 \newcommand{\print@lemma}[3]{%
3165   \bgroup%
3166   \nottoggle{Xlemmadisablefontselection@#3}%
3167   {\select@lemmafont#1|}%
3168   }%
3169   \bgroup%
3170   \csuse{Xlemmafont@#3}%Deprecated
3171   \csuse{Xwraplemma@#3}{#2}%
3172   \egroup%
3173   \egroup%
3174   \iftoggle{nosep@}{%
3175     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3176     \relax%
3177   }%
3178   {\ifcsemt{Xlemmaseparator@#3}%
3179     {%
3180       \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3181       \relax%
3182     }%
3183     {%
3184       \nobreak%
3185       \hskip\csuse{Xbeforelemmaseparator@#3}%
3186       \csuse{Xlemmaseparator@#3}%
3187       \hskip\csuse{Xafterlemmaseparator@#3}%
3188       \relax%
3189     }%

```

```

3190 }%
3191 }%
3192 %

```

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`.

```

3193 \newcommand{\printlinefootnote}[2]{%
3194   \l@dp@rsefootspec#1|%
3195   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3196     \edef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3197       \l@dparsedendline - \l@dparsedendsub}%
3198     {%
3199       \edef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3200     }%
3201     \iftoggle{nonum@}{%Try if the line number must printed for this specific
3202       not (by default, yes)
3203       \hspace{\csuse{Xinplaceofnumber@#2}}%
3204     }%
3205     {%
3206       \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
3207       default, yes)
3208       {%
3209         \hspace{\csuse{Xinplaceofnumber@#2}}%
3210       }%
3211       {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
3212         line number must be printed only in the first time.
3213         {%
3214           \ifcsdef{prevline#2}%
3215           {%Be sure the \prevline exists.
3216           \ifcsequal{prevline#2}{\lineinfo@}%Try it
3217           {%
3218             \ifcsequal{Xsymlinenum@#2}%
3219             {%
3220               \hspace{\csuse{Xinplaceofnumber@#2}}%
3221             }%
3222             {\printsymlinefootnotearea{#2}}%
3223           }%
3224           {%
3225             \printlinefootnotearea{#1}{#2}%
3226           }%

```

```

3227         {%
3228         \printlinefootnotearea{#1}{#2}%
3229         }%
3230     }%
3231     {%
3232     \printlinefootnotearea{#1}{#2}%
3233     }%
3234     \csxdef{prevline#2}{\lineinfo@}%
3235     }%
3236 }%
3237 }%
3238 }%
3239 }
3240 %

```

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

```

3241 \newcommand{\printsymlinefootnotearea}[1]{%
3242 \hspace{\csuse{Xbeforesymlinenum@#1}}%
3243 \csuse{Xnotenumfont@#1}%
3244 \ifdimequal{\csuse{Xboxsymlinenum@#1}}{\z@}%
3245     {\csuse{Xsymlinenum@#1}}%
3246     {\hbox to \csuse{Xboxsymlinenum@#1}%
3247       {\csuse{Xsymlinenum@#1}\hfill}%
3248     }%
3249 \hspace{\csuse{Xaftersymlinenum@#1}}%
3250 }%
3251 %

```

\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.)

```

3252 \newcommand{\printlinefootnotearea}[2]{%
3253 \printXbeforenumber{#2}%
3254 \csuse{Xnotenumfont@#2}%
3255 \boxfootnotenumbers{#1}{#2}%
3256 \printXafternumber{#2}%
3257 }%
3258 %

```

\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.

```

3259 \newcommand{\boxfootnotenumbers}[2]{%
3260 \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
3261     \printlinefootnotenumbers{#1}{#2}%

```

```

3262 }%
3263 {%
3264   \hbox to \csuse{Xboxlinenum@#2}%
3265   {%
3266     \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3267     \printlinefootnotenumbers{#1}{#2}%
3268     \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3269   }%
3270 }%
3271 }%
3272 %

```

\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.

```

3273 \newcommand{\printlinefootnotenumbers}[2]{%
3274   \xdef\@currentseries{#2}%
3275   \ifboolexpr{%
3276     (togl{Xpstart@#2} and bool{numberpstart})%
3277     or togl{Xpstarteverytime@#2}}%
3278   {\printpstart}{}%
3279   \iftoggle{Xstanza@#2}{%
3280     \ifnumberstanza%
3281       \printstanza%
3282       \csuse{Xstanzaseparator@#2}%
3283     \fi%
3284   }{%
3285     \iftoggle{Xonlypstart@#2}{%
3286       \csuse{Xtxtbeforenumber@#2}%
3287       \printlines#1|\ifledRcol@\@Rlineflag\fi}%
3288   }%
3289   %

```

\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.)

```

3290 \newcommand{\printXbeforenumber}[1]{%
3291   \hspace{\csuse{Xbeforenumber@#1}}%
3292 }%
3293 %

```

\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

```

3294 \newcommand{\printXafternumber}[1]{%
3295   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3296   \hspace{\csuse{Xafternumber@#1}}%
3297 }%
3298 %

```


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. 95: the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma.

edmac’ creator have defined six boolean in order to know which component of line number description we have to print:

- `\ifl@d@pnum` for page numbers;
- `\ifl@d@ssub` for starting sub-line;
- `\ifl@d@elin` for ending line;
- `\ifl@d@esl` for ending sub-line; and
- `\ifl@d@dash` for the dash between the starting and ending groups.

There is no boolean for the line number because it is always printed.

Maïeul Rouquette has added `\ifl@d@Xtwolines` and `\ifl@d@Xmorethantwolines` to print a symbol which stands for “and subsequent” when there are two, three or more lines.

```

\ifl@d@pnum99 \newif\ifl@d@pnum
\ifl@d@ssub00 \newif\ifl@d@ssub
\ifl@d@elin01 \newif\ifl@d@elin
\ifl@d@esl02 \newif\ifl@d@esl
\ifl@d@dash03 \newif\ifl@d@dash
\ifl@d@Xtwolines04 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines05 \newif\ifl@d@Xmorethantwolines%
06 %

\l@dp@rsefootsspec \l@dp@rsefootsspec{<spec>}{<lemma>}{<text>} parses a footnote specification. <lemma>
\l@dp@rsefootsspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@dp@rsefootsspec ber and lemma font specifier in \l@d@nums style format. The real work is done by
\l@dp@rsefootsspec \l@dp@rsefootsspec which defines macros holding the numeric values. In many cases,
\l@dp@rsefootsspec this last command is called directly. Just a reminder of the arguments:
\l@dp@rsefootsspec \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dp@rsefootsspec \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@dp@rsefootsspec \l@dp@rsefootsspec*{\l@dp@rsefootsspec}[3]{\l@dp@rsefootsspec#1|}
3307
3308 \def\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
3309 \gdef\l@dp@rsefootsspec#1|{%
3310 \gdef\l@dp@rsefootsspec#2|{%
3311 \gdef\l@dp@rsefootsspec#3|{%
3312 \gdef\l@dp@rsefootsspec#4|{%
3313 \gdef\l@dp@rsefootsspec#5|{%
3314 \gdef\l@dp@rsefootsspec#6|{%
3315 }
3316 %

```

Initialise the several number value macros.

```

3317 \def\l@dparsedstartpage{0}%
3318 \def\l@dparsedstartline{0}%
3319 \def\l@dparsedstartsub{0}%
3320 \def\l@dparsedendpage{0}%
3321 \def\l@dparsedendline{0}%
3322 \def\l@dparsedendsub{0}%
3323
3324 %

```

\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`.

```

3325 \newcommand*{\setprintlines}[6]{%
3326   \l@d@pnumfalse \l@d@dashfalse
3327 %

```

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

```

3328   \ifbypage@
3329     \ifnum#4=#1 \else
3330       \l@d@pnumtrue
3331       \l@d@dashtrue
3332     \fi
3333 \fi
3334 %

```

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.

```

3335   \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
3336   \ifnum#2=#5 \else
3337     \l@d@elintrue
3338     \l@d@dashtrue
3339   \fi
3340 %

```

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

```

3341   \l@d@ssubfalse
3342   \ifnum#3=0 \else
3343     \l@d@ssubtrue
3344   \fi
3345 %

```

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.

```

3346   \l@d@eslfalse
3347   \ifnum#6=0 \else
3348     \ifnum#6=#3

```

```

3349     \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3350     \else
3351         \l@d@esltrue
3352         \l@d@dashtrue
3353     \fi
3354 \fi%
3355 %

```

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

```

3356 \ifl@d@dash%
3357 \ifboolexpr{togl{fulllines@} or test{\ifcsempy{Xtwolines@
@currentseries}}}%
3358 {}%
3359 {%
3360 \setistwofollowinglines{#1}{#2}{#4}{#5}%
3361 \ifboolexpr{%
3362     (%
3363         togl {Xtwolinesbutnotmore@ \@currentseries}%
3364         and not%
3365         (%
3366             bool {istwofollowinglines@}%
3367         )%
3368     )%
3369     or%
3370     (%
3371         (not test{\ifnumequal{#1}{#4}})%
3372         and togl{Xtwolinesonlyinsamepage@ \@currentseries}%
3373     )%
3374 }%
3375 {}%
3376 {%
3377     \l@d@dashfalse%
3378     \l@d@Xtwolinestrue%
3379     \l@d@elinfalse%
3380     \l@d@eslfalse%
3381     \ifcsempy{Xmorethantwolines@ \@currentseries}%
3382     {%
3383         {\ifistwofollowinglines@ \else%
3384             \l@d@Xmorethantwolinestrue%
3385         \fi%
3386     }%
3387 }%
3388 }%
3389 \fi%
3390 %

```

End of `\setprintlines`.

```

3391 }%
3392 %

```

`\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.

```

3393 \newif\ifistwofollowinglines%
3394 \newcommand{\setistwofollowinglines}[4]{%
3395   \ifcsdef{lastlinenumberon#1}%
3396     {\numdef{\tmp}{\csuse{lastlinenumberon#1}}}%
3397     {\numdef{\tmp}{0}}%
3398   \istwofollowinglines@false%
3399   \ifnumequal{#4-#2}{1}%
3400     {\istwofollowinglines@true}%
3401     {\ifbypage%
3402       \ifnumequal{#3-#1}{1}%
3403       {%
3404         \ifnumequal{#2}{\tmp}%
3405         {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
3406         {}%
3407       }%
3408       {}%
3409     }%
3410   }%
3411 }%
3412 %

```

`\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

```

3413 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3414   \begingroup%
3415   %

```

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

```

3416 \ifluatex%
3417   \edef\@tmp{\the\textdir}%
3418   \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
    spurious space (bug #397)
3419   \fi%
3420 %

```

Decide which part of line number components we will print.

```

3421 \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3422 %

```

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.

```

3423 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3424   {\bgroup}%
3425   {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
    hfill}%
3426 \ifl@d@pnum%
3427   \wrap@edcrossref{\@this@crossref@start}{#1}%
3428   \csuse{Xpagelinesep@\@currentseries}%
3429   \fi%
3430 \wrap@edcrossref{\@this@crossref@start}{%
3431   \linenumrep{#2}%
3432   \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3433   }%
3434 \ifl@d@ssub%
3435   \csuse{Xsublinesep@\@currentseries}%
3436   \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3437   \fi
3438 \egroup%
3439 %

```

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

```

3440 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3441   {\bgroup}%
3442   {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
3443 \ifl@d@Xtwolines%
3444   \ifl@d@Xmorethantwolines%
3445     \csuse{Xmorethantwolines@\@currentseries}%
3446   \else%
3447     \csuse{Xtwolines@\@currentseries}%
3448   \fi%
3449 \else%
3450   \ifl@d@dash%
3451     \ifdefined\linerangesep%
3452       \linerangesep%

```

```

3453     \else%
3454       \csuse{Xlinerangeseparator@}\@currentseries}%
3455     \fi%
3456   \fi%
3457   \ifl@d@pnum%
3458     \wrap@edcrossref{\@this@crossref@end}{#4}%
3459     \csuse{Xpagelinesep@}\@currentseries}%
3460   \fi%
3461   \ifl@d@elin%
3462     \wrap@edcrossref{\@this@crossref@end}{%
3463       \linenumrep{#5}%
3464       \iftoggle{Xlineflag@}\@currentseries}{#8}{}%
3465     }%
3466   \fi%
3467   \ifl@d@esl%
3468     \ifl@d@elin%
3469       \csuse{Xsublinesep@}\@currentseries}%
3470     \fi%
3471     \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3472   \fi%
3473 \fi%
3474 \ifdimequal{\csuse{Xboxendlinenum@}\@currentseries}{0pt}%
3475 {}%
3476 {\hfill}%Prevent underfull hbox
3477 \egroup%
3478 \endgroup%
3479 }%
3480 %

```

XII.7.5 Footnote grouped by line

`\prepare@Xgroupbyline` `\prepare@Xgroupbyline` is a macro called on on the `\(XXX)vfootnote` if `\Xgroupbyline` is set to true, instead of calling directly the `\insert`.

```

3481 \newcommand{\prepare@Xgroupbyline}[3]{%
3482   \iftoggle{Xgroupbylineseparetwolines@#1}{%
3483     \l@dparsfootspec#2%
3484     \ifcsdef{#1@forinserting@}\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3485   {%
3486     \csgappto%
3487       {#1@forinserting@}\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3488   {%
3489     \ifcsemt{Xsymlinenum@#1}%
3490       {\csuse{Xparafootsep@#1}}%
3491     {}%
3492     #3{#1}{#2}%
3493     \hskip\csuse{Xafternote@#1}\relax%
3494   }%

```

```

3495 }%
3496 {%
3497   \csdef%
3498     {#1@forinserting@l@dparsedendpage-\l@dparsedendline-\
l@dparsedendsub}%
3499   {%
3500     #3{#1}{#2}%
3501     \hskip\csuse{Xafternote@#1}\relax%
3502   }%
3503 }%
3504   \listcsxadd{#1@forinserting}{\l@dparsedendpage-\l@dparsedendline-\
l@dparsedendsub}%
3505 }{%
3506   \ifcsdef{#1@forinserting@all}{%
3507     \csgappto%
3508       {#1@forinserting@all}%
3509     {%
3510       \ifcseempty{Xsymlinenumber@#1}%
3511         {\csuse{Xparafootsep@#1}}%
3512       {}%
3513       #3{#1}{#2}%
3514       \hskip\csuse{Xafternote@#1}\relax%
3515     }%
3516   }%
3517   {%
3518     \csdef%
3519       {#1@forinserting@all}%
3520     {%
3521       #3{#1}{#2}%
3522       \hskip\csuse{Xafternote@#1}\relax%
3523     }%
3524   }%
3525   \listcsgadd{#1@forinserting}{all}%
3526 }%
3527 }%
3528 %

```

XIII Familiar footnotes

XIII.1 Adjacent footnotes

The original edmac provided users with five series of critical footnotes (`\Afootnote` `\Bfootnote` `\Cfootnote` `\Dfootnote` `\Efootnote`), and \TeX provides a single numbered footnote. The `reledmac` package uses the edmac mechanism to provide six series of numbered footnotes.

First, though, the `footmisc` package has an option whereby two or more consecutive `\footnotes` have their marks separated by commas. This seemed to Peter Wilson such a useful ability that it was provided automatically by `eledmac`.

Maïeul Rouquette has maintained this feature in `reledmac`, despite he thought that is not directly in relationship with the aim of `reledmac`.

`\multiplefootnotemark` 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`.

```
3529 \providecommand*\multiplefootnotemark}{3sp}
3530 \providecommand*\multfootsep{\textsuperscript{\normalfont,}}
3531
3532 %
```

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

```
3533 \providecommand*\m@mmf@prepare}{%
3534   \kern-\multiplefootnotemark
3535   \kern\multiplefootnotemark\relax}
3536 %
```

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

```
3537 \providecommand*\m@mmf@check}{%
3538   \ifdim\lastkern=\multiplefootnotemark\relax
3539     \edef\x@sf{\the\spacefactor}%
3540     \unkern
3541     \multfootsep
3542     \spacefactor\x@sf\relax
3543   \fi}
3544
3545 %
```

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

```
3546 \@ifclassloaded{memoir}{}{%
3547   %
```

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

```
3548 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
3549 %
```

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

```
3550
3551 \patchcmd{\@footnotemark}
3552   {\nobreak}
3553   {\m@mmf@check
```



```

3554 \nobreak
3555 }
3556 {}{}
3557 \patchcmd{\@footnotemark}
3558 {\@makefnmark}
3559 {\@makefnmark
3560 \m@mmf@prepare
3561 }
3562 {}{}
3563 %

```

Finished the modifications for the non-memoir case.

```

3564 }
3565
3566 %

```

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.

```

3567 \pretocmd{\@footnotetext}{%
3568 \ifnumberedpar@
3569 \edtext{}{\l@dbfnote{#1}}}%
3570 \else
3571 }{}{}
3572 \apptocmd{\@footnotetext}{\fi}{}{}%
3573 %

```

`\l@dbfnote` `\l@dbfnote` adds the footnote to the insert list, and `\vl@dbfnote` calls the original `\@footnotetext`. We also patch `\footnote` in order to get the correct footnote numbers when typesetting parallel texts. This is moved into a `\get@fnmark` command.

```

\footnote
\get@fnmark
\get@thisfootnote
3574
3575 \patchcmd%
3576 {\footnote}%
3577 {\stepcounter\@mpfn}%
3578 {%
3579 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3580 \global\advance\footnote@reading by \@ne%
3581 \get@thisfootnote%
3582 \get@fnmark{\thisc@footnote}%
3583 \ifcsdef{footnotereading\the\footnote@reading=typeset}%
3584 {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%

```

```

3585     {\setcounter{\@mpfn}{\footnote@reading}}}%
3586   }{%
3587     \stepcounter{\@mpfn}%
3588   }%
3589 }%
3590 {}
3591 {}
3592
3593 \newcommand{\get@thisfootnote}{%
3594   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3595     l@dprintingcolumns}}{%
3596     \protected@xdef\thisc@footnote{\the\footnote@reading}%
3597   }{%
3598     \protected@xdef\thisc@footnote{\the\c@footnote}%
3599   }%
3600 }%
3601
3602 \newcommand{\l@dbfnote}[1]{%
3603   \get@thisfootnote%
3604   \gdef\@tag{#1\relax}%
3605   \ifledRcol%
3606     \xright@appenditem{%
3607       \ifdefined\Hy@footnote@currentHref%
3608         \noexpand\def\noexpand\Hy@footnote@currentHref{\
3609         Hy@footnote@currentHref}%
3610       \fi%
3611       \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
3612     }%
3613     \to\inserts@listR
3614     \global\advance\insert@countR \@ne%
3615   \else%
3616     \xright@appenditem{%
3617       \ifdefined\Hy@footnote@currentHref%
3618         \noexpand\def\noexpand\Hy@footnote@currentHref{\
3619         Hy@footnote@currentHref}%
3620       \fi%
3621       \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
3622     }%
3623     \to\inserts@list
3624     \global\advance\insert@count \@ne%
3625   \fi
3626   \ignorespaces%
3627 }%
3628
3629 \newcommand{\get@fnmark}[1]{%
3630   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3631     l@dprintingcolumns}}{%
3632     {%
3633       \stepcounter{footnote@typeset}%
3634       \setcounter{footnote}{\c@footnote@typeset}%

```

```

3631 \immediate\write\@mainaux{%
3632 \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
3633 }%
3634 \def\@thefnmark{\thefootnote}%
3635 }%
3636 {%
3637 \setcounter{footnote}{#1}%
3638 \def\@thefnmark{\thefootnote}%
3639 }%
3640 }%
3641
3642 \newcommand{\v1@dbfnote}[2]{%
3643 \get@fnmark{#2}%
3644 \@footnotetext{#1}%
3645 }%
3646 %

```

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
3647 \newcommand*\prebodyfootmark{%
3648 \leavevmode
3649 \ifhmode
3650 \edef\x@sf{\the\spacefactor}%
3651 \m@mzf@check
3652 \nobreak
3653 \fi}
3654 \newcommand*\postbodyfootmark{%
3655 \m@mzf@prepare
3656 \ifhmode\spacefactor\x@sf\fi\relax}
3657
3658 %

```

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.

```

3659 \newcommandx{\arrangementX}[2][1,usedefault]{%
3660 \def\do##1{%
3661 \csname arrangementX@#2\endcsname{##1}%
3662 }%
3663 \ifstrempy{#1}%

```

```

3664 { %
3665 \dolistloop{\@series}%
3666 } %
3667 {
3668 \docsvlist{#1}%
3669 } %
3670 } %
3671 %

```

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.

```

3672 \newcommand*{\normal@footnotemarkX}[1]{ %
3673 \prebodyfootmark
3674 \wrapped@bodyfootmarkX{#1}%
3675 \postbodyfootmark}
3676
3677 %

```

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

```

3678 \newcommand*{\normalbodyfootmarkX}[1]{ %
3679 \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}}
3680 %

```

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

```

3681 \notbool{parapparatus@}{\newcommand*}{\newcommand*}{\normalvfootnoteX}[2]{ %
3682 \csuse{beforeinsertingX@#1}%
3683 \insert\@nameuse{footins#1}\bgroup
3684 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3685 \noindent\csuse{bhooknoteX@#1}%
3686 \csuse{notefontsizeX@#1}%
3687 \footsplitskips
3688 \ifl@dpairing\ifl@dpaging\else%
3689 \setnoteswidthliketwocolumnsX@{#1}%
3690 \fi\fi%
3691 \setnotesXpositionliketwocolumns@{#1}%
3692 \spaceskip=\z@skip \xspaceskip=\z@skip
3693 \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
3694
3695 %

```

`\mpnormalvfootnoteX` The minipage version.

```

3696 \newcommand*{\mpnormalvfootnoteX}[2]{%
3697   \get@thisfootnoteX{#1}%
3698   \get@fnmarkX{#1}{\thisc@footnote}%
3699   \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3700   \global\setbox\@nameuse{mpfootins#1}\vbox{%
3701     \unvbox\@nameuse{mpfootins#1}
3702     \noindent\csuse{bhooknoteX@#1}%
3703     \csuse{notefontsizeX@#1}%
3704     \hsize\columnwidth
3705     \@parboxrestore
3706     \color@begingroup
3707     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
3708
3709 %

```

\normalfootfmtX `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

3710 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmtX}[2]{%
3711   \ifluatex%
3712     \texdir\footnote@luatextextdir%
3713     \pardir\footnote@luatexpardir%
3714     \par%
3715   \fi%
3716   \protected@edef\@currentlabel{%
3717     \@nameuse{@thefnmark#1}%
3718   }%
3719   \ledsetnormalparstuffX{#1}%
3720   \hangindent=\csuse{hangindentX@#1}%
3721   \everypar{\hangindent=\csuse{hangindentX@#1}}%
3722   \rule{z@}{splittopskip}%
3723   {{\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}}%
3724     \csuse{wrapcontentX@#1}{#2}%
3725   \strut\par}}
3726
3727 %

```

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

```

3728 \newcommand*{\normalfootfootmarkX}[1]{%
3729   \textsuperscript{\@nameuse{@thefnmark#1}}
3730
3731 %

```

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

```

3732 \newcommand*{\normalfootstartX}[1]{%
3733   \ifdimequal{Opt}{\prenotesX@}{}%
3734   {%

```

```

3735 \iftoggle{prenotesX@}{%
3736   \togglefalse{prenotesX@}%
3737   \skip\csname footins#1\endcsname=%
3738   \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3739   }%
3740   {%
3741   }%
3742   \vskip\skip\csname footins#1\endcsname%
3743   \leftskip=\z@
3744   \rightskip=\z@
3745   \ifl@dpairing\else%
3746     \hsize=\old@hsize%
3747   \fi%
3748   \setnoteswidthliketwocolumnsX@{#1}%
3749   \setnotesXpositionliketwocolumns@{#1}%
3750   \print@footnoteXrule{#1}%
3751 }%
3752 %
3753 %

```

\normalfootnoteruleX The rule drawn before the footnote series group.

```

3754 \let\normalfootnoteruleX=\footnoterule
3755
3756 %

```

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

```

3757 \newcommand*{\normalfootgroupX}[1]{%
3758   \csuse{bhookgroupX@#1}%
3759   \unvbox\@nameuse{footins#1}%
3760   \hsize=\old@hsize%
3761   }%
3762
3763 %

```

\mpnormalfootgroupX The minipage version.

```

3764 \newcommand*{\mpnormalfootgroupX}[1]{%
3765   \vskip\skip\@nameuse{mpfootins#1}
3766   \ifl@dpairing\ifparledgroup%
3767     \leavevmode\marks\parledgroup@{begin}%
3768     \marks\parledgroup@series{#1}%
3769     \marks\parledgroup@type{footnoteX}%
3770   \fi\fi\normalcolor
3771   \ifparledgroup%
3772     \ifl@dpairing%
3773     \else%
3774       \setnoteswidthliketwocolumnsX@{#1}%

```

```

3775 \setnotesXpositionliketwocolumns@{#1}%
3776 \print@footnoteXrule{#1}%
3777 \fi%
3778 \else%
3779 \setnoteswidthliketwocolumnsX@{#1}%
3780 \setnotesXpositionliketwocolumns@{#1}%
3781 \print@footnoteXrule{#1}%
3782 \fi%
3783 \csuse{bhookgroupX@#1}%
3784 \unvbox\@nameuse{mpfootins#1}}
3785
3786 %

```

```

\normalbfnoteX
3788 \newcommand{\normalbfnoteX}[2]{%
3789 \get@thisfootnoteX{#1}%
3790 \ifledRcol%
3791 \ifluatex
3792 \footnotelang@lua[R]%
3793 \fi
3794 \@ifundefined{xpg@main@language}%if polyglossia
3795 {}%
3796 {\footnotelang@poly[R]}%
3797 \xright@appenditem{%
3798 \noexpand\led@set@index@fornote{#1}%
3799 \noexpand\prepare@edindex@fornote{\led@nums}%
3800 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3801 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
3802 \noexpand\led@reinit@index@fornote%
3803 \unexpanded{\advance\@edindex@fornote@m@one}%
3804 }%
3805 \to\inserts@listR
3806 \global\advance\insert@countR \@ne%
3807 \else%
3808 \ifluatex
3809 \footnotelang@lua%
3810 \fi
3811 \@ifundefined{xpg@main@language}%if polyglossia
3812 {}%
3813 {\footnotelang@poly}%
3814 \xright@appenditem{%
3815 \noexpand\led@set@index@fornote{#1}%
3816 \noexpand\prepare@edindex@fornote{\led@nums}%
3817 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3818 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
3819 \noexpand\led@reinit@index@fornote%
3820 \unexpanded{\advance\@edindex@fornote@m@one}%

```

```

3821 }%
3822         \to\inserts@list
3823         \global\advance\insert@count \@ne%
3824     \fi
3825     \ignorespaces}
3826
3827 %

```

\get@thisfootnoteX The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

3828 \newcommand{\get@thisfootnoteX}[1]{%
3829     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3830         l@dprintingcolumns}}{%
3831         \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
3832             endcsname}%
3833     }{%
3834         \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}%
3835     }%
3836 }%
3837 %

```

\vbfnoteX This command calls the correct footnote-inserting commands.

```

3836 \newcommand{\vbfnoteX}[3]{%
3837     \get@fnmarkX{#1}{#3}%
3838     \@nameuse{regvfootnote#1}{#1}{#2}%
3839 }%
3840
3841 %

```

\get@fnmarkX This command gets the correct footnote number when typesetting parallel texts.

```

3842 \newcommand{\get@fnmarkX}[2]{%
3843     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3844         l@dprintingcolumns}}{%
3845         {%
3846             \stepcounter{footnote#1@typeset}%
3847             \setcounter{footnote#1}{\value{footnote#1@typeset}}%
3848             \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3849             \immediate\write\@mainaux{%
3850                 \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
3851                     @typeset\endcsname}%
3852             }%
3853         }%
3854     }{%
3855         \setcounter{footnote#1}{#2}%
3856         \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3857     }%
3858 }

```



```
3857 %
3858 %
```

```
\vnumfootnoteX59 \newcommand{\vnumfootnoteX}[2]{%
3860   \ifnumberedpar@
3861     \edtext{}{\normalbfnoteX{#1}{#2}}%
3862   \else
3863     \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3864     \get@thisfootnoteX{#1}%
3865     \get@fnmarkX{#1}{\expandonce\thisc@footnote}%
3866     \@nameuse{regvfootnote#1}{#1}{#2}%
3867   \fi}
3868
3869 %
```

`arrangementX@normal` `\arrangementX@normal{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```
3870 \newcommand*{\arrangementX@normal}[1]{%
3871   \csgdef{series@displayX#1}{normal}
3872   \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
3873   \@namedef{footnotemark#1}{\normal@footnotemarkX{#1}}
3874   \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
3875   \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
3876   \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
3877   \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
3878   \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
3879   \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
3880   \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3881   \count\csname footins#1\endcsname=1000
3882   \csxdef{default@footins#1}{1000}%Use to have note only for one side
3883   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3884   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3885   \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3886   %
```

Additions for minipages.

```
3887 \ifnoledgroup@else%
3888   \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3889   \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
3890   \count\csname mpfootins#1\endcsname=1000
3891   \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
3892   \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3893   \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3894 \fi
3895 }
3896
3897 %
```

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 398 \newcommand*\arrangementX@twocol}[1]{%
3899 \csgdef{series@displayX#1}{twocol}
3900 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
3901 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
3902 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
3903 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3904 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3905 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3906 \twocolfootsetupX{#1}
3907 \ifnoledgroup@%
3908 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3909 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
3910 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3911 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
3912 \mptwocolfootsetupX{#1}
3913 \fi%
3914 }
3915
3916 %

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
3917 \newcommand*\twocolfootsetupX[1]{%
3918 \count\csname footins#1\endcsname 500
3919 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
3920 \multiply\dimen\csname footins#1\endcsname by \tw@}
3921 \newcommand*\mptwocolfootsetupX[1]{%
3922 \count\csname mpfootins#1\endcsname 500
3923 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
3924
3925 %

\twocolvfootnoteX \twocolvfootnoteX{<series>}
3926 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
3927 \csuse{beforeinsertingX@#1}%
3928 \insert\csname footins#1\endcsname\bgroup%
3929 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3930 \noindent\csuse{bhooknoteX@#1}%
3931 \csuse{notefontsizeX@#1}%
3932 \footssplitskips%
3933 \spaceskip=\z@skip \xspaceskip=\z@skip%
3934 \@nameuse{footfmt#1}{#1}{#2}\egroup}
3935

```

```

3936 %
\twocolfootfmtX \twocolfootfmtX{<series>}
3937 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
3938   \protected@edef\@currentlabel{%
3939     \@nameuse{@thefnmark#1}%
3940   }%
3941   \normal@pars%
3942   \hangindent=\csuse{hangindentX@#1}%
3943   \everypar{\hangindent=\csuse{hangindentX@#1}}%
3944   \hspace{\csuse{hsizetwocolX@#1}%
3945   \nottoggle{parindentX@#1}{\parindent=\z@}{}}%
3946   \tolerance=5000\relax%
3947   \par%
3948   \@tempdima=\parindent%
3949   \csuse{colalignX@#1}%
3950   \parindent=\@tempdima%
3951   {\hspace{\parindent}%
3952   \csuse{notenumberX@#1}\wrapped@footfootmarkX{#1}\strut%
3953   \csuse{wrapcontentX@#1}{#2}%
3954   \strut\par}%
3955   \allowbreak%
3956 }%
3957 %
3958 %

```

```

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX
3959 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3960 \splittopskip=\ht\strutbox
3961 \expandafter
3962 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
3963
3964 \newcommand*{\mptwocolfootgroupX}[1]{%
3965   \vskip\skip\@nameuse{mpfootins#1}
3966   \ifl@dpairing\ifparledgroup%
3967     \leavevmode\marks\parledgroup@{begin}%
3968     \marks\parledgroup@series{#1}%
3969     \marks\parledgroup@type{footnoteX}%
3970   \fi\fi\normalcolor
3971   \ifparledgroup%
3972     \ifl@dpairing%
3973     \else%
3974       \setnoteswidthliketwocolumnsX@{#1}%
3975       \setnotesXpositionliketwocolumns@{#1}%
3976       \print@footnoteXrule{#1}%
3977     \fi%
3978   \else%

```

```

3979 \setnoteswidthliketwocolumnsX@{#1}%
3980 \setnotesXpositionliketwocolumns@{#1}%
3981 \print@footnoteXrule{#1}%
3982 \fi%
3983 \csuse{bhookgroupX@#1}%
3984 \splittopskip=\ht\strutbox
3985 \expandafter
3986 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
3987
3988 %

```

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 39 \newcommand*{\arrangementX@threecol}[1]{%
3990 \csgdef{series@displayX#1}{threecol}
3991 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
3992 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
3993 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
3994 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3995 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3996 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3997 \threecolfootsetupX{#1}
3998 \ifnoledgroup@ \else%
3999 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4000 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4001 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4002 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4003 \mpthreecolfootsetupX{#1}
4004 \fi%
4005 }
4006
4007 %

```

```

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX \newcommand*{\threecolfootsetupX}[1]{%
4008 \count\csname footins#1\endcsname 333
4009 \csxdef{default@footins#1}{333}%Use this to confine the notes to one
4010 side only
4011 \multiply\dimen\csname footins#1\endcsname by \thr@@
4012 \newcommand*{\mpthreecolfootsetupX}[1]{%
4013 \count\csname mpfootins#1\endcsname 333
4014 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@
4015
4016 %

```

`\threecolvfootnoteX` `\threecolvfootnoteX{<series>}{<text>}`

```

4017 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
4018 %
4019 \csuse{beforeinsertingX@#1}%
4020 \insert\csname footins#1\endcsname\bgroup%
4021 \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4022 \noindent\csuse{bhooknoteX@#1}%
4023 \csuse{notefontsizeX@#1}%
4024 \footssplitsskip%
4025 \@nameuse{footfmt#1}{#1}{#2}\egroup}
4026 %

```

`\threecolfootfmtX` `\threecolfootfmtX{<series>}`

```

4027 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4028 \protected@edef\@currentlabel{%
4029 \@nameuse{@thefnmark#1}%
4030 }%
4031 \hangindent=\csuse{hangindentX@#1}%
4032 \everypar{\hangindent=\csuse{hangindentX@#1}}%
4033 \normal@pars%
4034 \hspace \csuse{hsizethreecolX@#1}%
4035 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4036 \tolerance=5000\relax%
4037 \@tempdima=\parindent%
4038 \csuse{colalignX@#1}%
4039 \parindent=\@tempdima%
4040 {\hspace{\parindent}%
4041 \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4042 \csuse{wrapcontentX@#1}{#2}%
4043 \strut\par}\allowbreak}
4044
4045 %

```

`\threecolfootgroupX` `\threecolfootgroupX{<series>}`

```

\mpthreecolfootgroupX
4046 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4047 \splittopskip=\ht\strutbox
4048 \expandafter
4049 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4050
4051 \newcommand*{\mpthreecolfootgroupX}[1]{%
4052 \vskip\skip\@nameuse{mpfootins#1}
4053 \ifl@dpairing\ifparledgroup
4054 \leavevmode\marks\parledgroup@{begin}%
4055 \marks\parledgroup@series{#1}%
4056 \marks\parledgroup@type{footnoteX}%

```

```

4057 \fi\fi\normalcolor
4058 \ifparledgroup%
4059   \ifl@dpairing%
4060   \else%
4061     \setnoteswidthliketwocolumnsX@{#1}%
4062     \setnotesXpositionliketwocolumns@{#1}%
4063     \print@footnoteXrule{#1}%
4064   \fi%
4065 \else%
4066   \setnoteswidthliketwocolumnsX@{#1}%
4067   \setnotesXpositionliketwocolumns@{#1}%
4068   \print@footnoteXrule{#1}%
4069 \fi%
4070 \csuse{bhookgroupX@#1}%
4071 \splittopskip=\ht\strutbox
4072 \expandafter
4073 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4074
4075 %

```

XIII.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@threecol` `\footparagraphX{<series>}`

```

4076 \newcommand*{\arrangementX@paragraph}[1]{%
4077   \csgdef{series@displayX#1}{paragraph}%
4078   \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4079   \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4080   \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4081   \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4082   \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4083   \count\csname footins#1\endcsname=1000
4084   \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
4085   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4086   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4087   \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4088   \para@footsetupX{#1}
4089   \ifnoledgroup@else
4090     \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4091     \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4092     \count\csname mpfootins#1\endcsname=1000
4093     \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4094     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4095     \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4096   \fi
4097 }
4098

```

4099 %

`\para@footsetupX` `\para@footsetupX{<series>}`

```
4100 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4101 \setnoteswidthliketwocolumnsX@{#1}%
4102 \ifcsempy{widthX@#1}%
4103 {}%
4104 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4105 \dimen0=\baselineskip
4106 \multiply\dimen0 by 1024
4107 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
4108 %
4109 \expandafter
4110 \xdef\csname footfudgefactor#1\endcsname{%
\expandafter\strip@pt\dimen0 }}
4111 %
4112 %
```

`\parafootstartX` `\parafootstartX{<series>}`

```
4113 \newcommand*{\parafootstartX}[1]{%
4114 \ifdimequal{0pt}{\prenotesX@}{}%
4115 {%
4116 \iftoggle{prenotesX@}{%
4117 \togglefalse{prenotesX@}%
4118 \skip\csname footins#1\endcsname=%
4119 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4120 }%
4121 }%
4122 }%
4123 \leftskip=\z@
4124 \rightskip=\z@
4125 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4126 \vskip\skip\@nameuse{footins#1}%
4127 \setnoteswidthliketwocolumnsX@{#1}%
4128 \setnotesXpositionliketwocolumns@{#1}%
4129 \print@footnoteXrule{#1}%
4130 }
4131 %
4132 %
```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`

`\mppara@vfootnoteX`

```
4133 \newcommand*{\para@vfootnoteX}[2]{%
4134 \csuse{beforeinsertingX@#1}%
4135 \insert\csname footins#1\endcsname%
4136 \bgroup
4137 \csuse{notefontsizeX@#1}
```

```

4138 \footsplitskips
4139 \setbox0=\vbox{\hsize=\maxdimen%
4140 \let\bidirTL@everypar\@empty%
4141 \noindent\csuse{bhooknoteX@#1}%
4142 \@nameuse{footfmt#1}{#1}{#2}}%
4143 \setbox0=\hbox{\unvvhX{0}{#1}}%
4144 \dp0=\z@
4145 \ht0=\csname footfudgefactor#1\endcsname\wd0
4146 \box0
4147 \penalty0
4148 \egroup}
4149 \newcommand*\mppara@vfootnoteX}[2]{%
4150 \get@thisfootnoteX{#1}%
4151 \get@fnmarkX{#1}{\thisc@footnote}%
4152 \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4153 \global\setbox\@nameuse{mpfootins#1}\vbox{%
4154 \unvvhX\@nameuse{mpfootins#1}
4155 \csuse{notefontsizeX@#1}
4156 \footsplitskips
4157 \setbox0=\vbox{\hsize=\maxdimen%
4158 \let\bidirTL@everypar\@empty%
4159 \noindent\color@begingroup%
4160 \csuse{bhooknoteX@#1}%
4161 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
4162 \setbox0=\hbox{\unvvhX{0}{#1}}%
4163 \dp0=\z@
4164 \ht0=\csname footfudgefactor#1\endcsname\wd0
4165 \box0
4166 \penalty0}}
4167
4168 %

```

```

\unvvhX69 \newcommand*\unvvhX}[2]{% 2th is optional for retro-compatibility

```

```

4170 \setbox0=\vbox{\unvvhX#1%
4171 \global\setbox1=\lastbox}%
4172 \unhbox1
4173 \unskip % remove \rightskip,
4174 \unskip % remove \parfillskip,
4175 \unpenalty % remove \penalty of 10000,
4176 \hskip\csuse{afternoteX@#2}%
4177 \relax}% but add the glue to go between the notes
4178
4179 %

```

```

\parafootfmtX \parafootfmtX{<series>}

```

```

4180 \newcommand*\parafootfmtX}[2]{%
4181 \protected@edef\@currentlabel{%
4182 \@nameuse{@thefnmark#1}%

```



```

4183 }%
4184 \insertparafootsepX{#1}%
4185 \ledsetnormalparstuff@common%
4186 {\csuse{notenumfontX@#1}%
4187  \csuse{notenumfontX@#1}%
4188  \wrapped@footfootmarkX{#1}%
4189  \strut%
4190  \csuse{wrapcontentX@#1}{#2}%
4191  \penalty-10}}
4192
4193 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`

`\mppara@footgroupX`

```

4194 \newcommand*{\para@footgroupX}[1]{%
4195  \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4196  \unvbox\csname footins#1\endcsname
4197  \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4198  \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4199  \makehboxofhboxes
4200  \setbox0=\hbox{\unhbox0 \removehboxes}%
4201  \csuse{hookgroupX@#1}
4202  \csuse{notefontsizeX@#1}
4203  \unhbox0\par}
4204
4205 \newcommand*{\mppara@footgroupX}[1]{%
4206  \setnoteswidthliketwocolumnsX@{#1}%
4207  \vskip\skip\@nameuse{mpfootins#1}
4208  \ifl@dpairing\ifparledgroup
4209    \leavevmode%
4210    \leavevmode\marks\parledgroup@{begin}%
4211    \marks\parledgroup@series{#1}%
4212    \marks\parledgroup@type{footnoteX}%
4213    \fi\fi\normalcolor
4214    \ifparledgroup%
4215      \ifl@dpairing%
4216        \else%
4217          \setnoteswidthliketwocolumnsX@{#1}%
4218          \setnotesXpositionliketwocolumns@{#1}%
4219          \print@footnoteXrule{#1}%
4220        \fi%
4221      \else%
4222        \setnoteswidthliketwocolumnsX@{#1}%
4223        \setnotesXpositionliketwocolumns@{#1}%
4224        \print@footnoteXrule{#1}%
4225      \fi%
4226    \unvbox\csname mpfootins#1\endcsname
4227    \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4228    \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4229    \makehboxofhboxes

```

```

4230 \setbox0=\hbox{\unhbox0 \removehboxes}%
4231 \csuse{bhookgroupX@#1}%
4232 \csuse{notefontsizeX@#1}%
4233 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4234 \unhbox0\par}}
4235
4236 %

```

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

```

\insertparafootsepX \newcommand{\insertparafootsepX}[1]{%
4238   \ifledRcol{%
4239     \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
4240       {\csuse{Xparafootsep@#1}}%
4241       {}%
4242     \global\csname prevpage#1@numR\endcsname=\page@numR%
4243   \else%
4244     \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
4245       {\csuse{Xparafootsep@#1}}%
4246       {}%
4247     \global\csname prevpage#1@num\endcsname=\page@num%
4248   \fi%
4249 }
4250 %

```

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.

```

4251 \newcommand{\wrapped@footfootmarkX}[1]{%
4252   \ifdefined\hypertarget%
4253     \hyperlink%
4254       {@bodyfootmark#1@this@footnoteX@reading}%
4255       {@nameuse{footfootmark#1}}%
4256     \Hy@raisedlink{%
4257       \hypertarget%
4258       {@footnotemark#1@this@footnoteX@reading}%
4259       {}%
4260     }%
4261   \else%
4262     \@nameuse{footfootmark#1}%
4263   \fi%
4264 }%
4265 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package’s commands, if needed.

```

4266 \newcommand{\wrapped@bodyfootmarkX}[1]{%
4267   \ifdefined\hypertarget%
4268     \hyperlink%
4269       {@footnotemark#1\expandafter\the\csname footnote#1@reading\
endcsname}%
4270     {@nameuse{bodyfootmark#1}}%
4271     \Hy@raisedlink{%
4272       \hypertarget%
4273         {@bodyfootmark#1\expandafter\the\csname footnote#1@reading\
endcsname}%
4274       }%
4275     }%
4276   \else%
4277     \@nameuse{bodyfootmark#1}%
4278   \fi%
4279 }%
4280 %

```

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%81 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff%82   \ifluatex%
\ledsetnormalparstuffX%83     \textdir\footnote@luatextextdir%
4284     \pardir\footnote@luatexpardir%
4285     \fi%
4286     \csuse{\csuse{footnote@dir}}%
4287     \normal@pars%
4288     \parfillskip \z@ \@plus 1fil}%
4289
4290 \newcommand*{\Xledsetnormalparstuff}[1]{%
4291   \ledsetnormalparstuff@common%
4292   \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
4293 }%
4294

```

```

4295 \newcommand*\ledsetnormalparstuffX}[1]{%
4296   \ledsetnormalparstuff@common%
4297   \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}}%
4298 }%
4299 %

```

XV Footnotes' width for two columns

We define here some commands which make sense only with `reledpar`, but must be called when defining notes parameters. These commands change the width of block notes to allow them to have the same size than two parallel columns.

`\old@hsize` These two commands are called at the beginning of critical or familiar notes groups. They set, if the option is enabled, the `\hsize`. They are also called at the on the setup for paragraphed notes.

`\setXnoteswidthliketwocolumns@`

`\setnoteswidthliketwocolumnsX@`

```

4300
4301 \newdimen\old@hsize%
4302 \AtBeginDocument{\old@hsize=\hsize}%
4303
4304 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
4305   \global\let\hsize@fornote=\hsize%
4306   \global\old@hsize=\hsize%
4307   \let\old@columnwidth=\columnwidth%
4308   \iftoggle{Xnoteswidthliketwocolumns@#1}%
4309     {%
4310       \csuse{setwidthliketwocolumns@\columns@position}%
4311       \global\let\hsize@fornote=\hsize%
4312     }%
4313   {}%
4314   \let\hsize=\hsize@fornote%
4315   \let\columnwidth=\old@columnwidth%
4316 }%
4317
4318 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4319   \global\let\hsize@fornote=\hsize%
4320   \global\old@hsize=\hsize%
4321   \let\old@columnwidth=\columnwidth%
4322   \iftoggle{noteswidthliketwocolumnsX@#1}%
4323     {%
4324       \csuse{setwidthliketwocolumns@\columns@position}%
4325       \global\let\hsize@fornote=\hsize%
4326     }%
4327   {}%
4328   \let\hsize=\hsize@fornote%
4329   \let\columnwidth=\old@columnwidth%
4330 }%
4331
4332 %

```

`\setnotespositionliketwocolumns@` These two commands set the position of the critical / familiar footnotes, depending on the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They call commands which are defined only in `reledpar`, because this feature has no sens without `reledpar`.

```

4333 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
4334   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4335     \csuse{setnotespositionliketwocolumns@\columns@position}%
4336   }{}%
4337 }%
4338
4339 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
4340   \iftoggle{noteswidthliketwocolumnsX@#1}{%
4341     \csuse{setnotespositionliketwocolumns@\columns@position}%
4342   }{}%
4343 }%
4344
4345 %

```

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.

```

4346 \def\@fnpos{familiar-critical}
4347 \def\@mpfnpos{critical-familiar}
4348 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
4349 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
4350 %

```

XVII Footnotes' rule

Because the footnotes' rules can be shifted to the right when footnotes are set like two columns, we do not print them directly, but we put them in a `\vbox`.

```

\print@Xfootnoterule\newcommand{\print@Xfootnoterule}[1]{%
\print@footnoteXrule\  \vskip-\csuse{Xafterterrule@#1}%Because count in \dimen\csuse{#1footins}
4353 \nointerlineskip%
4354 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
4355 \nointerlineskip%
4356 \vskip\csuse{Xafterterrule@#1}%
4357 }%
4358
4359 \newcommand{\print@footnoteXrule}[1]{%
4360   \vskip-\csuse{afterterruleX@#1}%Because count in \dimen\csuse{footins#1}
4361   \nointerlineskip%
4362   \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%

```

```

4363 \nointerlineskip%
4364 \vskip\csuse{afterruleX@#1}%
4365 }%
4366
4367 %

```

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.

```

4368 \newtoggle{Xprenotes@}%
4369 \toggletrue{Xprenotes@}%
4370 \newcommand{\Xprenotes@}{0pt}%

```

```

4371 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
4372 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
4373 %

```

The same, but for familiar footnotes.

```

\Xprenotes 74 \newtoggle{prenotesX@}
\Xprenotes@ 75 \toggletrue{prenotesX@}
4376 \newcommand{\prenotesX@}{Opt}
4377 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
4378 %

```

XVIII.0.3 Internal commands

```

firstXseries@ 79 \gdef\firstXseries@{}
prepare@Xprenotes 80 \newcommand{\prepare@Xprenotes}[1]{%
4381   \ifdimequal{Opt}{\Xprenotes@}%
4382   {%
4383     {%
4384       \IfStrEq{\firstXseries@}{%
4385         \global\skip\csuse{#1footins}=\Xprenotes@%
4386         \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4387         \gdef\firstXseries@{#1}%
4388       }%
4389     {%
4390       \ifseriesbefore{#1}{\firstXseries@}%
4391       {%
4392         \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
4393         \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4394         \gdef\firstXseries@{#1}%
4395       }%
4396     }%
4397   }%
4398 }%
4399 }
4400 %

```

The same thing is required for familiar notes and \prenotesX.

```

firstseriesX@ 01 \gdef\firstseriesX@{}
prepare@prenotesX 02 \newcommand{\prepare@prenotesX}[1]{%
4403   \ifdimequal{Opt}{\prenotesX@}%
4404   {%
4405     {%

```

```

4406 \IfStrEq{\firstseriesX@}{}{\%
4407 \global\skip\csuse{footins#1}=\prenotesX@%
4408 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4409 \gdef\firstseriesX@{#1}%
4410 }%
4411 {%
4412 \ifseriesbefore{#1}{\firstseriesX@}%
4413 {%
4414 \global\skip\csuse{footins#1}=\csuse{beforenotesX@firstseriesX@}%
4415 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4416 \gdef\firstXseries@{#1}%
4417 }%
4418 {%
4419 }%
4420 }%
4421 }
4422 %

```

XIX Endnotes

First, check the noend option.

```

4423 \ifbool{noend@}{}{\%Used instead of \ifnoend@ to prevent expansion problem
4424 %

```

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.

```

4425 \newcommand{\l@dend@open}[1]{\%
4426 \global\booltrue{l@dend@#1}%
4427 \expandafter\immediate%
4428 \expandafter\openout%
4429 \csname l@d@#1end\endcsname%
4430 =\l@auxdir\jobname.#1end\relax%
4431 }%
4432 \newcommand{\l@dend@close}[1]{\%
4433 \global\boolfalse{l@dend@#1}%
4434 \expandafter\immediate%
4435 \expandafter\closeout\csname l@d@#1end\endcsname%
4436 }%
4437
4438 %

```


`\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.

```

4439 \newcommand{\l@dend@stuff}{%
4440   \def\do##1{%
4441     \ifbool{l@dend@##1}{}%
4442     {\l@dend@open{##1}}%
4443     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
4444   }%
4445   \dolistloop{\@series}%
4446 }%
4447
4448 %

```

`\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.

```

4449 \global\newboolean{parapparatus@}\def\endprint#1#2#3#4#5#6#7{%
4450   \csuse{Xendhooknote@#4}%
4451   \csuse{Xendnotefontsize@#4}%
4452   \hangindent=\csuse{Xendhangindent@#4}%
4453   \ifXendinsertsep%
4454     \hskip\csuse{Xendafternote@#4}\relax%
4455     \csuse{Xendsep@#4}%
4456   \else%
4457     \iftoggle{Xendparagraph@#4}%
4458       {\global\Xendinsertsep@true}%
4459     {}%
4460   \fi%
4461   \xdef\@currentseries{#4}%
4462   \def\do##1{%

```

```

4463 \setkeys[mac]{truefootnoteoption}{##1}%
4464 }%
4465 \notblank{#5}{\docsvlist{#5}}{}%
4466 \IfStrEq{#6}{R}{\ledRcol@true}{}%
4467 \def\@this@crossref@start{#7:start}%
4468 \def\@this@crossref@end{#7:end}%
4469 \printlineendnote{#1}{#4}%
4470 \IfStrEq{#6}{R}{\ledRcol@false}{}%
4471 \undef\@this@crossref@start%
4472 \undef\@this@crossref@end%
4473 \nottoggle{Xendlemmadisablefontselection@#4}%
4474 {\select@lemmafont#1|}%
4475 {}%
4476 \bgroup%
4477 \csuse{Xendlemmafont@#4}%
4478 \csuse{Xendwraplemma@#4}{#2}%
4479 \egroup%
4480 \ifboolexpr{%
4481   togl {nosep@}%
4482   or test{\ifcsemt{Xendlemmaseparator@#4}}%
4483 }%
4484 {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
4485 {\nobreak%
4486   \hskip\csuse{Xendbeforelemmaseparator@#4}%
4487   \csuse{Xendlemmaseparator@#4}%
4488   \hskip\csuse{Xendafterlemmaseparator@#4}%
4489   \relax}%
4490 }%
4491 \csuse{Xendwrapcontent@#4}{#3}%
4492 \nottoggle{Xendparagraph@#4}{\par}{}%
4493 \def\do##1{%
4494   \setkeys[mac]{falsefootnoteoption}{##1}%
4495 }%
4496 \notblank{#5}{\docsvlist{#5}}{}%
4497 }}%
4498
4499 \let\l@d@section=\@gobble
4500
4501 %

```

\printlineendnote This macro controls, in endnote, whether the line number is printed or not, according to the series options. Its first argument is the information about lines; its second is the series of the footnote.

```

4502 \newcommand{\printlineendnote}[2]{%
4503   \l@dp@rsefootspec#1|%
4504   \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
4505     \edef\lineinfo{\l@dparsedstartpage - \l@dparsedstartline - \
4506       \l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
4507       \l@dparsedendsub}%

```

```

4506     }%
4507     {%
4508     \edef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
l@dparsedstartsub}%
4509     }%
4510     \ifboolexpr{%
4511     togl {nonum@}%
4512     or togl {Xendnonumber@#2}%
4513     }%
4514     {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4515     {%
4516     \iftoggle{Xendnumberonlyfirstinline@#2}%
4517     {\ifcsdef{prevendline#2}%
4518     {\ifcsequal{prevendline#2}{\lineinfo@}%
4519     {%
4520     \csuse{Xendbhookinplaceofnumber@#2}%
4521     \ifcsequal{Xendsymmlinenumber@#2}%
4522     {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4523     {\printsymlineendnotearea{#2}}}%
4524     \csuse{Xendahookinplaceofnumber@#2}%
4525     }%
4526     {\printlineendnotearea{#1}{#2}}}%
4527     {\printlineendnotearea{#1}{#2}}}%
4528     }%
4529     {\printlineendnotearea{#1}{#2}}%We keep every time line
4530     \csxdef{prevendline#2}{\lineinfo@}%
4531     }%
4532 }%
4533 %

```

```

\printsymlineendnotearea \newcommand{\printsymlineendnotearea}[1]{%
4535 \hspace{\csuse{Xendbeforesymmlinenumber@#1}}%
4536 \csuse{Xendnotenumfont@#1}%
4537 \ifdimequal{\csuse{Xendboxsymmlinenumber@#1}}{\z@}%
4538 {\csuse{Xendsymmlinenumber@#1}}%
4539 {\hbox to \csuse{Xendboxsymmlinenumber@#1}%
4540 {\csuse{Xendsymmlinenumber@#1}\hfill}}%
4541 }%
4542 \hspace{\csuse{Xendaftersymmlinenumber@#1}}%
4543 }%
4544 %

```

\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.)

```

4545 \newcommand{\printlineendnotearea}[2]{%
4546 \csuse{Xendbhooklinenumber@#2}%

```

```

4547 \hspace{\csuse{Xendbeforenumber@#2}}}%
4548 \bgroup%
4549 \csuse{Xendnotenumfont@#2}%
4550 \ifdimequal{\csuse{Xendboxlinenum@#2}}{0pt}%
4551 {\printendlines#1||\ifledRcol@{\Rlineflag\fi}%
4552 {\leavevmode%
4553 \hbox to \csuse{Xendboxlinenum@#2}%
4554 {%
4555 \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4556 \printendlines#1||\ifledRcol@{\Rlineflag\fi}%
4557 \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4558 }}%
4559 \egroup%
4560 \hspace{\csuse{Xendafternumber@#2}}}%
4561 \csuse{Xendahooklinenum@#2}%
4562 }%
4563 %

```

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*64 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes*65 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4566 \def\do##1{%
4567 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4568 {\unexpanded{#2}\@percentchar}%
4569 }%
4570 \ifstrempy{#1}%
4571 {\dolistloop{\@series}}%
4572 {\docsvlist{#1}}%
4573 }{\led@err@toendnotes@outsidenumbering}%
4574 }%
4575 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
4576 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4577 \def\do##1{%
4578 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4579 {#2\@percentchar}%
4580 }%
4581 \ifstrempy{#1}%
4582 {\dolistloop{\@series}}%
4583 {\docsvlist{#1}}%
4584 }{\led@err@toendnotes@outsidenumbering}%
4585 }%

```

4586 %

XIX.2.2 Printing endnotes

`\doendnotes` `\ifXendinsertsep@` `\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.

```
4587 \newif\ifXendinsertsep@%
4588 \newcommand*{\doendnotes}[1]{%
4589   \l@dend@close{#1}%
4590   \begingroup
4591     \csxdef{prevpagenum@#1}{}%
4592     \csxdef{prevpagerange@#1}{}%
4593     \makeatletter
4594     \expandafter\let\csname #1end\endcsname=\endprint
4595     \input\l@auxdir\jobname.#1end%
4596     \global\Xendinsertsep@false%
4597   \endgroup}
4598 %
```

`\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.

```
4599 \newcommand*{\doendnotesbysection}[1]{%
4600   \l@dend@close{#1}%
4601   \csxdef{prevpagenum@#1}{}%
4602   \csxdef{prevpagerange@#1}{}%
4603   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
4604   \begingroup%
4605     \makeatletter%
4606     \def\l@d@section##1{%
4607       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
4608       {\cslet{#1end}{\endprint}}%
4609       {\cslet{#1end}{\@gobblefive}}%
4610     }%
4611     \input\l@auxdir\jobname.#1end%
4612     \global\Xendinsertsep@false%
4613   \endgroup%
4614 }%
4615 %
```

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.

```
4616 }%
4617 %
```

`\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.

```
4618 \newcommand*{\setprintendlines}[6]{%
4619   \l@dpnumfalse \l@ddashfalse
4620   \ifnum#4=#1 \else
4621     \l@dpnumtrue
4622     \l@ddashtrue
4623   \fi
4624   %
```

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.

```
4625   \ifl@dpnum \l@d@elintrue \else \l@d@elinfalse \fi
4626   \ifnum#2=#5 \else
4627     \l@d@elintrue
4628     \l@ddashtrue
4629   \fi
4630   %
```

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

```
4631   \l@d@ssubfalse
4632   \ifnum#3=0 \else
4633     \l@d@ssubtrue
4634   \fi
4635   %
```

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.

```
4636   \l@d@eslfalse
4637   \ifnum#6=0 \else
4638     \ifnum#6=#3
4639       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4640     \else
4641       \l@d@esltrue
4642       \l@ddashtrue
4643     \fi
4644   \fi%
4645   %
4646   \ifl@d@dash%
```

```

4647 \ifbool{expr{togl{fulllines@} or test{\ifcempty{Xendtwolines@}\
@currentseries}}}%
4648 {}%
4649 {%
4650 \setistwofollowinglines{#1}{#2}{#4}{#5}%
4651 \ifbool{expr{%
4652 (%
4653     togl {Xendtwolinesbutnotmore@\@currentseries}%
4654     and not%
4655     (%
4656         bool {istwofollowinglines@}%
4657     )%
4658 )%
4659 or%
4660 (%
4661     (not test{\ifnumequal{#1}{#4}})%
4662     and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
4663 )%
4664 }%
4665 {}%
4666 {%
4667 \l@d@dashfalse%
4668 \l@d@Xtwolinestrue%
4669 \l@d@elinfalse%
4670 \l@d@eslfalse%
4671 \ifcempty{Xendmorethantwolines@\@currentseries}%
4672 {}%
4673 {\ifistwofollowinglines@\else%
4674     \l@d@Xmorethantwolinestrue%
4675 \fi%
4676 }%
4677 }%
4678 }%
4679 \fi%
4680 %

```

End of \setprintendlines.

```

4681 }%
4682 %

```

\printendlines Now we are ready to print it all.

```

4683 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4684 \begingroup
4685 \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
4686 %

```

The only subtlety left here is when to print a period between numbers. But the only instance in which this is tricky is for the ending sub-line number: it could be coming

after the starting sub-line number (in which case we want only the dash) or after an ending line number (in which case we need to insert a period).

So, first, start the start line box, if required.

```

4687 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
4688 {\bgroup}%
4689 {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}%
4690 %

```

Then, print the start page-

```

4691 \ifboolexpr{%
4692 (%)
4693 test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
4694 and not%
4695 (togl{Xendpagenumberonlyfirstifsingle@\@currentseries} and bool{
1@d@pnum})}%
4696 )%
4697 or%
4698 (%
4699 test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
4700 )%
4701 }%
4702 {%
4703 \ifcempty{Xendsympagenum@\@currentseries}%
4704 {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
4705 {\csuse{Xendsympagenum@\@currentseries}}%
4706 }%
4707 {%
4708 \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
4709 }%
4710 %

```

Then, determine what must be printed before the start line.

```

4711 \ifl@d@dash%
4712 \ifl@d@pnum%
4713 \csuse{Xendlineprefixsingle@\@currentseries}%
4714 \else%
4715 \ifcempty{Xendlineprefixmore@\@currentseries}%
4716 {\csuse{Xendlineprefixsingle@\@currentseries}}%
4717 {\csuse{Xendlineprefixmore@\@currentseries}}%
4718 \fi%
4719 \else%
4720 \csuse{Xendlineprefixsingle@\@currentseries}%
4721 \fi%
4722 %

```

The print the start line, following, if required, by the side flag and the start sub line.

```

4723 \wrap@edcrossref{\@this@crossref@start}{%
4724 \ifledRcol@%

```



```

4725 \linenumrepR{#2}%
4726 \else%
4727 \linenumrep{#2}%
4728 \fi%
4729 }%
4730 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi}{}%
4731 \ifl@d@ssub%
4732 \csuse{Xendsublinesep@ \@currentseries}%
4733 \wrap@edcrossref{\@this@crossref@start}{%
4734 \ifledRcol@%
4735 \sublinenumrepR{#3}%
4736 \else%
4737 \sublinenumrep{#3}%
4738 \fi%
4739 }%
4740 \fi%
4741 %

```

Close the box.

```

4742 \egroup%
4743 %

```

Open the box for the end line.

```

4744 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
4745 {\bgroup}%
4746 {\hbox to \csuse{Xendboxendlinenum@ \@currentseries}\bgroup}%
4747 %

```

Print the dash + the end line number, or the line number range symbol.

```

4748 \ifl@d@xtwolines%
4749 \ifl@d@Xmorethantwolines%
4750 \csuse{Xendmorethantwolines@ \@currentseries}%
4751 \else%
4752 \csuse{Xendtwolines@ \@currentseries}%
4753 \fi%
4754 \else%
4755 \ifl@d@dash%
4756 \ifdefined\linerangesep@%
4757 \linerangesep@%
4758 \else%
4759 \csuse{Xendlinerangeseparator@ \@currentseries}%
4760 \fi%
4761 \fi%
4762 %

```

Print the end page number.

```

4763 \ifl@d@pnum%
4764 \ifcsstring{prevpagerange@ \@currentseries}{#1-#4}%
4765 {%

```

```

4766 \ifcempty{Xendsympagenum@\@currentseries}%
4767 {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
4768 {\csuse{Xendsympagenum@\@currentseries}}}%
4769 }%
4770 {%
4771 \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
4772 }%
4773 \fi%
4774 %

```

Print the end line number, with if required the line prefix, and followed by the side flag and the subline number.

```

4775 \ifl@d@elin%
4776 \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
4777 \wrap@edcrossref{\@this@crossref@end}{%
4778 \ifledRcol@%
4779 \linenumrepR{#5}%
4780 \else%
4781 \linenumrep{#5}%
4782 \fi%
4783 }%
4784 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
4785 \fi%
4786 \ifl@d@esl%
4787 \ifl@d@elin%
4788 \csuse{Xendsublinesep@\@currentseries}%
4789 \fi%
4790 \wrap@edcrossref{\@this@crossref@end}{%
4791 \ifledRcol@%
4792 \sublinenumrepR{#6}%
4793 \else%
4794 \sublinenumrep{#6}%
4795 \fi%
4796 }%
4797 \fi%
4798 \fi%
4799 %

```

Close the end line box.

```

4800 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
4801 {}%
4802 {\hfill}%Prevent underfull hbox
4803 \egroup%
4804 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

4805 \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
4806 {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%

```

```

4807     {\csxdef{prevpagerange@ \@currentseries}{#1-#4}}%
4808     {\csxdef{prevpagenum@ \@currentseries}{#4}}%
4809     }%
4810     {}%
4811 %

```

Now, the end of \printendlines macro.

```

4812 \endgroup%
4813 }%
4814
4815 %

```

\printnpnum A macro to print a page number in an endnote. Should not be override anymore

```

4816 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@ \@currentseries
}#1\csuse{Xendafterpagenumber@ \@currentseries}}
4817
4818 %

```

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@.

```

4819 \newcommand{\newseries}[1]{%
4820     \def\do##1{\newseries@{##1}}%
4821     \docsvlist{#1}
4822 }
4823 %

```

\@series The \series@ macro is an etoolbox list, which contains the name of all series.

```

4824 \newcommand{\@series}{}
4825 %

```

The command \newseries@\series creates a new series of the footnote.

\newseries@₂₆ \newcommand{\newseries@}[1]{

```

4827 %

```

XX.1 Test if series is still existing

```

4828 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
4829 {%
4830 %

```

XX.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```

4831 \ifdefined\newseries@par%
4832 \newseries@par{#1}%
4833 \fi%
4834 %

```

XX.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```

4835 \unless\ifnocritical@
4836 %

```

XX.3.1 Options

```

4837 \newtoggle{Xlineflag@#1}
4838 \newtoggle{Xparindent@#1}
4839 \newtoggle{Xlemmadisablefontselection@#1}
4840 \csgdef{Xwrapcontent@#1}{}%
4841 \csgdef{Xbeforeinserting@#1}{}%
4842 \csgdef{Xhangindent@#1}{Opt}%
4843 \csgdef{Xragged@#1}{}%
4844 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
4845 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
4846 \csgdef{Xcolalign@#1}{\raggedright}%
4847 \csgdef{Xnotenumfont@#1}{\normalfont}%
4848 \csgdef{Xnotefontsize@#1}{\footnotesize}%
4849 \csgdef{Xbhooknote@#1}{}%
4850 \csgdef{Xbhookgroup@#1}{}%
4851
4852 \csgdef{Xboxlinenum@#1}{Opt}%
4853 \csgdef{Xboxlinenumalign@#1}{L}%
4854
4855 \csgdef{Xboxstartlinenum@#1}{Opt}%
4856 \csgdef{Xboxendlinenum@#1}{Opt}%
4857
4858 \csgdef{Xboxsymlinenum@#1}{Opt}%
4859 \newtoggle{Xnumberonlyfirstinline@#1}%

```

```

4860 \newtoggle{Xgroupbyline@#1}%
4861 \newtoggle{Xgroupbylineseparetwelines@#1}%
4862 \newtoggle{Xnumberonlyfirstintwelines@#1}%
4863 \csgdef{Xtwelines@#1}{}%
4864 \csgdef{Xmorethantwelines@#1}{}%
4865 \csgdef{Xsublinesep@#1}{\fullstop}%
4866 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
4867 \newtoggle{Xtwelinesbutnotmore@#1}%
4868 \newtoggle{Xtwelinesonlyinsamepage@#1}%
4869 \newtoggle{Xonlypstart@#1}%
4870 \newtoggle{Xpstarteverytime@#1}%
4871 \newtoggle{Xpstart@#1}%
4872 \newtoggle{Xstanza@#1}%
4873 \csgdef{Xstanzaseparator@#1}{}%
4874 \csgdef{Xsymlinenum@#1}{}%
4875 \newtoggle{Xnonumber@#1}%
4876 \csgdef{Xbeforenumber@#1}{0pt}%
4877 \csgdef{Xtxtbeforenumber@#1}{}%
4878 \csgdef{Xafternumber@#1}{0.5em}%
4879 \newtoggle{Xnonbreakableafternumber@#1}%
4880 \csgdef{Xbeforenymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
4881 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
4882 \csgdef{Xinplaceofnumber@#1}{1em}%
4883 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
4884 \csgdef{Xbeforelemmaseparator@#1}{0em}%
4885 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
4886 \csgdef{Xinplaceofflemmaseparator@#1}{1em}%
4887 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
4888 \csgdef{Xafterrule@#1}{0pt}
4889
4890 \csgdef{Xtxtbeforenotes@#1}{%
4891 \newtoggle{Xtxtbeforenotes@#1@typeset}}%Not directly used by user,
but internal
4892
4893 \csgdef{Xmaxhnotes@#1}{0.8\vsizer}
4894 \newtoggle{Xnoteswidthliketwocolumns@#1}%
4895 \csgdef{Xparafootsep@#1}{}%
4896 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%
4897 \csgdef{Xlinrangeseparator@#1}{\endashchar}%
4898
4899 \csgdef{Xlemmafont@#1}{}%
4900 \csgdef{Xwraplemma@#1}{%
4901 \csgdef{Xwidth@#1}{\hsizer}%
4902 %

```

XX.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

4903 \expandafter\newinsert\csname #1footins\endcsname%

```

```

4904 \unless\ifnoledgroup%
4905 \expandafter\newinsert\csname mp#1footins\endcsname%
4906 \fi%
4907 %

```

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.

```

4908 \global\newbool{parapparatus@}\expandafter\newcommand\expandafter
*{\expandafter\newcommand\csname #1footnote\endcsname[2] []{%
4909 \if@edtext@secondarg%
4910 \ifledRcol%
4911 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}}%
4912 \else%
4913 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}}%
4914 \fi%
4915 \begingroup%
4916 \newcommand{\content}{##2}%
4917 \ifnumberedpar%
4918 \ifledRcol%
4919 \ifluatex%
4920 \footnotelang@lua[R]%
4921 \fi%
4922 \@ifundefined{xpg@main@language}%if polyglossia
4923 {}%
4924 {\footnotelang@poly[R]}%
4925 \footnoteoptions@{R}{##1}{true}%
4926 \xright@appenditem{%
4927 \ifbool{indtl@innote}%
4928 {\unexpanded{\let\index\nindex}}%
4929 {}%
4930 \ifbool{indtl@notenumber}%
4931 {\unexpanded{\let\index\nindex}}%There is no note
...number so
4932 {}%
4933 \noexpand\Xnote@true%
4934 \noexpand\prepare@Xprenotes{#1}%
4935 \noexpand\prepare@edindex@fornote{\led@nums}%
4936 \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.
4937 \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
4938 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%

```

```

4939         \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
4940         \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
4941         \noexpand\csuse{v#1footnote}{#1}%
4942         {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
4943         \noexpand\Xnote@false%
4944         \unexpanded{\advance\@edindex@fornote@m@ne}%
4945         \unexpanded{\undef\@this@crossref@start}%
4946         \unexpanded{\undef\@this@crossref@end}%
4947         \ifbool{indtl@innote}%
4948             {\unexpanded{\let\index\orig@@index}}%
4949             {}%
4950         \ifbool{indtl@notenumber}%
4951             {\unexpanded{\let\index\orig@@index}}%
4952             {}%
4953         }\to\inserts@listR
4954         \footnoteoptions@{R}{##1}{false}%
4955         \global\advance\insert@countR \@ne%
4956     \else%
4957         \ifluatex%
4958             \footnotelang@lua%
4959             \fi%
4960         \@ifundefined{xpg@main@language}%if polyglossia
4961             {}%
4962             {\footnotelang@poly}%
4963         \footnoteoptions@{L}{##1}{true}%
4964         \xright@appenditem{%
4965             \ifbool{indtl@innote}%
4966                 {\unexpanded{\let\index\nindex}}%
4967                 {}%
4968             \ifbool{indtl@notenumber}%
4969                 {\unexpanded{\let\index\nindex}}%There is no note
...number so
4970             {}%
4971             \noexpand\Xnote@true%
4972             \noexpand\prepare@Xprenotes{#1}%
4973             \noexpand\prepare@edindex@fornote{\l@d@nums}%
4974             \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.
4975             \ifl@dpairing%
4976                 \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
4977             \fi%
4978             \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
4979             \unexpanded{\def\@this@crossref@end}{\theedtext:end}%

```

```

4980 \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
4981 \noexpand\csuse{v#1footnote}%
4982 {#1}%
4983 {{\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
4984 \unexpanded{\undef\@this@crossref@start}%
4985 \unexpanded{\undef\@this@crossref@end}%
4986 \noexpand\Xnote@false%
4987 \unexpanded{\advance\@edindex@fornote@m@ne}%
4988 \ifbool{indtl@innote}%
4989 {\unexpanded{\let\index\orig@@index}}%
4990 {}%
4991 \ifbool{indtl@notenumber}%
4992 {\unexpanded{\let\index\orig@@index}}%
4993 {}%
4994 }\to\inserts@list
4995 \global\advance\insert@count \@ne%
4996 \footnoteoptions@{L}{#1}{false}%
4997 \fi
4998 \else
4999 \csuse{v#1footnote}{#1}{{0|0|0|0|0|0|0|0}{#1}}%
5000 \fi%
5001 \endgroup%
5002 \else%
5003 \led@err@FootnoteNotInSecondArgEdtext{#1}%
5004 \fi%
5005 \ignorespaces%
5006 }
5007 %

```

Create counter used to determine on which page the previous note was called.

```

5008 \expandafter\newcount\csname #1prevpage@num\endcsname%
5009 \expandafter\newcount\csname #1prevpage@numR\endcsname%
5010 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

5011 \global\csletcs{#1@@footnote}{#1footnote}
5012 %

```

XX.3.4 Set standard display

```

5013 \Xarrangement@normal{#1}%
5014 %

```

End of for critical footnotes.

```

5015 \fi
5016 %

```


XX.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```
5017 \unless\ifnofamiliar@
5018 %
```

XX.4.1 Options

```
5019 \newtoggle{parindentX@#1}
5020 \csgdef{wrapcontentX@#1}{}%
5021 \csgdef{hangindentX@#1}{Opt}%
5022 \csgdef{beforeinsertingX@#1}{}%
5023 \csgdef{raggedX@#1}{}%
5024 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
5025 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
5026 \csgdef{colalignX@#1}{\raggedright}%
5027 \csgdef{notenumfontX@#1}{\normalfont}%
5028 \csgdef{notefontsizeX@#1}{\footnotesize}%
5029 \csgdef{bhooknoteX@#1}{}%
5030 \csgdef{bhookgroupX@#1}{}%
5031 \csgdef{afterruleX@#1}{Opt}
5032 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
5033 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
5034 \newtoggle{noteswidthliketwocolumnsX@#1}%
5035 \csgdef{parafootsepX@#1}{}%
5036 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
5037 \csgdef{widthX@#1}{\hsizex}%
5038 % End of for familiar footnotes.
5039 % \subsubsection{Create inserts, needed to add notes in foot}
5040 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
5041 % \begin{macrocode}
5042 \expandafter\newinsert\csname footins#1\endcsname%
5043 \unless\ifnoledgroup%
5044 \expandafter\newinsert\csname mpfootins#1\endcsname%
5045 \fi%
5046 %
```

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.

```
5047 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
5048 \begingroup%
5049 \prepare@prenotesX{#1}%
5050 \newcommand{\content}{##1}%
5051 %
5052 %
```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.1.2 p. 48).

```

5053         \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
5054         \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or
bool{!@dprintingcolumns}}{%
5055             \ifcsdef{footnote#1@reading\the\csname footnote#1@reading\
endcsname=typeset}%
5056                 {\setcounter{footnote#1}{\csuse{footnote#1@reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
5057                 {\setcounter{footnote#1}{\the\csname footnote#1@reading
\endcsname}}}%
5058             }{%
5059                 \stepcounter{footnote#1}%
5060             }%
5061 %

```

We also have to check consistency with `\onlysideX` setting.

```

5062         \ifledRcol%
5063             \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}{%
5064                 \else%
5065                     \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}{%
5066                         \fi%
5067 %

```

And now, the feature not depending of wether we are preparing parallel typesetting

```

5068         \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5069         \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
5070             {\csuse{@footnotemark#1}}%
5071             {}%
5072         \ifluatex%
5073             \xdef\footnote@luatextextdir{\the\textdir}%
5074             \xdef\footnote@luatexpardir{\the\pardir}%
5075         \fi%
5076         \if@ledgroup%
5077             \led@set@index@fornote{#1}%
5078         \fi%
5079         \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
5080         \ifbool{indtl@innote}%
5081             {\let\index\orig@@index}%
5082             {}%
5083         \ifbool{indtl@notenumber}%
5084             {\let\index\orig@@index}%
5085             {}%
5086         \endgroup%
5087     }

```

5088 %

Then define the counters. The \LaTeX counter `footnoteX` is the only one manipulated by the user. This is the one which is printed. The \TeX counter `\footnoteX@reading` is increased at each footnote. It is used for hyperlinks, for using `hyperlink` package, and for getting the correct footnote number when using parallel typesetting (V.1.2 p. 48).

```
5089 \newcounter{footnote#1}
5090 \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
5091 \expandafter\newcount\csname footnote#1@reading\endcsname%
5092 %
```

Create counter used to determine on which page the previous note was called.

```
5093 \expandafter\newcount\csname prevpage#1@num\endcsname%
5094 \expandafter\newcount\csname prevpage#1@numR\endcsname%
5095 %
```

Add `\let\footnoteX\@gobble` to `\no@expands`.

```
5096 \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
5097 %
```

Do not forget to initialize the series

```
5098 \arrangementX@normal{#1}%
5099 \fi
5100 %
```

XX.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```
5101 \unless\ifnoend@
5102 %
```

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`

```
5103 \expandafter\newwrite\csname l@d@#1end\endcsname%
5104 \expandafter\newif\csname ifl@dend@#1\endcsname%
5105 %
```

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.

```

5106      \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,
5107      usedefault]{%
5108          \bgroup%
5109          \newlinechar='40%
5110          \global\@noneed@Footnotetrue%
5111          \newcommand{\content}{##2}%
5112          \stepcounter{labidx}%
5113          \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
5114          \unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\
expandafter{\sw@inthisedtext}}\@percentchar\space%Voluntary space, to add
linebreak in the output file
5115          \expandafter\string\csname #1end\endcsname%
5116          {\ifnumberedpar@l@d@nums\fi}%
5117          {\ifnumberedpar@expandonce\@tag\fi}%
5118          {\expandonce\content}%
5119          {\#1}%
5120          {\unexpanded{##1}}%
5121          {\ifledRcol R\else L\fi}%
5122          {\theedtext}%
5123          \@percentchar%
5124          }%
5125          \egroup%
5126          \ignorespaces%
5127      }%
5128      %

```

XX.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoenotes`

```

5129      \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
5130      \toendnotes[#1]{##1}%
5131      }%
5132
5133      \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
endcsname*[1]{%
5134      \toendnotes*[#1]{##1}%
5135      }%
5136
5137

```

5138 %

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.

5139
5140 \global\cslet{#1end}{\@gobblefive}
5141 %

We need to store the number of times \doendnotesbysection is called for one series.

5142 \global\expandafter\newcount\csname #1end@bysection\endcsname%
5143 %

XX.5.5 The options

5144 \csgdef{Xendwraplemma@#1}{%
5145 \csgdef{Xendwrapcontent@#1}{}%
5146 \csgdef{Xendtwolines@#1}{}%
5147 \csgdef{Xendmoreethantwolines@#1}{}%
5148 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
5149 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
5150 \newtoggle{Xendlemmadisablefontselection@#1}%
5151 \csgdef{Xendnotenumfont@#1}{\normalfont}%
5152 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
5153 \csgdef{Xendbhooknote@#1}{}%
5154
5155 \csgdef{Xendsublinesep@#1}{\fullstop}%
5156
5157 \csgdef{Xendbeforenumber@#1}{0pt}
5158 \csgdef{Xendafternumber@#1}{0.5em}
5159
5160 \csgdef{Xendboxlinenum@#1}{0pt}%
5161 \csgdef{Xendboxlinenumalign@#1}{L}%
5162
5163 \csgdef{Xendboxstartlinenum@#1}{0pt}%
5164 \csgdef{Xendboxendlinenum@#1}{0pt}%
5165
5166 \csgdef{Xendlemmaseparator@#1}{}%
5167 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
5168 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
5169 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
5170
5171 \newtoggle{Xendparagraph@#1}%

```

5172 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
5173 \csgdef{Xendsep@#1}{}%
5174
5175 \csgdef{Xendinplaceofnumber@#1}{Opt}%
5176 \newtoggle{Xendnonnumber@#1}%
5177
5178 \csgdef{Xendhangindent@#1}{Opt}%
5179 \newtoggle{Xendnumberonlyfirstinline@#1}%
5180 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
5181
5182 \csgdef{Xendbeforesymmlinenumber@#1}{\csuse{Xendbeforenumber@#1}}%
5183 \csgdef{Xendaftersymmlinenumber@#1}{\csuse{Xendafternumber@#1}}%
5184 \csgdef{Xendsymmlinenumber@#1}{}%
5185 \csgdef{Xendboxsymmlinenumber@#1}{Opt}%
5186
5187 \csgdef{Xendbhooklinenumber@#1}{}%
5188 \csgdef{Xendehooklinenumber@#1}{}%
5189 \csgdef{Xendbhookinplaceofnumber@#1}{}%
5190 \csgdef{Xendehookinplaceofnumber@#1}{}%
5191
5192 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%
5193
5194 \csgdef{Xendbeforepagenumber@#1}{p.}%
5195 \csgdef{Xendafterpagenumber@#1}{) }%
5196 \csgdef{Xendlineprefixsingle@#1}{}%
5197 \csgdef{Xendlineprefixmore@#1}{}%
5198
5199 \newtoggle{Xendlineflag@#1}
5200
5201 \csgdef{Xendlemmafont@#1}{}%
5202
5203 \newtoggle{Xendpagenumberonlyfirst@#1}%
5204 \newtoggle{Xendpagenumberonlyfirstifsingle@#1}%
5205 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
5206 \csgdef{Xendsympagenumber@#1}{}%
5207 \csgdef{Xendinplaceofpagenumber@#1}{Opt}%
5208
5209 %

```

End of endnotes declaration

```

5210 \fi%
5211 %

```

Dump series in \@series

```

5212 \listxadd{\@series}{#1}
5213 }
5214 }% End of \newseries
5215 %

```

XX.6 Init standards series (A,B,C,D,E)

```
5216 \expandafter\newseries\expandafter{\default@series}
5217 %
```

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.

```
5218 \newcommand{\seriesatbegin}[1]{%
5219   \StrDel{\@series}{#1}[\@series]%
5220   \edef\@new{}%
5221   \listead{\@new}{#1}%
5222   \listead{\@new}{\@series}%
5223   \xdef\@series{\@new}%
5224 }
5225 %
```

\seriesatend And `\seriesatend` moves the series to the end of the list.

```
5226 \newcommand{\seriesatend}[1]{%
5227   \StrDel{\@series}{#1}[\@series]%
5228   \edef\@new{}%
5229   \listead{\@new}{\@series}%
5230   \listead{\@new}{#1}%
5231   \xdef\@series{\@new}%
5232 }
5233 %
```

XXI.2 Test series order

\ifseriesbefore `\ifseriesbefore{⟨seriesA⟩}{⟨seriesB⟩}{⟨true⟩}{⟨false⟩}` expands `⟨true⟩` if `⟨seriesA⟩` is printed before `⟨seriesB⟩`, expands `⟨false⟩` otherwise.

```
5234 \newcommand{\ifseriesbefore}[4]{%
5235   \StrPosition{\@series}{#1}[\@first]%
5236   \StrPosition{\@series}{#2}[\@second]%
5237   \ifnumgreater{\@second}{\@first}{#3}{#4}%
5238 }
5239 %
```

XXI.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries40 \newcommand{\@getfirstseries}{%
5241 \ifdefempty{\@series}%
5242 {\xdef\@firstseries{}}%
5243 {\StrChar{\@series}{1}{\@firstseries}}%
5244 }%
5245 %

```

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`.


```

5246 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
5247   \def\do##1{%
5248     \global\settoggle{#2@##1}{#3}%
5249     \ifstrequal{#4}{critical}{
5250       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
5251     }{}
5252     \ifstrequal{#4}{familiar}{
5253       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
5254     }{}
5255   }%
5256   \ifstreempty{#1}{%
5257     \dolistloop{\@series}%
5258     \ifstreempty{#5}{}{%
5259       \docsvlist{#5}%
5260     }
5261   }%
5262   {%
5263     \docsvlist{#1}%
5264   }%
5265 }
5266 %

```

\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`.

```

5267 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
5268   \def\do##1{
5269     \csgdef{#2@##1}{#3}
5270     \ifstrequal{#4}{critical}{%
5271       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
5272     }{}
5273     \ifstrequal{#4}{familiar}{%
5274       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
5275     }{}%
5276   }%
5277   \ifstreempty{#1}{%
5278     \dolistloop{\@series}%

```

```

5279         \ifstrempy{#5}{}{%
5280             \docsvlist{#5}
5281         }
5282     }%
5283     {%
5284         \docsvlist{#1}%
5285     }%
5286 }%
5287 %

```

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><pseudoseries>`, like `\Xtwolinesappref`.

```

5288 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
5289   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5290     \setcommand@series{##1}{#1}{##2}[] [#2]%
5291   }%
5292   \ifstrempy{#2}{-}{%
5293     \def\do##1{%
5294       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
5295       [1]{%
5296         \csuse{#1}[##1]{####1}%
5297       }%
5298     }%
5299     \docsvlist{#2}%
5300   }%
5301 }%

```

`\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><pseudoseris>`, like `\Xtwolinesbutnotmoreappref`.

```

5302 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
5303   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
5304     \settoggle@series{##1}{##2}[] [#2]%
5305   }%
5306   \ifstrepty{#2}{}%
5307   \def\do##1{%
5308     \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
5309       \csuse{#1}[##1]%
5310     }%

```

```

5311 }%
5312 \docsvlist{#2}%
5313 }%
5314 }
5315 %

```

\newhooktoggle@series@reload `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement, depending of type os notes

```

5316 \newcommand{\newhooktoggle@series@reload}[2]{%
5317 \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
5318 \settoggle@series{##1}{#1}{##2}[#2]%
5319 }%
5320 }%
5321 %

```

\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.

```

5322 \newcommand{\newhookcommand@series@reload}[2]{%
5323 \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5324 \setcommand@series{##1}{#1}{##2}[#2]%
5325 }%
5326 }
5327 %

```

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.

```

5328 \unless\ifnocritical@
5329 \newhookcommand@series{Xwrapcontent}%
5330 \newhookcommand@series{Xbeforeinserting}%
5331 \newhookcommand@series{Xlemmafont}%
5332 \newhookcommand@series{Xwraplemma}%
5333 \newhooktoggle@series{Xparindent}
5334 \newhookcommand@series{Xhangindent}
5335 \newhookcommand@series{Xragged}
5336 \newhookcommand@series{Xhsizetwocol}
5337 \newhookcommand@series{Xhsizethreecol}
5338 \newhookcommand@series{Xcolalign}%
5339 \newhookcommand@series{Xnotenumfont}
5340 \newhookcommand@series{Xbhooknote}
5341 \newhookcommand@series@reload{Xbhookgroup}{critical}
5342 \newhookcommand@series{Xboxsymlinenum}%

```

```

5343 \newhookcommand@series{Xsymlinenum}
5344 \newhookcommand@series{Xbeforenumber}
5345 \newhookcommand@series{Xtxtbeforenumber}
5346 \newhookcommand@series{Xafternumber}
5347 \newhookcommand@series{Xbeforesymlinenum}
5348 \newhookcommand@series{Xaftersymlinenum}
5349 \newhookcommand@series{Xinplaceofnumber}
5350 \newhookcommand@series{Xlemmaseparator}
5351 \newhookcommand@series{Xbeforelemmaseparator}
5352 \newhookcommand@series{Xafterlemmaseparator}
5353 \newhookcommand@series{Xinplaceoflemmaseparator}
5354 \newhookcommand@series{Xtxtbeforenotes}
5355 \newhookcommand@series@reload{Xafterrule}{critical}
5356 \newhooktoggle@series{Xnumberonlyfirstinline}
5357 \newhooktoggle@series{Xnumberonlyfirstintwolines}
5358 \newhooktoggle@series{Xgroupbyline}%
5359 \newhooktoggle@series{Xgroupbylineseparetwolines}%
5360 \newhooktoggle@series{Xnonumber}
5361 \newhooktoggle@series{Xpstart}
5362 \newhooktoggle@series{Xpstarteverytime}%
5363
5364 \newhooktoggle@series{Xstanza}%
5365 \newhookcommand@series{Xstanzaseparator}%
5366
5367 \newhooktoggle@series{Xonlypstart}
5368 \newhooktoggle@series{Xnonbreakableafternumber}
5369 \newhooktoggle@series{Xlemmadisablefontselection}
5370 \newhookcommand@series@reload{Xmaxhnotes}{critical}
5371 \newhookcommand@series@reload{Xbeforenotes}{critical}
5372 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
5373 \newhookcommand@series@reload{Xnotefontsize}{critical}
5374
5375 \newhookcommand@series{Xboxlinenum}%
5376 \newhookcommand@series{Xboxlinenumalign}%
5377
5378 \newhookcommand@series{Xboxstartlinenum}%
5379 \newhookcommand@series{Xboxendlinenum}%
5380
5381 \newhookcommand@series{Xafternote}%
5382 \newhookcommand@series{Xparafootsep}
5383
5384 \newhookcommand@series@reload{Xwidth}{critical}%
5385
5386 \ifundef{\Xhsize}%
5387 {%
5388   \newcommandx{\Xhsize}[2][1,usedefault]{%
5389     \led@warning@Xhsize@deprecated%
5390     \Xwidth[#1]{#2}%
5391   }%
5392 }%

```

```

5393     {}%
5394 \fi
5395 \newhooktoggle@series{Xlineflag}[appref,SEref]
5396 \newhookcommand@series{Xtwolines}[appref,SEref]
5397 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
5398 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
5399 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
5400 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
5401 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
5402 \newhookcommand@series{Xlinrangeseparator}[appref,SEref]
5403 %

```

XXI.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

5404 \unless\ifnofamiliar@
5405   \newhookcommand@series{wrapcontentX}%
5406   \newhookcommand@series{beforeinsertingX}%
5407   \newhooktoggle@series{parindentX}
5408   \newhookcommand@series{hangindentX}
5409   \newhookcommand@series{raggedX}
5410   \newhookcommand@series{hsizetwocolX}
5411   \newhookcommand@series{hsizethreecolX}
5412   \newhookcommand@series{colalignX}%
5413   \newhookcommand@series{notenumfontX}
5414   \newhookcommand@series{bhooknoteX}
5415   \newhookcommand@series@reload{bhookgroupX}{familiar}
5416   \newhookcommand@series@reload{beforenotesX}{familiar}
5417   \newhookcommand@series@reload{maxhnotesX}{familiar}
5418   \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
5419   \newhookcommand@series@reload{afterruleX}{familiar}
5420   \newhookcommand@series@reload{notefontsizeX}{familiar}
5421   \newhookcommand@series{afternoteX}
5422   \newhookcommand@series{parafootsepX}
5423   \newhookcommand@series@reload{widthX}{familiar}%
5424   \ifundef{\hsizeX}%
5425     {%
5426       \newcommandx{\hsizeX}[2][1,usedefault]{%
5427         \led@warning@hsizeX@deprecated%
5428         \widthX[#1]{#2}%
5429       }%
5430     }%
5431   {}%
5432 \fi
5433 %

```

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.

```

5434 \unless\ifnoend@
5435   \newhookcommand@series{Xendwraplemma}
5436   \newhookcommand@series{Xendwrapcontent}
5437   \newhookcommand@series{Xendnotenumfont}
5438   \newhookcommand@series{Xendlemmafont}%
5439   \newhookcommand@series{Xendbhooknote}
5440
5441   \newhookcommand@series{Xendboxlinenum}%
5442   \newhookcommand@series{Xendboxlinenumalign}%
5443
5444   \newhookcommand@series{Xendboxstartlinenum}%
5445   \newhookcommand@series{Xendboxendlinenum}%
5446
5447   \newhookcommand@series{Xendnotefontsize}
5448   \newhooktoggle@series{Xendlemmadisablefontselection}
5449   \newhookcommand@series{Xendlemmaseparator}
5450   \newhookcommand@series{Xendbeforelemmaseparator}
5451   \newhookcommand@series{Xendafterlemmaseparator}
5452   \newhookcommand@series{Xendinplaceoflemmaseparator}
5453
5454   \newhookcommand@series{Xendbeforenumber}%
5455   \newhookcommand@series{Xendafternumber}%
5456
5457   \newhooktoggle@series{Xendparagraph}
5458   \newhookcommand@series{Xendafternote}
5459   \newhookcommand@series{Xendsep}
5460
5461   \newhookcommand@series{Xendinplaceofnumber}%
5462   \newhooktoggle@series{Xendnonumber}%
5463
5464   \newhooktoggle@series{Xendnumberonlyfirstinline}%
5465   \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
5466
5467   \newhookcommand@series{Xendsymlinenum}%
5468   \newhookcommand@series{Xendbeforesymlinenum}%
5469   \newhookcommand@series{Xendaftersymlinenum}%
5470   \newhookcommand@series{Xendboxsymlinenum}%
5471
5472   \newhookcommand@series{Xendbhooklinenumber}%
5473   \newhookcommand@series{Xendahooklinenumber}%
5474   \newhookcommand@series{Xendbhookinplaceofnumber}%
5475   \newhookcommand@series{Xendahookinplaceofnumber}%
5476
5477   \newhookcommand@series{Xendhangindent}%
5478

```

```

5479 \newhooktoggle@series{Xendpagenumberonlyfirst}%
5480 \newhooktoggle@series{Xendpagenumberonlyfirsttifsingle}%
5481 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
5482 \newhookcommand@series{Xendsympagenum}%
5483 \newhookcommand@series{Xendinplaceofpagenumber}%
5484
5485 \fi
5486 \newhooktoggle@series{Xendlineflag}[apprefwithpage,Serefwithpage]
5487 \newhookcommand@series{Xendtwolines}[apprefwithpage,Serefwithpage]
5488 \newhookcommand@series{Xendmorethantwolines}[apprefwithpage,Serefwithpage]
5489 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,Serefwithpage]
5490 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
5491 Serefwithpage]
5492 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,Serefwithpage]
5493 ]
5494 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,Serefwithpage,
5495 Serefonlypage]
5496 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,Serefwithpage]
5497 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,Serefwithpage]
5498 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,Serefwithpage]
5499 \newhookcommand@series{Xendsublinesep}[apprefwithpage,Serefwithpage]
5500
5501 %

```

XXI.4 Hooks for a particular footnote

\newhooktoggle@specific \newhooktoggle@specific is a generic command to create boolean hook specific to a note.

```

5499 \newcommand{\newhooktoggle@specific}[1]{%
5500   \newtoggle{#1}%
5501   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
5502   When enabling footnote option
5503   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false}}
5504 }
5505 %

```

\newhookarg@specific \newhookarg@specific is a generic command to create argumen hook specific to a note.

```

5505 \newcommand{\newhookarg@specific}[1]{%
5506   \define@key[mac]{truefootnoteoption}{#1}{\global\def\linrangesep@{##1}}%
5507   When enabling footnote option
5508   \define@key[mac]{falsefootnoteoption}{#1}{\global\undef\linrangesep@}%
5509   When
5510 }
5511 %

```

And now, we define some hooks specific to a note.

```
5510 \newhooktoggle@specific{fulllines}%
5511 \newhooktoggle@specific{nonum}
5512 \newhooktoggle@specific{nosep}
5513 \newhookarg@specific{linangesep}
5514 %
```

`linangesep@` `\linangesep@` is defined by the option `linangesep` of critical notes to change temporarily the line range separator for a specific line. As we have to define it before typesetting the line and undefine it after, we use the family of `xkeyval` package's key.

```
5515 %
```

`\nomk@` `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```
5516 \newtoggle{nomk}%
5517 %
```

XXI.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```
5518 \newcommand*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]}
5519 %
```

XXII Output routine

Now we begin the output routine and associated things.

XXII.1 Extra footnotes output

With luck we might only have to change `\@makecol` and `\@reinserts` of the \TeX 's kernel. Since `reledmac`, we use `etoolbox`'s patching commands instead of overriding. It should provides better compatibility with other package which modify these commands

`\doxtrafeet` `\doxtrafeet` is the code extending `\@makecol` to cater for the extra `reledmac` feet. We have two categories of extra footnotes. By default, we order the footnote inserts so that the regular footnotes of \TeX are first, then familiar familiar footnotes and finally the critical footnotes.

```
5520 \newcommand*{\l@ddoxtrafeet}{%
5521   \IfStrEq{familiar-critical}{\@fnpos}
5522   {\do@feetX\do@Xfeet}%
5523   {%
5524     \IfStrEq{critical-familiar}{\@fnpos}%
```



```

5525     {\do@Xfeet\do@feetX}%
5526     {%
5527         \setbox\@outputbox \vbox{%
5528             \unvbox\@outputbox%
5529             \do@feet@custom@order{}\@fnpos}%
5530         }%
5531     }%
5532 }%
5533 }%
5534
5535 %

```

\do@feet@custom@order \do@feet@custom@order is called when \@fnpos is neither ‘familiar-critical’, nor ‘critical-familiar’, that is, when the order is more complex. In this case, people must define the order for all footnote series. If they don’t, \TeX could perform an infinite run.

```

5536 \newcommand{\do@feet@custom@order}[2]{%
5537     \def\do##1{%
5538         \edef\@notesseries{\@firstoftwo##1}%
5539         \edef\@notetype{\@secondoftwo##1}%
5540         \ifdefstring{\@notetype}{critical}%
5541             {\csuse{#1append@Xnotes}\@notesseries}}%
5542         {\ifdefstring{\@notetype}{familiar}%
5543             {\csuse{#1append@notesX}\@notesseries}}%
5544             {}%
5545         }%
5546     }%
5547     \expandafter\docsvlist\expandafter{#2}%
5548 }%
5549 %

```

\do@Xfeet \do@Xfeet is the code extending \makecol to cater to the extra critical feet.

```

5550 \newcommand*\do@Xfeet{%
5551     \setbox\@outputbox \vbox{%
5552         \unvbox\@outputbox
5553         \opXfeet}}
5554 %

```

\opXfeet The extra critical feet to be added to the output. . A macro which appends critical notes to the output’s routine, also adding vertical space before notes

```

\append@Xnotes
\print@Xnotes
5555 \newcommand{\append@Xnotes}[1]{%
5556     \ifvoid\csuse{#1footins}\else%
5557         \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
5558         \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
5559         \print@Xnotes{#1}%
5560     \fi%
5561 }%
5562 %

```

The normal way to add one series, `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

5563 \newcommand\print@Xnotes[1]{%
5564   \xdef\@currentseries{#1}%
5565   \csuse{#1footstart}{#1}%
5566   \csuse{#1footgroup}{#1}%
5567 }%
5568 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

5569 \newcommand*\@opXfeet{%
5570   \unless\ifnocritical@%
5571     \gdef\firstXseries@{}%
5572     \def\do##1{%
5573       \append@Xnotes{##1}%
5574     }%
5575     \dolistloop{\@series}%
5576   \fi%
5577 }%
5578 %

```

`\l@ddodoreinextrafeet` `\l@ddodoreinextrafeet` is the code for catering for the extra footnotes within `\@reinserts`. We use the same category and ordering as in `\l@ddoxtrafeet`.

```

5579 \newcommand*\l@ddodoreinextrafeet{%
5580   \IfStrEq{familiar-critical}{\@fnpos}
5581     {\@doreinfeetX\X@doreinfeet}%
5582   {%
5583     \IfStrEq{critical-familiar}{\@fnpos}%
5584       {\X@doreinfeet\@doreinfeetX}%
5585       {\@doreinfeetX\X@doreinfeet}%
5586     }%
5587 }
5588 %
5589 %

```

`\X@doreinfeet` `\X@doreinfeet` is the code for catering for the extra critical footnotes within `\@reinserts`.

```

5590 \newcommand*\X@doreinfeet{%
5591   \unless\ifnocritical@%
5592     \def\do##1{%
5593       \ifvoid\csuse{##1footins}\else%
5594         \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
5595       \fi%
5596     \dolistloop{\@series}
5597   \fi%
5598 }
5599 %
5600 %

```

`\print@notesX` We have to add all the new kinds of familiar footnotes to the output routine. A macro
`\append@notesX` which appends the familiar footnotes of one series onto the output routine, also adding
`\do@feetX` vertical skip before notes.

```

5601 \newcommand{\append@notesX}[1]{%
5602   \ifvoid\csuse{footins#1}\else%
5603     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
5604     \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
5605     \print@notesX{#1}%
5606   \fi%
5607 }%
5608 %

```

The normal way to print one series of notes. `\print@Xnotes` is replaced by `reledpar` when using `\Pages`.

```

5609 \newcommand\print@notesX[1]{%
5610   \xdef\@currentseries{#1}%
5611   \csuse{footstart#1}{#1}%
5612   \csuse{footgroup#1}{#1}%
5613 }%
5614 %

```

We print all the series of notes by looping on them. We check before printing them that they are not voided.

```

5615 \newcommand*\do@feetX{%
5616   \unless\ifnofamiliar@%
5617   \gdef\firstseriesX@{}%
5618   \setbox\@outputbox \vbox{%
5619     \unvbox\@outputbox%
5620     \def\do##1{%
5621       \append@notesX{##1}%
5622     }%
5623     \dolistloop{\@series}%
5624   \fi%
5625 }%
5626
5627 \newcommand{\@doreinfeetX}{%
5628   \unless\ifnofamiliar@%
5629   \def\do##1{%
5630     \ifvoid\csuse{footins##1}\else
5631       \insert%
5632         \csuse{footins##1}
5633         {\unvbox\csuse{footins##1}}%
5634     \fi%
5635   }%
5636   \dolistloop{\@series}%
5637   \fi%
5638 }%
5639
5640 %

```

XXII.2 Patching standard output's commands

The memoir class does not use the ‘standard’ versions of `\@makecol` and `\@reinserts`, due to its sidebar insert. We had better add that code if memoir is used. (It can be awkward dealing with `\if` code within `\if` code, so don't use `\ifl@dmemoir` here.)

```
5641 \@ifclassloaded{memoir}{%
5642 %
```

memoir is loaded so we use memoir's built in hooks.

```
5643 \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
5644 \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
5645 }{%
5646 %
```

memoir has not been loaded, so patch `\@makecol` and `\@reinserts`. If the `fancyhdr` package < version 3.8 has been loaded, we patch the `\latex@makecol` command, because this package redefines the standard `\@makecol` in the preamble, to call `\latex@makecol` which have been `\let` to `\@makecol`. If this package is not loaded, we directly patch `\@makecol`. If the `fancyhdr` package \geq version 3.8, we also directly patch `\@makecol`, because `fancyhdr` does its own patch `\AtBeginDocument`.

```
5647 \ifbool{expr}{%
5648   test{\@ifpackageloaded{fancyhdr}}%
5649   and test {\ifdef{\latex@makecol}}%
5650 }{%
5651   \patchcmd%
5652     {\latex@makecol}%
5653     {\xdef\@freelist{\@freelist\@midlist}}%
5654     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5655     {}%
5656     {\led@error@fail@patch@makecol}%
5657   }{%
5658     \patchcmd%
5659       {\@makecol}%
5660       {\xdef\@freelist{\@freelist\@midlist}}%
5661       {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5662       {}%
5663       {\led@error@fail@patch@makecol}%
5664     }%
5665
5666   \patchcmd%
5667     {\@reinserts}%
5668     {\ifvbox}%
5669     {\l@ddodoreinextrafeet\ifvbox}%
5670     {}%
5671     {\led@error@fail@patch@reinserts}%
5672   }
5673
5674 %
```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

5675 \newif\if@led@nofoot
5676
5677 %

5678 \@ifclassloaded{memoir}{%
5679 %

```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet% \g@addto@macro{\@mem@extranofeet}{%%
5680
5681 \def\do#1{%
5682 \unless\ifnocritical@%
5683 \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
5684 \fi%
5685 \unless\ifnofamiliar@%
5686 \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
5687 \fi%
5688 }
5689 \dolistloop{\@series}%
5690 }%
5691 }{%
5692 %

```

As memoir is not loaded we have patch `\@doclearpage`.

```

\@led@testifnofoot% \newcommand*{\@led@testifnofoot}{%
5693
5694 \@doclearpage% \@led@nofoottrue%
5695 \ifvoid\footins\else%
5696 \@led@nofootfalse%
5697 \fi%
5698 \def\do##1{%
5699 \unless\ifnocritical@%
5700 \ifvoid\csuse{##1footins}\else%
5701 \@led@nofootfalse%
5702 \fi%
5703 \fi%
5704 \unless\ifnofamiliar@%
5705 \ifvoid\csuse{footins##1}\else%
5706 \@led@nofootfalse%
5707 \fi%
5708 \fi%
5709 }%
5710 \dolistloop{\@series}%
5711 }%
5712

```

```

5713 \pretocmd%
5714   {\@docclearpage}%
5715   {\@led@testifnofoot}%
5716   {}%
5717   {\led@error@fail@patch@@docclearpage}%
5718
5719 \patchcmd%
5720   {\@docclearpage}%
5721   {\ifvoid\footins}%
5722   {\if@led@nofoot}%
5723   {}%
5724   {\led@error@fail@patch@@docclearpage}%
5725
5726 }
5727
5728 %

```

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.

```

5729 \list@create{\labelref@list}
5730 %

```

\zz@@@ A convenience macro to zero two labeling counters in one go.

```

5731 \newcommand*{\zz@@@}{000|000} % set two counters to zero in one go
5732
5733 %

```

\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

³²The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

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.

```

5734 \newcommand*{\edlabel}[1]{%
5735   \ifl@dpairing\ifautopar%
5736     \strut%
5737   \fi\fi%
5738   \@bsphack%
5739   \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
5740     \ifXnote@%
5741       \protected@write\@auxout{%
5742         {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
5743         l@dparsedstartsub|\the\c@pstartR|{#1}}}%
5744       \ifdef{\hypertarget}%
5745         {\Hy@raisedlink{\hypertarget{#1}{}}}%
5746       {}%
5747     \else%
5748       \write\linenum@outR{\string\@lab}%
5749       \ifx\labelref@listR\empty%
5750         \xdef\label@refs{\zz@@@}%
5751       \else%
5752         \gl@p\labelref@listR\to\label@refs%
5753       \fi%
5754       \ifvmode%
5755         \advancelabel@refs%
5756       \fi%
5757     }%

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

5757   \protected@write\@auxout{%
5758     {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
5759     |{#1}}}%
5760   \ifdef{\hypertarget}%
5761     {\Hy@raisedlink{\hypertarget{#1}{}}}%
5762   {}%
5763 \fi%
5764 \ifXnote@%
5765   \protected@write\@auxout{%
5766     {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
5767     l@dparsedstartsub|\the\c@pstartR|{#1}}}%
5768   \ifdef{\hypertarget}%
5769     {\Hy@raisedlink{\hypertarget{#1}{}}}%
5770   {}%
5771 \else%

```

³³(jdb43@cam.ac.uk) via the ctt thread ‘ledmac cross referencing’, 25 August 2003.

```

5771 \write\linenum@out{\string\@lab}%
5772 \ifx\labelref@list\empty%
5773 \xdef\label@refs{\zz@@@}%
5774 \else%
5775 \gl@p\labelref@list\to\label@refs%
5776 \fi%
5777 \ifvmode%
5778 \advancelabel@refs%
5779 \fi%
5780 \protected@write\@auxout{%
5781 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}}%
5782 \ifdef{\hypertarget}%
5783 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5784 {}%
5785 \fi%
5786 }%
5787 \@esphack}%
5788 %
5789 %

```

`\advancelabel@refs` In cases where `\edlabel` is the first element in a paragraph, we have a problem with line counts, because line counts change only at the first horizontal box of the paragraph. Hence, we need to test `\edlabel` if it occurs at the start of a paragraph. To do so, we use `\ifvmode`. If the test is true, we must advance by one unit the amount of text we write into the `.aux` file. We do so using `\advancelabel@refs` command.

`\labelrefsparseline`

`\labelrefsparsesubline`

```

5790 \newcounter{line}%
5791 \newcounter{subline}%
5792 \newcommand{\advancelabel@refs}{%
5793 \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
5794 \stepcounter{line}%
5795 \ifsublines@%
5796 \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
5797 %
5798 \stepcounter{subline}{1}%
5799 \def\label@refs{\theline|\thesubline}%
5800 \else%
5801 \def\label@refs{\theline|0}%
5802 \fi%
5803 }
5804 \def\labelrefsparseline#1|#2{#1}
5805 \def\labelrefsparsesubline#1|#2{#2}
5806 %

```

`\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.

```

5806 \newcommand*{\l@dmake@labels}{}
5807 \def\l@dmake@labels#1|#2|#3|#4|#5{%
5808   \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
else
5809     \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
5810   \fi
5811   \global\providetoggle{label@#5@ledRcol}%False is the default value of
this toggle, which tell us if a label is linked to a right or a left side
5812   \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
{#1|#2|#3|#4|\relax}%
5813   \ignorespaces}
5814
5815 %

```

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.

```

5816 \AtBeginDocument{%
5817   \def\l@dmake@labels#1|#2|#3|#4|#5{%
5818   }
5819
5820 %

```

\@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`.

```

5821
5822 \newcommand*{\@lab}{%
5823   \ifledRcol
5824     \xright@appenditem{\linenumr@p{\line@numR}}|{%
5825       \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi}%
5826     \to\labelref@listR
5827   \else
5828     \xright@appenditem{\linenumr@p{\line@num}}|{%
5829       \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi}%
5830     \to\labelref@list
5831   \fi}
5832 %

```

`\applabel` `\applabel`, if called in `\edtext` will insert automatically both a start and an end label for the current edtext lines.

```
5833 \newcommand*{\applabel}[1]{%
5834   \if@edtext@secondarg%
5835   %
```

Label should not be already defined.

```
5836   \ifcsundef{the@label#1}{%
5837     \csdef{the@label#1}{\applabel}%
5838   }%
5839   {%
5840     \led@warn@DuplicateLabel{#1 (\applabel)}%
5841   }%
5842   %
```

Parse the `\edtext` line numbers.

```
5843   \expandafter\l@dp@rsefootspec\l@d@nums| %
5844   %
```

Use the \TeX standard hack for label.

```
5845   \@bsphack%
5846   %
```

And now, write the data in the auxiliary file.

```
5847   \ifl@Rcol%
5848     \protected@write\@auxout{%
5849       {\string\l@dmake@labelsR\space\l@dparsedstartpage|\
5850       \l@dparsedstartline|\l@dparsedstartsub|\the\c@pstartR|{#1:start}}%
5851       \ifdef{\hypertarget}%
5852         {\Hy@raisedlink{\hypertarget{#1:start}}{}}%
5853       }%
5854     \protected@write\@auxout{%
5855       {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline
5856       |\l@dparsedendsub|\the\c@pstartR|{#1:end}}%
5857     }%
5858   \else%
5859     \protected@write\@auxout{%
5860       {\string\l@dmake@labels\space\l@dparsedstartpage|\
5861       \l@dparsedstartline|\l@dparsedstartsub|\the\c@pstart|{#1:start}}%
5862       \ifdef{\hypertarget}%
5863         {\Hy@raisedlink{\hypertarget{#1:start}}{}}%
5864       }%
5865     \protected@write\@auxout{%
5866       {\string\l@dmake@labels\space\l@dparsedendpage|\l@dparsedendline
5867       |\l@dparsedendsub|\the\c@pstart|{#1:end}}%
5868     }%
5869   \fi%
5870   %
```

Use the \TeX standard hack for label.

```

5865 \esphack%
5866 %
Warning if \applabel is called outside of \edtext.
5867 \else%
5868 \led@warn@AppLabelOutSecondArgEdtext{#1}%
5869 \fi%
5870 %
End of \applabel
5871 }%
5872 %

```

\edlabelS \edlabelS and \edlabelE are just used to mark the beginning and the end of a passage.

\edlabelE

\edlabelSE

```

5873 \newcommand{\edlabelS}[1]{%
5874 \edlabel{#1:start}%
5875 }
5876 \newcommand{\edlabelE}[1]{%
5877 \edlabel{#1:end}%
5878 }
5879 \newcommand{\edlabelSE}[1]{%
5880 \edlabelS{#1}%
5881 \edlabelE{#1}%
5882 }
5883 %

```

\wrap@edcrossref \wrap@edcrossref is called around all reledmac crossref commands, except those which start with x. It adds the hyperlink.

```

5884 \newrobustcmd{\wrap@edcrossref}[2]{%
5885 \ifdef{\hyperlink}%
5886 {\hyperlink{#1}{#2}}%
5887 {#2}%
5888 }
5889 %

```

\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.

```

5890 \newcommand*{\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
5891 \newcommand*{\xpageref}[1]{\l@dgetref@num{1}{#1}}

```

5892

5893 %

\edlineref If the specified label exists, \lineref gives its line number.

\xlineref

```
5894 \newcommand*\edlineref}[1]{%
5895   \l@dref@undefined{#1}%
5896   \wrap@edcrossref{#1}{%
5897     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not been yet parsed on the .aux file
5898     \iftoggle{label@#1@ledRcol}%
5899       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
5900       {\linenumrep{\l@dgetref@num{2}{#1}}}%
5901     \xflagref{#1}%
5902   }%
5903 }%
5904 \newcommand*\xlineref}[1]{\l@dgetref@num{2}{#1}}%
5905
5906 %
```

\sublineref If the specified label exists, \sublineref gives its sub-line number.

\xsublineref

```
5907 \newcommand*\sublineref}[1]{%
5908   \l@dref@undefined{#1}%
5909   \wrap@edcrossref{#1}{%
5910     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not been yet parsed on the .aux file
5911     \iftoggle{label@#1@ledRcol}%
5912       {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
5913       {\sublinenumrep{\l@dgetref@num{3}{#1}}}%
5914     }%
5915   }%
5916 \newcommand*\xsublineref}[1]{\l@dgetref@num{3}{#1}}
5917
5918 %
```

\pstartref If the specified label exists, \pstartref gives its pstart number.

\xpstartref

```
5919 \newcommand*\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{4}{#1}}}%
5920 \newcommand*\xpstartref}[1]{\l@dgetref@num{4}{#1}}%
5921
5922 %
```

\xflagref \xflagref finds the side flag of any ref defined with \edlabel.

```
5923 \newcommand*\xflagref}[1]{\l@dgetref@num{5}{#1}}%
5924 %
```

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.

```

5925 \newcommand*\l@dref@undefined}[1]{%
5926   \expandafter\ifx\csname the@label#1\endcsname\relax
5927     \led@warn@RefUndefined{#1}%
5928   \fi}
5929
5930 %

```

\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.

```

5931 \newcommand*\l@dgetref@num}[2]{%
5932   \expandafter
5933   \ifx\csname the@label#2\endcsname \relax
5934     000%
5935   \else
5936     \expandafter\expandafter\expandafter
5937     \l@dlabel@parse\csname the@label#2\endcsname| #1%
5938   \fi}
5939
5940 %

```

\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`.)

```

5941 \newcommand*\l@dlabel@parse}{%
5942 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6{%
5943   \ifcase #6%
5944     \or #1%
5945     \or #2%
5946     \or #3%
5947     \or #4%
5948     \or #5%
5949   \fi}
5950 %

```

\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.

```

5951 \newcommand*\xxref}[2]{%
5952   {%
5953     \expandafter\ifx\csname the@label#1\endcsname \relax%
5954       \expandafter\let\csname the@@label#1\endcsname\zz@@@%
5955     \else%
5956       \expandafter\def\csname the@@label#1\endcsname{\l@getref@num
{1}{#1}|\l@getref@num{2}{#1}|\l@getref@num{3}{#1}}%
5957     \fi%
5958     \expandafter\ifx\csname the@label#2\endcsname \relax%
5959       \expandafter\let\csname the@@label#2\endcsname\zz@@@%
5960     \else%
5961       \expandafter\def\csname the@@label#2\endcsname{\l@getref@num
{1}{#2}|\l@getref@num{2}{#2}|\l@getref@num{3}{#2}}%
5962     \fi%
5963     \letcs{\@tempa}{the@@label#1}%
5964     \letcs{\@tempb}{the@@label#2}%
5965     \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@start{#1}}%
5966     \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@end{#2}}%
5967     \linenum{\@tempa|}%
5968     \@tempb}}%
5969
5970 %

```

\appref `\appref`, `\SEref`, `\apprefwithpage`, `\SErefwithpage` and `\SEonlypage` print cross-ref to some start / end lines defined by specific commands. It prints the lines as they should be printed in the apparatus (critical notes for not suffixed versions, endnotes for suffixed versions).

\SErefwithpage
\SErefwithpage

Here we define hooks similar to some those related to critical footnotes or endnotes. So, first declare the default value of the hooks for the pseudo-series. Also declare the internal toggle which are switch by `reledmac`.

```

5971 \def\Xtwolines@appref{%
5972 \def\Xtwolines@SEref{%
5973
5974 \def\Xmorethantwolines@appref{%
5975 \def\Xmorethantwolines@SEref{%
5976
5977 \def\Xlinerangeseparator@appref{\endashchar}%
5978 \def\Xlinerangeseparator@SEref{\endashchar}%
5979

```

```

5980 \def\Xsublinesep@appref{\fullstop}%
5981 \def\Xsublinesep@SEref{\fullstop}%
5982
5983 \def\Xpagelinesep@appref{\fullstop}%
5984 \def\Xpagelinesep@SEref{\fullstop}%
5985
5986
5987 \newtoggle{Xtwolinesbutnotmore@appref}%
5988 \newtoggle{Xtwolinesbutnotmore@SEref}%
5989
5990 \newtoggle{Xtwolinesonlyinsamepage@appref}%
5991
5992 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
5993
5994 \newtoggle{Xlineflag@appref}%
5995 \toggletrue{Xlineflag@appref}%Here exception
5996 \newtoggle{Xlineflag@SEref}%
5997 \toggletrue{Xlineflag@SEref}%Here exception
5998
5999 \def\Xendtwolines@apprefwithpage{}%
6000 \def\Xendtwolines@SErefwithpage{}%
6001
6002 \def\Xendmorethantwolines@apprefwithpage{}%
6003 \def\Xendmorethantwolines@SErefwithpage{}%
6004
6005 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
6006 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
6007 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
6008
6009 \def\Xendbeforepagenumber@apprefwithpage{p.}%
6010 \def\Xendbeforepagenumber@SErefwithpage{p.}%
6011 \def\Xendbeforepagenumber@SEonlypage{p.}%
6012
6013 \def\Xendafterpagenumber@apprefwithpage{} }%
6014 \def\Xendafterpagenumber@SErefwithpage{} }%
6015
6016
6017 \def\Xendlineprefixsingle@apprefwithpage{}%
6018 \def\Xendlineprefixsingle@SErefwithpage{}%
6019
6020 \def\Xendlineprefixmore@apprefwithpage{}%
6021 \def\Xendlineprefixmore@SErefwithpage{}%
6022
6023 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
6024 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
6025
6026 \def\Xendsublinesep@apprefwithpage{\fullstop}%
6027 \def\Xendsublinesep@SErefwithpage{\fullstop}%
6028
6029 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%

```

```

6030 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
6031
6032 \newtoggle{Xendlineflag@apprefwithpage}
6033 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
6034 \newtoggle{Xendlineflag@SErefwithpage}
6035 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
6036
6037 %

```

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.

```

6038
6039 \gdef\Xboxstartlinenum@appref{Opt}
6040 \gdef\Xboxstartlinenum@SEref{Opt}
6041
6042 \gdef\Xboxendlinenum@appref{Opt}
6043 \gdef\Xboxendlinenum@SEref{Opt}
6044
6045 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}
6046 \gdef\Xendboxstartlinenum@SErefwithpage{Opt}
6047
6048 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
6049 \gdef\Xendboxendlinenum@SErefwithpage{Opt}
6050
6051 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
6052 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
6053
6054 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
6055 \newtoggle{Xendpagenumberonlyfirstifsingle@SErefwithpage}
6056
6057 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
6058 \newtoggle{Xendpagenumberonlyfirstintwo@SErefwithpage}
6059
6060 \gdef\Xendsympagenum@apprefwithpage{}
6061 \gdef\Xendsympagenum@SErefwithpage{}
6062
6063 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
6064 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
6065
6066 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

6067 \newcommand\@apprefprefixsingle{}%
6068 \newcommand\@SErefprefixsingle{}%
6069
6070 \newcommand\@apprefprefixmore{}%
6071 \newcommand\@SErefprefixmore{}%

```



```

6072 \newcommand{\setapprefprefixsingle}[1]{%
6073   \gdef\@apprefprefixsingle{#1}%
6074 }
6075 \newcommand{\setSErefprefixsingle}[1]{%
6076   \gdef\@SErefprefixsingle{#1}%
6077 }
6078 }
6079 \newcommand{\setapprefprefixmore}[1]{%
6080   \gdef\@apprefprefixmore{#1}%
6081 }
6082 \newcommand{\setSErefprefixmore}[1]{%
6083   \gdef\@SErefprefixmore{#1}%
6084 }
6085 }
6086 %
6087 %

```

And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.

```

6088 \let\setSErefonlypageprefixsingle\XendbeforepagenumberSErefonlypage%
6089 \newcommand{\setSErefonlypageprefixmore}[1]{%
6090   \gdef\SErefonlypage@prefixmore{#1}%
6091 }%
6092 %

```

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.

```

6093 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
6094 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
6095
6096 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
6097 {#1}{#2}{appref}}
6098 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
6099 {#1}{#2}{SEref}}
6100 \newcommandx{\SErefonlypage}[2][1,usedefault]{\reformattedonlypage@
6101 {#1}{#2}{SEref}}
6102
6103 \newcommand{\reformatted@}[3]{%
6104   \def\do##1{%
6105     \setkeys[mac]{truefootnoteoption}{##1}%
6106   }%
6107   \notblank{#1}{\docsvlist{#1}}{}%
6108   \xdef\@currentseries{#3}%
6109   \ifcempty{#3prefixmore}%
6110     {\@apprefprefixsingle}%
6111     {%

```

```

6111 \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
6112 {\csuse{@#3prefixsingle}}%
6113 {\csuse{@#3prefixmore}}%
6114 }%
6115 \ifboolexpr{%
6116 test{\ifcsundef{the@label#2:start}}}%
6117 or test{\ifcsundef{the@label#2:end}}}%
6118 }%
6119 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6120 {%
6121 \def\@this@crossref@start{#2:start}%
6122 \def\@this@crossref@end{#2:end}%
6123 \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}}|}%
6124 \undef\@this@crossref@end%
6125 \undef\@this@crossref@start%
6126 }%
6127 \def\do##1{%
6128 \setkeys[mac]{falsefootnoteoption}{##1}%
6129 }%
6130 \notblank{#1}{\docsvlist{#1}}}%
6131 }%
6132
6133 \newcommand{\reformattedwithpage@}[3]{%
6134 \def\do##1{%
6135 \setkeys[mac]{truefootnoteoption}{##1}%
6136 }%
6137 \notblank{#1}{\docsvlist{#1}}}%
6138 \xdef\@currentseries{#3withpage}%
6139 \ifboolexpr{%
6140 test{\ifcsundef{the@label#2:start}}}%
6141 or test{\ifcsundef{the@label#2:end}}}%
6142 }%
6143 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6144 {%
6145 \def\@this@crossref@start{#2:start}%
6146 \def\@this@crossref@end{#2:end}%
6147 \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}}|}%
6148 \undef\@this@crossref@end%
6149 \undef\@this@crossref@start%
6150 }%
6151 \def\do##1{%
6152 \setkeys[mac]{falsefootnoteoption}{##1}%
6153 }%
6154 \notblank{#1}{\docsvlist{#1}}}%
6155 }%
6156

```

```

6157 \newcommand{\reformattedonlypage@}[3]{%
6158   \def\do##1{%
6159     \setkeys[mac]{truefootnoteoption}{##1}%
6160   }%
6161   \notblank{#1}{\docsvlist{#1}}{}%
6162   \xdef\@currentseries{#3onlypage}%
6163   \ifboolexpr{%
6164     test{\ifcsundef{the@label#2:start}}%
6165     or test{\ifcsundef{the@label#2:end}}%
6166   }%
6167   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6168   {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
6169     {%
6170       \printnpnum{%
6171         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6172       }%
6173     }%
6174   {%
6175     \ifcsvoid{#3onlypage@prefixmore}%
6176     {}%
6177     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
6178     \ifdefined\linangesep@%
6179     \printnpnum{%
6180       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6181       \linangesep@%
6182       \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
6183     }%
6184     \else%
6185     \printnpnum{%
6186       \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6187       \csuse{Xendlinangeseparator@}\@currentseries%
6188       \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
6189     }%
6190     \fi%
6191   }%
6192 }%
6193 \def\do##1{%
6194   \setkeys[mac]{falsefootnoteoption}{##1}%
6195 }%
6196 \notblank{#1}{\docsvlist{#1}}{}%
6197 }%
6198 %

```

\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`.

```
6199 \newcommand*{\edmakelabel}[2]{\expandafter\edef\csname the@label#1\
endcsname{#2}}
6200
6201 %
```

(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. 128 and V.9 p. 95), 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`.

```
6202 \pretocmd{\XR@test}{%
6203   {\XR@test@mac+++#1#2#3#4+++}%
6204   {}%
6205   {}%
6206   %
```

\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`.

```
6207 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
6208 %
```

\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`.

```
6209 \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
6210   \ifx#1\l@dmake@labels%
6211     \l@dmake@labels#2%
6212   \else
6213     \ifx#1\l@dmake@labelsR%
6214       \l@dmake@labelsR #2%
6215     \fi%
6216   \fi%
6217 }%
6218 %
```

XXIV Side notes

Regular `\marginpar` do not work inside numbered text — they do not produce any note but do put an extra unnumbered blank line into the text.

`\@xympar` Changing `\@xympar` a little at least ensures that `\marginpar` in numbered text do not disturb the flow.

```

6219 \pretocmd{\@xympar}%
6220   {\ifnumberedpar@
6221     \led@warn@NoMarginpars
6222     \@esphack
6223   \else}%
6224   {}%
6225   {}%
6226
6227 \apptocmd{\@xympar}%
6228   {\fi}%
6229   {}
6230   {}
6231
6232 %

```

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

```

6233 \newcount\sidenote@margin
6234 \newcommand*{\sidenotemargin}[1]{\{%
6235   \l@dgetsidenote@margin{#1}%
6236   \ifnum\@l@tempcntb>\m@ne
6237     \ifledRcol
6238       \global\sidenote@marginR=\@l@tempcntb
6239     \else
6240       \global\sidenote@margin=\@l@tempcntb
6241     \fi
6242   \fi}}
6243 \newcommand*{\l@dgetsidenote@margin}[1]{%
6244   \def\@tempa{#1}\def\@tempb{left}%

```

```

6245 \ifx\@tempa\@tempb
6246 \l@ldtempcntb \z@
6247 \else
6248 \def\@tempb{right}%
6249 \ifx\@tempa\@tempb
6250 \l@ldtempcntb \@ne
6251 \else
6252 \def\@tempb{outer}%
6253 \ifx\@tempa\@tempb
6254 \l@ldtempcntb \tw@
6255 \else
6256 \def\@tempb{inner}%
6257 \ifx\@tempa\@tempb
6258 \l@ldtempcntb \thr@@
6259 \else
6260 \led@warn@BadSidenotemargin
6261 \l@ldtempcntb \m@ne
6262 \fi
6263 \fi
6264 \fi
6265 \fi}
6266 \sidenotemargin{right}
6267
6268 %

```

`\l@ldlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
6269 \newbox\l@ldlp@rbox
6270 \newbox\l@drp@rbox
6271
6272 %

```

`\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
6273 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
6274 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
6275 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup
6276 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
6277 \newcommand*{\ledlsnotefontsetup}{\raggedleft\footnotesize}
6278 \newcommand*{\ledrsnotefontsetup}{\raggedright\footnotesize}
6279
6280 %

```

`\ledleftnote` `\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.

```

6281 \newcommand*{\ledleftnote}[1]{\edtext{}{\l@dlsnote{#1}}}
6282 \newcommand*{\ledrightnote}[1]{\edtext{}{\l@drsnote{#1}}}
6283 \newcommand*{\ledsidenote}[1]{\edtext{}{\l@dcsnote{#1}}}%
6284 \newcommand*{\ledinnernote}[1]{\edtext{}{\l@disnote{#1}}}%
6285 \newcommand*{\ledouternote}[1]{\edtext{}{\l@dosnote{#1}}}%
6286 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.

```

6287 \newif\ifrightrightnoteup
6288 \rightnoteuptrue
6289 \l@disnote
6290 \newcommand*{\l@dlsnote}[1]{%
6291   \begingroup%
6292   \newcommand{\content}{#1}%
6293   \ifnumberedpar@
6294     \ifledRcol%
6295       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}{%
6296         \to\inserts@listR
6297         \global\advance\insert@countR \@ne%
6298       }%
6299     \else%
6300       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}{%
6301         \to\inserts@list
6302         \global\advance\insert@count \@ne%
6303       }%
6304     \fi
6305   \fi%
6306   \ignorespaces%
6307   \endgroup%
6308 }%
6309 \newcommand*{\l@drsnote}[1]{%
6310   \begingroup%
6311   \newcommand{\content}{#1}%
6312   \ifnumberedpar@
6313     \ifledRcol%
6314       \xright@appenditem{\noexpand\l@drsnote{\expandonce\content}}{%
6315         \to\inserts@listR
6316         \global\advance\insert@countR \@ne%
6317       }%
6318     \else%
6319       \xright@appenditem{\noexpand\l@drsnote{\expandonce\content}}{%
6320         \to\inserts@list
6321         \global\advance\insert@count \@ne%
6322       }%
6323     \fi
6324   \fi\ignorespaces%
6325   \endgroup%
6326 }%
6327 \newcommand*{\l@dcsnote}[1]{%
6328   \begingroup%

```

```

6327 \newcommand{\content}{#1}%
6328 \ifnumberedpar@
6329 \ifledRcol%
6330 \xright@appenditem{\noexpand\vl@dcnote{\expandonce\content}}%
6331 \to\inserts@listR
6332 \global\advance\insert@countR \@ne%
6333 \else%
6334 \xright@appenditem{\noexpand\vl@dcnote{\expandonce\content}}%
6335 \to\inserts@list
6336 \global\advance\insert@count \@ne%
6337 \fi
6338 \fi\ignorespaces%
6339 \endgroup%
6340 }%
6341
6342 \newcommand*{\l@disnote}[1]{%
6343 \begingroup%
6344 \newcommand{\content}{#1}%
6345 \ifnumberedpar@%
6346 \ifledRcol%
6347 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}%
6348 \to\inserts@listR%
6349 \global\advance\insert@countR \@ne%
6350 \else%
6351 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}%
6352 \to\inserts@list%
6353 \global\advance\insert@count \@ne%
6354 \fi%
6355 \fi\ignorespaces%
6356 \endgroup%
6357 }%
6358
6359 \newcommand*{\l@dosnote}[1]{%
6360 \begingroup%
6361 \newcommand{\content}{#1}%
6362 \ifnumberedpar@%
6363 \ifledRcol%
6364 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}%
6365 \to\inserts@listR%
6366 \global\advance\insert@countR \@ne%
6367 \else%
6368 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}%
6369 \to\inserts@list%
6370 \global\advance\insert@count \@ne%
6371 \fi%
6372 \fi\ignorespaces%
6373 \endgroup%
6374 }%
6375
6376 %

```


`\vl@dlsnote` Put the left/right text into boxes, but just save the moveable text. `\l@dcsnotetext`, `\vl@drsnote` `\l@dcsnotetext@l` and `\l@dcsnotetext@r` are `etoolbox`'s lists which will store the content of side notes. We store the content in lists, because we need to loop later on them, in case many sidenote co-exist for the same line. That is there some special test to do, in order to:

- Store the content of `\ledsidenote` to `\l@dcsnotetext` in any cases.
- Store the content of `\rightsidenote` to:
 - `\l@dcsnotetext` if `\ledsidenote` is to be put on right.
 - `\l@dcsnotetext@r` if `\ledsidenote` is to be put on left.
- Store the content of `\leftsidenote` to:
 - `\l@dcsnotetext` if `\ledsidenote` is to be put on left.
 - `\l@dcsnotetext@l` if `\ledsidenote` is to be put on right.

`\vl@disnote` and `\vl@dosnote` just call `\vl@dlsnote` or `\vl@drsnote`, depending of the page.

```

6377 \newcommand*{\vl@dlsnote}[1]{%
6378   \ifledRcol%
6379     \@l@tempcntb=\sidenote@marginR%
6380     \ifnum\@l@tempcntb>\@ne%
6381       \advance\@l@tempcntb by\page@numR%
6382     \fi%
6383   \else%
6384     \@l@tempcntb=\sidenote@margin%
6385     \ifnum\@l@tempcntb>\@ne%
6386       \advance\@l@tempcntb by\page@num%
6387     \fi%
6388   \fi%
6389   \ifodd\@l@tempcntb%
6390     \listgadd{\l@dcsnotetext@l}{#1}%
6391   \else%
6392     \listgadd{\l@dcsnotetext}{#1}%
6393   \fi
6394 }
6395 \newcommand*{\vl@drsnote}[1]{%
6396   \ifledRcol%
6397     \@l@tempcntb=\sidenote@marginR%
6398     \ifnum\@l@tempcntb>\@ne%
6399       \advance\@l@tempcntb by\page@numR%
6400     \fi%
6401   \else%
6402     \@l@tempcntb=\sidenote@margin%
6403     \ifnum\@l@tempcntb>\@ne%
6404       \advance\@l@tempcntb by\page@num%
6405     \fi%

```

```

6406 \fi%
6407 \ifodd\l@dttempcntb%
6408 \listgadd{\l@dcsnotetext}{#1}%
6409 \else%
6410 \listgadd{\l@dcsnotetext@r}{#1}%
6411 \fi%
6412 }
6413 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
6414
6415 \newcommand{\vl@disnote}[1]{%
6416 \ifledRcol%
6417 \@tempcnta=\page@numR%
6418 \else%
6419 \@tempcnta=\page@num%
6420 \fi%
6421 \ifodd\@tempcnta% ODD => right page => inner side = left side
6422 \vl@dlsnote{#1}%
6423 \else%
6424 \vl@drsnote{#1}%
6425 \fi%
6426 }%
6427
6428 \newcommand{\vl@dosnote}[1]{%
6429 \ifledRcol%
6430 \@tempcnta=\page@numR%
6431 \else%
6432 \@tempcnta=\page@num%
6433 \fi%
6434 \ifodd\@tempcnta% ODD => right page => outer side = right side
6435 \vl@drsnote{#1}%
6436 \else%
6437 \vl@dlsnote{#1}%
6438 \fi%
6439 }%
6440
6441 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box. Similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

6442 \newcommand*{\setl@dlp@rbox}[1]{%
6443 \begingroup%
6444 \parindent\z@\hspace=\ledlsnotewidth%
6445 \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
6446 \global\setbox\l@dlp@rbox%
6447 \ifleftnoteup%
6448 =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note

```

```

the {} to keep setting local.
6449   \else%
6450     =\vbox to 0.70\baselineskip{{\ledrsnotefontsetup\strut#1\vss}}%
6451   \fi%
6452 \endgroup%
6453 }
6454
6455 \newcommand*{\setl@drp@rbox}[1]{%
6456   \begingroup%
6457   \parindent\z@ \hsize=\ledrsnotewidth%
6458   \ledrsnotefontsetup% We kept it outside of the vbox, because can affect
the ragging
6459   \global\setbox\l@drp@rbox%
6460   \ifrightnoteup%
6461     =\vbox to\z@{{\ledrsnotefontsetup\vss#1}}% We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
6462   \else%
6463     =\vbox to0.7\baselineskip{{\ledrsnotefontsetup\strut#1\vss}}%
6464   \fi%
6465   \endgroup%
6466 }%
6467 \newif\ifleftnoteup
6468 \leftnoteuptrue
6469 %

```

\@sidenotesep This macro is used to separate sidenotes of the same line.

```

6470 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
6471 \newcommand{\@sidenotesep}{, }
6472 %

```

\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.

```

6473 \newcommand*{\affixside@note}{%
6474   \prepare@edindex@fornote{\the\page@num|\the\line@num|\the\subline@num|\
the\page@num|\the\line@num|\the\subline@num|}%
6475   \def\sidenotecontent@{}%
6476   \numgdef{\itemcount@}{0}%
6477   \def\do##1{%
6478     \ifnumequal{\itemcount@}{0}%
6479     {%
6480       \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
6481     {\appto\sidenotecontent@{\@sidenotesep ##1}}%
6482   }%

```

```

6483         \numgdef{\itemcount@}{\itemcount@+\@ne}%
6484     }%
6485     \dolistloop{\l@dcstotetext}%
6486     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
6487 %

```

And we do the same for left and right notes (not movable).

```

6488 \gdef\@templ@d{%
6489 \gdef\@templ@n{\l@dcstotetext\l@dcstotetext@l\l@dcstotetext@r}%
6490 \ifx\@templ@d\@templ@n \else%
6491 \if@twocolumn%
6492 \if@firstcolumn%
6493 \setl@dlp@rbox{##1}{\sidenotecontent@}%
6494 \else%
6495 \setl@drp@rbox{\sidenotecontent@}%
6496 \fi%
6497 \else%
6498 \l@dtempcntb=\sidenote@margin%
6499 \ifnum\l@dtempcntb>\@ne%
6500 \advance\l@dtempcntb by\page@num%
6501 \fi%
6502 \ifodd\l@dtempcntb%
6503 \setl@drp@rbox{\sidenotecontent@}%
6504 \gdef\sidenotecontent@{%
6505 \numgdef{\itemcount@}{0}%
6506 \dolistloop{\l@dcstotetext@l}%
6507 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
6508 \setl@dlp@rbox{\sidenotecontent@}%
6509 \else%
6510 \setl@dlp@rbox{\sidenotecontent@}%
6511 \gdef\sidenotecontent@{%
6512 \numgdef{\itemcount@}{0}%
6513 \dolistloop{\l@dcstotetext@r}%
6514 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
6515 \setl@drp@rbox{\sidenotecontent@}%
6516 \fi%
6517 \fi%
6518 \fi%
6519 \advance\@edindex@fornote@m@ne%
6520 }
6521 %

```

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 `\@iiiminpage` and `\endminipage`.
`\l@dfeetendmini` They can be extended to handle other things if necessary.

```

6522 \ifnoledgroup@else%
6523 \newcommand*{\l@dfeetbeginmini}{\@ledgrouptrue\l@dedbeginmini\
l@dfambeginmini}
6524 \newcommand*{\l@dfeetendmini}{%
6525   \IfStrEq{critical-familiar}{\@mpfnpos}%
6526   {\l@dedendmini\l@dfamendmini}%
6527   {%
6528     \IfStrEq{familiar-critical}{\@mpfnpos}%
6529     {\l@dfamendmini\l@dedendmini}%
6530     {\do@feet@custom@order{mp@}{\@mpfnpos}}%
6531   }%
6532 }%
6533 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.
`\l@dedendmini`
`\mp@append@Xnotes`

```

6534 \newcommand*{\l@dedbeginmini}{%
6535   \unless\ifnocritical@%
6536   \def\do##1{%
6537     \csletcs{v##1footnote}{mpv##1footnote}%
6538   }%
6539   \dolistloop{\@series}%
6540   \fi%
6541 }
6542 \newcommand*{\l@dedendmini}{%
6543   \unless\ifnocritical@%
6544   \ifl@dpairing%
6545     \ifledRcol%
6546       \flush@notesR%
6547     \else%
6548       \flush@notes%
6549     \fi%
6550   \fi
6551   \def\do##1{%
6552     \mp@append@Xnotes{##1}%
6553   }%
6554   \dolistloop{\@series}%
6555   \fi%
6556 }%
6557 \newcommand{\mp@append@Xnotes}[1]{%
6558   \ifvoid\csuse{mp#1footins}\else%
6559     \ifl@dpairing%
6560       \ifparledgroup%
6561         \ifledRcol%
6562           \dimgdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
6563         \else%

```

```

6564 \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\
skip\@nameuse{mp#1footins}}%
6565 \fi%
6566 \fi%
6567 \fi%
6568 \ifcsstring{series@display#1}{paragraph}{}{%
6569 \setbox\@nameuse{mp#1footins}=\vbox{%
6570 \csuse{Xnotefontsize@#1}%
6571 \ifcsdef{Xhsize\csuse{series@display#1}@#1}{%
6572 \hsize\csuse{Xhsize\csuse{series@display#1}@#1}%
6573 }{}%
6574 \noindent\csuse{Xtxtbeforenotes@#1}%
6575 \unvbox\@nameuse{mp#1footins}%
6576 \@parboxrestore%
6577 }%
6578 }%
6579 \csuse{mp#1footgroup}{#1}%
6580 \fi%
6581 }%
6582 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
6583 \newcommand*\l@dfambeginmini{%
6584 \unless\ifnofamiliar%
6585 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
6586 \dolistloop{\@series}%
6587 \fi%
6588 }%
6589
6590 \newcommand*\l@dfamendmini{%
6591 \unless\ifnofamiliar%
6592 \def\do##1{%
6593 \mp@append@notesX{##1}%
6594 }%
6595 \dolistloop{\@series}%
6596 \fi%
6597 }%
6598 \newcommand{\mp@append@notesX}[1]{%
6599 \ifvoid\csuse{mpfootins#1}\else%
6600 \csuse{mpfootgroup#1}{#1}%
6601 \fi%
6602 }%
6603 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

6604 \patchcmd%
6605 {\@iiiminipage}%
6606 {\let\@footnotetext\@mpfootnotetext}%

```

```

6607 {\let\@footnotetext\@mpfootnotetext\l@dfetbeginmini}%
6608 {}%
6609 {\led@error@fail@patch@iiiminipage}%
6610 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

6611 \patchcmd%
6612 {\endminipage}%
6613 {\footnoterule}%
6614 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
6615 {}%
6616 {\led@error@fail@patch@endminipage}
6617
6618 \patchcmd%
6619 {\endminipage}%
6620 {\@minipagefalse}%
6621 {\l@dfetendmini\@minipagefalse}%
6622 {}%
6623 {\led@error@fail@patch@endminipage}
6624
6625 %

```

`\l@dunboxmpfoot` `\l@dunboxmpfoot` insert normal footnotes for `ledgroup`.
`edgroup@beforenormalnotes`

```

6626 \newcommand*{\l@dunboxmpfoot}{%
6627   \vskip\skip\@mpfootins
6628   \normalcolor
6629   \footnoterule
6630   \l@advance@parledgroup@beforenormalnotes
6631   \unvbox\@mpfootins%
6632 }
6633 %

```

When using parallel `ledgroup`, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

6634 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
6635   \ifparledgroup
6636     \ifl@pairing
6637       \ifledRcol
6638         \dimdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
6639       \else
6640         \dimdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
6641       \fi
6642     \fi
6643   \fi
6644 }
6645 %

```

`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.

```

6646 \newenvironment{ledgroup}{%
6647   \resetprevpage@num%
6648   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
6649   \let\@footnotetext\@mpfootnotetext
6650   \l@dfetbeginmini%
6651 }{%
6652   \par
6653   \unskip
6654   \ifvoid\@mpfootins\else
6655     \l@dunboxmpfoot
6656   \fi
6657   \l@dfetendmini%
6658   \@ledgroupfalse%
6659 }
6660
6661
6662
6663 %

```

`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.

```

6664 \newenvironment{ledgroupsize}[2][1]{%
6665 %

```

Set the various text measures.

```

6666   \hsize #2\relax
6667 %

```

Initialize fills for centering.

```

6668   \let\ledllfill\hfil
6669   \let\ledrlfill\hfil
6670   \def\@tempa{#1}\def\@tempb{1}%
6671 %

```

Left adjusted numbered lines

```

6672   \ifx\@tempa\@tempb
6673     \let\ledllfill\relax
6674   \else
6675     \def\@tempb{r}%
6676     \ifx\@tempa\@tempb
6677 %

```

Right adjusted numbered lines


```

6678 \let\ledrlfill\relax
6679 \fi
6680 \fi
6681 %

```

Set up the footnoting.

```

6682 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
6683 \let\@footnotetext\@mpfootnotetext
6684 \l@dfetbeginmini%
6685 }{f%
6686 \par
6687 \unskip
6688 \ifvoid\@mpfootins\else
6689 \l@dunboxmpfoot
6690 \fi
6691 \l@dfeetendmini%
6692 }
6693
6694 %

```

Close the \ifnoledgroup@\else.

```

6695 \fi%
6696 %

```

`\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`.

```

6697 \newif\ifledgroupnotesL@
6698 \newif\ifledgroupnotesR@
6699 %

```

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`.

```

6700 \AtBeginDocument{%
6701 \unless\ifl@imakeidx%
6702 \@ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
6703 %
6704 \fi%
6705 \unless\ifl@indextools%
6706 \@ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}%
6707 \fi%

```

```

6707 \unless\ifl@footmisc%
6708 \@ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
6709 %
6710 \fi%
6711 }
6712 %

```

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
6712 \newcommand{\pagelinesep}{-}
6713 \newcommand{\edindexlab}{$&}
6714 \newcounter{labidx}
6715 \setcounter{labidx}{0}
6716
6717 %

```

`\doedindexlabel` This macro sets an `\edlabel`.

```

6718 \newcommand{\doedindexlabel}{%
6719 \stepcounter{labidx}%
6720 \edlabel{\edindexlab\thelabidx}%
6721 }
6722
6723 %

```

`\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`.

```

6724 \newcounter{pageline}%
6725 \renewcommand{\thepageline}{%
6726 \thepage%
6727 \pagelinesep%
6728 \xlineref{\edindexlab\thelabidx}%
6729 }
6730 %

```

`\thestartpageline` These macros make up the page/line start/end number when the `\edindex` command is called in critical notes.

```

\theendpageline
6731 \newcommand{\thestartpageline}{%
6732 \l@dparsedstartpage%
6733 \pagelinesep%
6734 \l@dparsedstartline%
6735 }
6736 \newcommand{\theendpageline}{%

```

```

6737 \l@dparsedendpage%
6738 \pagelinesep%
6739 \l@dparsedendline%
6740 }
6741 %

```

XXVI.3 Code specific to \edindex in critical footnotes

\@edindex@fornote@ This counter is increased at the beginning of every notes (footnote and and sidenotes), and decreased at the end of every notes. If its value is greater than 0, that means we are inside a note.

```

6742 \newcount\@edindex@fornote@
6743 %

```

\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 |.

```

6744 \newcommand{\prepare@edindex@fornote}[1]{%
6745   \l@dp@rsefootspec#1|}%
6746   \advance\@edindex@fornote@\@ne%
6747 }
6748 %

```

\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.

```

6749 \newcommand{\get@edindex@ledinnote@command}{%
6750   \ifxindy@%
6751     \gdef\@ledinnote@command{%
6752       ledinnote\thelabidx%
6753     }%
6754     \ifxindyhyperref@%
6755       \immediate\write\eledmac@xindy@out{%
6756         (define-attributes ("ledinnote\thelabidx"))^^J
6757         \space\space(markup-locref^^J
6758         \eledmacmarkuplocrefdepth^^J
6759         :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
6760       }{"^^J
6761         :close "}"^^J
6762         :attr "ledinnote\thelabidx"^^J
6763       )
6764     }%
6765   \else%
6766     \immediate\write\eledmac@xindy@out{%

```

```

6766      (define-attributes ("ledinnote\thelabidx"))^^J
6767      \space\space(markup-locref^^J
6768      \eledmacmarkuplocdepth^^J
6769      :open "\string\ledinnote{\@index@command}"^^J
6770      :close "}"^^J
6771      :attr "ledinnote\thelabidx"^^J
6772      )
6773      }%
6774      \fi%
6775      %

```

If we do not use xindy option, \@ledinnote@command will produce something like ledinnote{formattingcommand}.

```

6776      \else%
6777      \gdef\@ledinnote@command{%
6778      ledinnote[\edindexlab\thelabidx]{\@index@command}%
6779      }%
6780      \fi%
6781      }
6782      %

```

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 |.

```

6783 \def\get@index@command#1|#2+{%
6784   \gdef\@index@txt{#1}%
6785   \gdef\@index@command{#2}%
6786   \xdef\@index@parenthesis{%
6787     \IfBeginWith{\@index@command}{(}{%
6788       \StrGobbleLeft{\@index@command}{1}{\@index@command}%
6789       \global\let\@index@command\@index@command@%
6790       \xdef\@index@parenthesis{(%
6791     )}%
6792     \IfBeginWith{\@index@command}{)}{%
6793       \StrGobbleLeft{\@index@command}{1}{\@index@command}%
6794       \global\let\@index@command\@index@command@%
6795       \xdef\@index@parenthesis{)%
6796     }}%
6797   }
6798   %

```

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`

```

6799 \newcommandx{\ledinnote}[3][1,usedefault]{%
6800   \ifboolexpr{%
6801     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
6802     or%
6803     bool {xindyhyperref@}%
6804   }%
6805   {%
6806     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
6807   }%
6808   {%
6809     \csuse{#2}{\ledinnotemark{#3}}%
6810   }%
6811 }%
6812 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage
6813   {#2}}}}%
6814 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
6815 %

```

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
6815 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
6816   index name, #2 = the text
6817   \ifl@imakeidx%
6818     \ifnum\@edindex@fornote@>\z@%
6819     \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6820     \get@edindex@ledinnote@command%
6821     \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
6822     }\the startpageline}%
6823     \expandafter\imki@wrindexentry{#1}{\@index@txt)\@ledinnote@command
6824     }\the endpageline}%
6825   \else%
6826     \get@edindex@hyperref{#2}%
6827     \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}\the pageline}%
6828   \fi%
6829   \else%
6830     \ifnum\@edindex@fornote@>\z@%
6831     \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6832     \get@edindex@ledinnote@command%

```

```

6830 \expandafter\protected@write\@indexfile{}\%
6831 {\string\indexentry{\@index@txt|(\@ledinnote@command)}{\thestartpageline}
6832 }%
6833 \expandafter\protected@write\@indexfile{}\%
6834 {\string\indexentry{\@index@txt|)\@ledinnote@command)}{\theendpageline}
6835 }%
6836 \else%
6837 \protected@write\@indexfile{}\%
6838 {\string\indexentry{#2}{\thepageline}
6839 }%
6840 \fi%
6841 \fi%
6842 \endgroup
6843 \@esphack%
6844 }
6845 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

6846 \pretocmd{\makeindex}{\%
6847 \def\edindex{\%
6848 \ifboolexpr{bool{numbering} or bool{numberingR}}{\%
6849 \@bsphack%
6850 \doedindexlabel%
6851 \begingroup%
6852 \@sanitize%
6853 \@wredindex%
6854 }%
6855 {\%
6856 \led@warn@edinde@outsidenumbering%
6857 \index%
6858 }%
6859 }%
6860 }%
6861 {}%
6862 {\led@error@fail@patch@makeindex}%
6863 \newcommand{\edindex}[1]{\@bsphack\@esphack}
6864 %

```

XXVI.7 Hyperlink

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```

6865 \newcommand{\hyperlinkformat}[3]{\%
6866 \ifstrempy{#1}%
6867 {\hyperlink{#2}{#3}}%
6868 {\csuse{#1}{\hyperlink{#2}{#3}}%
6869 }}
6870 %

```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```
6871 \newcommand{\hyperlinkR}[2]{%
6872   \hyperlink{#1}{#2\@Rlineflag}%
6873 }%
6874
6875 %
```

`\hyperlinkformatR` `\hyperlinkformatR` command is to be used to create a internal hyperlink, a format and a `\@Rlineflag`, when indexing.

```
6876 \newcommand{\hyperlinkformatR}[3]{%
6877   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
6878 }%
6879
6880 %
```

`\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`).

```
6881 \newcommand{\get@edindex@hyperref}[1]{%
6882   %
```

We have to disable temporary spaces to work through a `xstring` bug (or feature?)

```
6883 \edef\temp@{%
6884   \catcode`\ =9 %space need for catcode
6885   \detokenize{#1}%For active character in unicode
6886   \catcode`\ =10 % space need for catcode
6887 }%
6888 %
```

Now, we define `\@edindex@hyperref` if the `hyperindex` of `hyperref` is enabled.

```
6889 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
6890   \IfSubStr{\temp@}{|}%
6891   {\get@index@command#1+%
6892     \ifledRcol%
6893       \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6894         hyperlinkformatR{\@index@command}%
6895         {\edindexlab\thelabidx}}%
6896     \else%
6897       \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6898         hyperlinkformat{\@index@command}%
6899         {\edindexlab\thelabidx}}%
6900     \fi%
6901   }%
6902   {\get@index@command#1|+%
6903     \ifledRcol%
6904       \gdef\@edindex@hyperref{\hyperlinkR{\edindexlab\thelabidx}}%
6905     \else%
```

```

6906     \gdef\@edindex@hyperref{|hyperlink{\edindexlab\thelabidx}}%
6907     \fi%
6908   }%
6909 }%
6910 %

6911 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
6912 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
6913 % \begin{macrocode}
6914 {\ifxindyhyperref%
6915   \IfSubStr{\temp@}{|}%
6916   {\get@index@command#1+}%
6917   {\get@index@command#1|+}%
6918   \gdef\@edindex@hyperref{|eledmac\thelabidx}%
6919 %

```

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`.

```

6920   \IfStrEq{\@index@parenthesis}{(}%
6921   {%
6922     \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
6923     \gdef\@edindex@hyperref{| (eledmac\thelabidx}%
6924   }%
6925   {}%
6926 %

```

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`.

```

6927   \IfStrEq{\@index@parenthesis}{)}%
6928   {%
6929     \xdef\@edindex@hyperref{|)eledmac\csuse{xindyparenthesis@\@index@txt}}%
6930     \global\csundef{xindyparenthesis@\@index@txt}%
6931   }%
6932 %

```

Write in the `.xdy` file the attributes of the location.

```

6933   {%
6934   \immediate\write\eledmac@xindy@out{%
6935     (define-attributes ("eledmac\thelabidx"))^^J
6936     \space\space(markup-locoref^^J
6937     \eledmacmarkuplocorefdepth^^J
6938     :open "\string\hyperlink%
6939           \ifledRcol R\fi%
6940           {\edindexlab\thelabidx}%
6941           {\ifdefempty{\@index@command}}%

```



```

6942         {}%
6943         {@backslashchar\@index@command}%
6944         {"^^J
6945         :close "}"^^J
6946         :attr "eledmac\thelabidx"^^J
6947     )
6948 }%
6949 }%
6950 %

```

And now, in any other case.

```

6951 \else%
6952 \gdef\@index@txt{#1}%
6953 \gdef\@edindex@hyperref{}%
6954 \fi%
6955 }%
6956 }
6957 %

```

XXVI.8 ‘innote’ and ‘notenumber’ option of indextols package

\led@set@index@fornote The `\led@set@index@fornote` is called when a familiar footnote is inserted — and not when it is read — and changes the `\index` command depending of the option of the indextools package. Its only argument is the note series.

```

6958 \newcommand{\led@set@index@fornote}[1]{%
6959 \ifbool{indtl@innote}%
6960 {\let\index\nindex}%
6961 {}%
6962 \ifbool{indtl@notenumber}%
6963 {%
6964 \renewcommand{\index}[2][\indtl@jobname]{%
6965 \orig@@index[##1]{%
6966 ##2|innotenumber{\this@footnoteX@reading}%
6967 }%
6968 }%
6969 }%
6970 {}%
6971 }%
6972 %

```

\led@reinit@index@fornote The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

6973 \newcommand{\led@reinit@index@fornote}{%
6974 \ifbool{indtl@innote}%
6975 {\let\index\orig@@index}%
6976 {}%
6977 \ifbool{indtl@notenumber}%

```

```

6978 {\let\index\orig@index}%
6979 {}%
6980 }%
6981 %

```

XXVII Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

6982 \AtBeginDocument{%
6983   \ifpackageloaded{glossaries}{%
6984     %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

6985   \renewcommand{\do}[1]{%
6986     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
6987       \doedindexlabel%
6988       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
6989     }%
6990     \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,3,usedefault]{%
6991       \doedindexlabel%
6992       \csname#1\endcsname*[counter=pageline,##1]{##2}[##3]%
6993     }%
6994   }%
6995   \docsvlist{%
6996     gls,%
6997     Gls,%
6998     GLS,%
6999     glspl,%
7000     Glspl,%
7001     GLSpl,%
7002     glstext,%
7003     Glstext,%
7004     GLStext,%
7005     Glsfirst,%
7006     GLSfirst,%
7007     glsplural,%
7008     Glsplural,%
7009     GLSplural,%
7010     glsfirstplural,%
7011     Glsfirstplural,%
7012     GLSfirstplural,%
7013     glsname,%
7014     Glsname,%
7015     GLSname,%
7016     glssymbol,%

```

```

7017     Glssymbol,%
7018     GLSsymbol,%
7019     glsdsc,%
7020     GLsdsc,%
7021     GLSdsc,%
7022     glsuseri,%
7023     GLsuseri,%
7024     GLSuseri,%
7025     glsuserii,%
7026     GLsuserii,%
7027     GLSuserii,%
7028     glsuseriii,%
7029     GLsuseriii,%
7030     GLSuseriii,%
7031     glsuseriv,%
7032     GLsuseriv,%
7033     GLSuseriv,%
7034     glsuserv,%
7035     GLsuserv,%
7036     GLSserv,%
7037     glsuservi,%
7038     GLsuservi,%
7039     GLSservi%
7040   }%
7041 %

```

First those which arguments are [*options*]{*label*}{*link text*}.

```

7042   \renewcommand{\do}[1]{%
7043     \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,
usedefault]{%
7044       \doedindexlabel%
7045       \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
7046     }%
7047     \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,usedefault]{%
7048       \doedindexlabel%
7049       \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
7050     }%
7051   }%
7052   \docsvlist{glsdisp,glslink}%
7053 }{}%
7054 }%
7055 %

```

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.

```
7056 \def\@hangingsymbol{}
7057 \newcommand*{\sethangingsymbol}[1]{%
7058   \gdef\@hangingsymbol{#1}%
7059 }%
7060 \newif\ifinstanza
7061 %
```

`\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.

```
7062 \newif\ifinserthangingsymbol
7063 \newcommand{\inserthangingsymbol}{%
7064   \ifinserthangingsymbol%
7065     \ifinstanza%
7066       \@hangingsymbol%
7067     \fi%
7068   \fi%
7069 }
7070 %
```

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.

```
7071 \newcommand*{\ampersand}{\char`\&}
7072
7073 %
```

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.

```

7074 \chardef\body=\catcode`\@
7075 \catcode`\@=11
7076 \chardef\next=\catcode`\&
7077 \catcode`\&=\active
7078
7079 %

```

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.

```

7080 \newcount\stanza@count
7081 \newlength{\stanzaindentbase}
7082 \setlength{\stanzaindentbase}{20pt}
7083
7084 %

```

`\strip@szacnt`
`\setstanzavalues`

The indentations of stanza lines are non-negative integer multiples of the unit called `\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.

```

7085 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
7086 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,,|}%
7087   \stanza@count\z@
7088   \def\next{\expandafter\strip@szacnt\@tempa
7089     \ifx\@tempb\empty\let\next\relax\else
7090     \expandafter\mathchardef\csname #1@number\stanza@count
7091     @\endcsname\@tempb\relax
7092     \advance\stanza@count\@ne\fi\next}%
7093   \next}
7094
7095 %

```

`\setstanzaindents`
`\setstanzapenalties`

In the original edmac, `\setstanzavalues{sza}{⟨...⟩}` had to be called to set the indents, and similarly `\setstanzavalues{szp}{⟨...⟩}` to set the penalties. `\setstanzaindents` and `\setstanzapenalties` macros are a convenience to give the user one less thing to worry about (misspelling the first argument).

```

7096 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
7097 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
7098 %
7099 %

```

`\managestanza@modulo` Since version 0.13, the `stanzaindentsrepetition` counter can be used when the indentation is repeated every n verses. The `\managestanza@modulo` is a command which modifies the counter `stanza@modulo`. The command adds 1 to `stanza@modulo`, but if `stanza@modulo` is equal to the `stanzaindentsrepetition` counter, the command restarts it.

```

7100 \newcounter{stanzaindentsrepetition}
7101 \newcount\stanza@modulo
7102
7103 \newcommand*{\managestanza@modulo}[0]{%
7104     \advance\stanza@modulo\@ne%
7105     \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
7106         \stanza@modulo\@ne%
7107     \fi%
7108 }
7109 %

```

`\stanzaindent` The macro `\stanzaindent`, when called at the beginning of a verse, changes the indentation normally defined for this verse by `\setstanzaindent`. The starred version `\stanzaindent*` skips the current verse for the repetition of stanza indent.

```

7110 \newcommand{\stanzaindent}[1]{%
7111     \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
7112     \ignorespaces%
7113 }%
7114 \WithSuffix\newcommand\stanzaindent*[1]{%
7115     \stanzaindent{#1}%
7116     \global\advance\stanza@modulo-\@ne%
7117     \ifnum\stanza@modulo=0%
7118         \global\stanza@modulo=\value{stanzaindentsrepetition}%
7119     \fi%
7120     \ignorespaces%
7121 }%
7122 %

```

XXVIII.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza23 \newcounter{stanza}
7124 \renewcommand{\thestanza}{%
7125     \textbf{\arabic{stanza}}%
7126 }
7127 %

```

`\ifnumberstanza` Then, macro to activate automatically numbering of stanza.

```

7128 \newif\ifnumberstanza%
7129 %

```

`\@insertstanzanumber` Now, macro called at the first line of of verse of a stanza.

```

7130 \newcommand{\@insertstanzanumber}[0]{%
7131   \ifnumberstanza%
7132   \ifl@dpairing%
7133   \ifledRcol%
7134   \stanzanumwrapper{\thestanzaR}%
7135   \else%
7136   \stanzanumwrapper{\thestanzaL}%
7137   \fi%
7138   \else%
7139   \stanzanumwrapper{\thestanza}%
7140   \fi%
7141   \setline{1}%
7142   \fi%
7143 }%
7144 %

```

`\@advancestanzanumber` Also a command to advance the counter of stanza.

```

7145 \newcommand{\@advancestanzanumber}[0]{%
7146   \ifnumberstanza%
7147   \ifl@dpairing%
7148   \ifledRcol%
7149   \addtocounter{stanzaR}{1}%
7150   \else%
7151   \addtocounter{stanzaL}{1}%
7152   \fi%
7153   \else%
7154   \addtocounter{stanza}{1}%
7155   \fi%
7156   \fi%
7157 }%
7158 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

7159 \newcommand{\stanzanumwrapper}[1]{%
7160   \flagstanza{#1}%
7161 }%
7162 %

```

XXVIII.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

7163 \newcommand{\printstanza}[0]{%
7164   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
7165     l@dprintingcolumns}}{%

```

```

7166         \thestanzaR%
7167     \else%
7168         \thestanzaL%
7169     \fi%
7170 }{%
7171     \thestanza%
7172 }%
7173 }
7174 %

```

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.

`\sza@penalty` 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.

```

7175 \newcommand{\stanza@line}[1][1]{
7176     \ifnum\value{stanzaindentrepetition}=0
7177         \ifcsdef{sza@\number\stanza@count @}%
7178             {%
7179                 \parindent=\csname sza@\number\stanza@count @\endcsname\
7180                 stanzaindentbase%
7181             }{%
7182                 \led@err@StanzaIndentNotDefined%
7183             }%
7184         \else
7185             \ifcsdef{sza@\number\stanza@modulo @}{%
7186                 \parindent=\csname sza@\number\stanza@modulo @\endcsname\
7187                 stanzaindentbase%
7188                 \managestanza@modulo%
7189             }%
7190             {%
7191                 \led@err@StanzaIndentNotDefined%
7192             }%
7193         \fi
7194     \pstart[#1]\stanza@hang\ignorespaces}
7195 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
7196     \hangindent\expandafter
7197     \noexpand\csname sza@0@\endcsname\stanzaindentbase
7198     \hangafter\@ne}
7199 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
7200     \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
7201     \penalty\fi\count@}
7202 %

```


`\@startstanza` Now we have the components of the `\stanza` macro, which appears at the start of a
`\stanza` group of lines. This macro initializes the count and checks to see if hanging indentation
`\@stopstanza` and penalties are to be included. Hanging indentation suspends the line count, so that
`\AtEveryStopStanza` the enumeration is by verse line rather than by print line. If the print line count is
`\AtEveryStanza` desired, invoke `\let\startlock\relax` and do the same for `\endlock`. Here and
`\newverse` above we have used `\xdef` to make the stored macros take up a bit less space, but it also
 makes them more obscure to the reader. Lines of the stanza are delimited by ampersands
 &. The last line of the stanza must end with `\&`.

```

7201 \xdef\@startstanza[#1]{%
7202   \noexpand\instanzatrue\expandafter
7203   \begingroup%
7204   \catcode`\noexpand\&\active%
7205   \global\stanza@count\@ne\stanza@modulo\@ne
7206   \noexpand\ifnum\expandafter\noexpand
7207   \csname sz@00\endcsname=z@\let\noexpand\stanza@hang\relax
7208   \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
7209   \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
7210   \expandafter\noexpand\csname szp@00\endcsname=z@
7211   \let\noexpand\sza@penalty\relax\noexpand\fi%
7212   \def\noexpand&{%
7213     \noexpand\newverse[] []}%
7214   \def\noexpand\&\noexpand\@stopstanza}%
7215   \noexpand\@advancestanza@number%
7216   \noexpand\stanza@line[#1]%
7217   \noexpand\@insertstanza@number%
7218   \let\par\relax\ignorespaces%No paragraph in verses
7219 }
7220
7221 \newcommandx{\stanza}[1][1,usedefault]{%
7222   \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}} and test{\ifstrempy
7223     {\@startstanza[\at@every@stanza]}}%
7224     {\@startstanza[#1]}}%
7225 }%
7226
7227 \newcommandx{\@stopstanza}[1][1,usedefault]{%
7228   \unskip%
7229   \endlock%
7230   \ifboolexpr{not test{\ifdefvoid{\at@every@stop@stanza}} and test{\
7231     ifstrempy{#1}}}%
7232     {\pend[\at@every@stop@stanza]}}%
7233     {\pend[#1]}}%
7234   \endgroup%
7235   \instanzafalse%
7236 }
7237
7238 \newcommand{\AtEveryStopStanza}[1]{%
7239   \ifstrempy{#1}%
7240     {\xdef\at@every@stop@stanza{}}%

```

```

7240     {\gdef\at@every@stop@stanza{#1}}}%
7241   }%
7242   \def\at@every@stop@stanza{}%
7243
7244   \newcommand{\AtEveryStanza}[1]{%
7245     \ifstrempy{#1}%
7246       {\xdef\at@every@stanza{}}}%
7247     {\gdef\at@every@stanza{#1}}}%
7248   }%
7249   \def\at@every@stanza{}%
7250
7251
7252   \newcommand*{\newverse}[2][1,2,usedefault]{%
7253     \unskip%
7254     \endlock\pend[#1]\sza@penalty\global%
7255     \advance\stanza@count\@ne\stanza@line[#2]%
7256   }
7257
7258   %

```

\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`.

```

7259   \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
7260     \hskip -#1\llap{#2}\hskip #1\ignorespaces}
7261
7262   %

```

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.

```

7263   \catcode`\&=\next
7264   \catcode`\@=\body
7265   \setstanzavalues{szp}{0}
7266
7267   %

```

XXIX Apparatus of Manuscripts

XXIX.1 User level macro

\msdata The user level `\msdata` command only writes the manuscripts data in numbered auxiliary file.

```

7268 \newcommand{\msdata}[1]{%
7269   \leavevmode%
7270   \unless\ifstopmsdata@inserted@%
7271     \stopmsdata%
7272     \led@warning@msdatawithoutstop%
7273   \fi%
7274   \global\stopmsdata@inserted@false%
7275   \unless\ifledRcol%
7276     \protected@write\linenum@out{}{%
7277       \string\@msd{#1}%
7278     }%
7279   \else%
7280     \protected@write\linenum@outR{}{%
7281       \string\@msd{#1}%
7282     }%
7283   \fi%
7284 }%
7285 %

```

\stopmsdata The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

7286 \newcommand{\stopmsdata}[0]{%
7287   \leavevmode%
7288   \unless\ifledRcol%
7289     \protected@write\linenum@out{}{%
7290       \string\@stopmsd%
7291     }%
7292   \else%
7293     \protected@write\linenum@outR{}{%
7294       \string\@stopmsd%
7295     }%
7296   \fi%
7297   \global\stopmsdata@inserted@true%
7298 }%
7299 %

```

\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`

```

7300 \newif\ifstopmsdata@inserted@%
7301 %

```

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?

```

7302 \newcommand{\setmsdataseries}[1]{%
7303   \gdef\@msdata@series{#1}%
7304 }%
7305 \def\@msdata@series{A}%
7306 %

```

\setmsdatalabel The label for the manuscripts data.

```

7307 \def\ms@data@label{Ms.}%
7308 \newcommand{\setmsdatalabel}[1]{%
7309   \gdef\ms@data@label{#1}%
7310 }%
7311 %

```

XXIX.3 Counters and lists

\@msd@c \@msd@c is a counter incremented at each \@msd read in auxiliary file.

```

7312 \numdef{\@msd@c}{0}
7313 \numdef{\@msd@cR}{0}
7314 %

```

\@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.

```

7315 \numdef{\@add@msd@c}{0}%
7316 \numdef{\@add@msd@cR}{0}%
7317 %

```

\@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.

```

7318 \def\@msdata@list{}%
7319 %

```

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.

```

7320 \newcommand{\@msd}[1]{%
7321   \unless\ifledRcol%
7322     \numdef{\@msd@c}{\@msd@c+\@ne}%
7323     \csgdef{\@msdata@\@msd@c @data}{#1}%

```

```

7324 \csxdef{@msdata@\msd@c @linenumber}{\the\line@num}%
7325 \csxdef{@msdata@\msd@c @abslinenumber}{\the\absline@num}%
7326 \xright@appenditem{\the\absline@num}\to\actionlines@list%
7327 \xright@appenditem{-1010}\to\actions@list%
7328 \else%
7329 \numdef{@msd@cR}{\msd@cR+\@ne}%
7330 \csgdef{@msdata@\msd@cR @dataR}{#1}%
7331 \csxdef{@msdata@\msd@cR @linenumberR}{\the\line@numR}%
7332 \csxdef{@msdata@\msd@cR @abslinenumberR}{\the\absline@numR}%
7333 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
7334 \xright@appenditem{-1010}\to\actions@listR%
7335 \fi%
7336 }%
7337 %

```

\@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.

```

7338 \newcommand{\@stopmsd}[0]{%
7339 \unless\ifledRcol%
7340 \ifcsundef{@msdata@\msd@c @stoplinenumber}{%
7341 \csxdef{@msdata@\msd@c @stopabslinenumber}{\the\absline@num}%
7342 \csxdef{@msdata@\msd@c @stoplinenumber}{\the\line@num}%
7343 }{}%
7344 \else%
7345 \ifcsundef{@msdata@\msd@cR @stoplinenumberR}{%
7346 \csxdef{@msdata@\msd@cR @stopabslinenumberR}{\the\absline@numR}%
7347 \csxdef{@msdata@\msd@cR @stoplinenumberR}{\the\line@numR}%
7348 }%
7349 {}%
7350 \fi%
7351 }%
7352 %

```

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.

```

7353 \newcommand{\add@msdata}{%
7354 \bgroup%
7355 \normalfont%
7356 \unless\ifledRcol%
7357 \numgdef{\add@msd@c}{\add@msd@c+\@ne}%
7358 \ifcsdef{@msdata@\add@msd@c @data}{%
7359 \letcs{\@data}{@msdata@\add@msd@c @data}%
7360 \edef\l@d@nums{%

```

```

7361      000| % Start page = we don't print it
7362      \csuse{@msdata@add@msd@c @linenumber}| % Start line number
7363      000| % Start subline number, for now, not used
7364      000| % End page number, we don't print it
7365      \ifnumless{\csuse{@msdata@add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}%
7366      {\csuse{@msdata@add@msd@c @stoplinenumber}} % End line number if
in the same page
7367      {\csuse{@lastline@forpage@the\page@num}} % Otherwise, last
number of the page
7368      | %
7369      000| % End sub line number, for now, not used
7370      \edfont@info % Font
7371      } %
7372      \msd@options@fullpagefalse %
7373      \if@firstlineofpage % Try if the data are for the full page. If yes
, will add options to the list.
7374      \unless\if@msdata@insertedfrompreviouspage %
7375      \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse{
{@msdata@add@msd@c @stopabslinenumber}+\@ne}%
7376      { %
7377      \numdef{\@tmp}{\add@msd@c+\@ne}%
7378      \ifcsdef{@msdata@\@tmp @abslinenumber}%
7379      {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}%
7380      {} %
7381      {\msd@options@fullpagetrue}%
7382      } %
7383      {\msd@options@fullpagetrue}%
7384      } %
7385      {} %
7386      \fi %
7387      \fi %
7388      \listxadd{\msdata@list}{ %
7389      \msd@options@iffullpage %
7390      \noexpand\csuse{v\msdata@series footnote}{\msdata@series}{\{
expandonce\l@d@nums}{\msdata@label}{\expandonce\data}} %
7391      \reset@msd@options@iffullpage %
7392      } %
7393      } %
7394      {} %
7395      \else %
7396      \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
7397      \ifcsdef{@msdata@\add@msd@cR @dataR}{ %
7398      \letcs{\@data}{@msdata@\add@msd@cR @dataR}%
7399      \edef\l@d@nums{ %
7400      000| % Start page = we don't print it
7401      \csuse{@msdata@\add@msd@cR @linenumberR}| % Start line number
7402      000| % Start subline number, for now, not used
7403      000| % End page number, we don't print it

```

```

7404 \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}}%
7405 {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
7406 {\csuse{@lastline@forpageR@the\page@numR}}}%Otherwiser, last
number of the page
7407 |%
7408 000|% End sub line number, for now, not used
7409 \edfont@info%Font
7410 }%
7411 \@msd@options@fullpagefalse%
7412 \if@firstlineofpageR%
7413 \unless\if@msdata@insertedfrompreviouspage%
7414 \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
csuse{@msdata@\add@msd@c @stopabslinenumberR}+\@one}%
7415 {%
7416 \numdef{\@tmp}{\add@msd@cR+\@one}%
7417 \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7418 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
7419 {}%
7420 {\@msd@options@fullpagetrue}%
7421 }%
7422 {\@msd@options@fullpagetrue}%
7423 }%
7424 {}%
7425 \fi%
7426 \fi%
7427 \listxadd{\@msdata@list}{%
7428 \@msd@options@iffullpage%
7429 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
7430 \reset@msd@options@iffullpage%
7431 }%
7432 }%
7433 {}%
7434 \fi%
7435 \egroup%
7436 }%
7437 %

```

`\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.

```

7438 \newif\if@msdata@insertedfrompreviouspage%
7439 %

```

`\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.

```

7440 \newcommand{\add@msdata@firstlineofpage}{%
7441   \bgroup%
7442   \normalfont%
7443   \unless\ifledRcol@%
7444     \ifcsdef{@msdata@\add@msd@c @data}{%
7445       \ifnumless{\the\absline@num-\@ne}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}}%
7446       {%
7447         \global\@msdata@insertedfrompreviouspagetrue%
7448         \letcs{@data}{@msdata@\add@msd@c @data}%
7449         \edef\l@d@nums{%
7450           000}% Start page = we don't print it
7451           \numexpr\the\line@num+\@ne\relax}% Start line number = first line
of the page. As \add@msdata@firstlineofpage is called before line number
has been incremented, we increment it for printing
7452           000}% Start subline number, for now, not used
7453           000}% End page number, we don't print it
7454           \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
7455           {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
7456           {\csuse{@lastline@forpage@\the\page@num}}}%Otherwise, last
number of the page
7457           |%
7458           000}% End sub line number, for now, not used
7459           \edfont@info%Font
7460           }%
7461           \@msd@options@fullpagefalse%
7462           \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
7463           {%
7464             \numdef{@tmp}{\add@msd@c+\@ne}%
7465             \ifcsdef{@msdata@\@tmp @abslinenumber}%
7466               {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
7467               {}%
7468               {\@msd@options@fullpagetrue}%
7469               }%
7470               {\@msd@options@fullpagetrue}%
7471               }%
7472             {}%
7473             \listxadd{\@msdata@list}{%
7474               \@msd@options@iffullpage%
7475               \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}}%
7476             \reset@msd@options@iffullpage%
7477             }%
7478           }%
7479           {\global\@msdata@insertedfrompreviouspagefalse}%

```



```

7480 }{}%
7481 \else%
7482 \ifcsdef{@msdata@add@msd@cR @dataR}{%
7483 \ifnumless{\the\absline@numR-\@one}{\csuse{@msdata@add@msd@cR
@stopabslinenumberR}}}%
7484 {%
7485 \global\@msdata@insertedfrompreviouspagetrue%
7486 \letcs{\@data}{@msdata@add@msd@cR @dataR}%
7487 \edef\l@d@nums{%
7488 000}% Start page = we don't print it
7489 \numexpr\the\line@numR+\@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
7490 000}% Start subline number, for now, not used
7491 000}% End page number, we don't print it
7492 \ifnumless{\csuse{@msdata@add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}}%
7493 {\csuse{@msdata@add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
7494 {\csuse{@lastline@forpageR@the\page@numR}}}%Otherwise, last
number of the page
7495 |%
7496 000}% End sub line number, for now, not used
7497 \edfont@info%Font
7498 }%
7499 \@msd@options@fullpagefalse%
7500 \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\csuse{
@msdata@add@msd@cR @stopabslinenumberR}+\@one}%
7501 {%
7502 \numdef{\@tmp}{\add@msd@cR+\@one}%
7503 \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7504 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
7505 {}%
7506 {\@msd@options@fullpagetrue}%
7507 }%
7508 {\@msd@options@fullpagetrue}%
7509 }%
7510 {}%
7511 \listxadd{\@msdata@list}{%
7512 \@msd@options@iffullpage%
7513 \noexpand\csuse{v@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
7514 \reset@msd@options@iffullpage%
7515 }%
7516 }%
7517 {\global\@msdata@insertedfrompreviouspagefalse}%
7518 }{}%
7519 \fi%
7520 \egroup%

```

```
7521 }%
7522 %
```

XXIX.6 Inserting footnote

Just before inserting standard insert (familiar and critical footnotes, sidenotes), we call `\insert@msdata` to insert manuscripts data's footnotes.

```
\insert@msdata%23 \newcommand{\insert@msdata}{%
7524 \def\do##1{##1}%
7525 \dolistloop{\@msdata@list}%
7526 \global\let\@msdata@list\relax%
7527 }%
7528 %
```

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.

```
7529 \newif\if@msd@options@fullpage%
7530 \newcommand{\@msd@options@iffullpage}[0]{%
7531 \if@msd@options@fullpage%
7532 \noexpand\toggletrue{nonum@}%
7533 \ifdefvoid{\ms@data@label}%
7534 {\noexpand\toggletrue{nosep@}}%
7535 }%
7536 \fi%
7537 }%
7538 \newcommand{\reset@msd@options@iffullpage}[0]{%
7539 \noexpand\togglefalse{nonum@}%
7540 \noexpand\togglefalse{nosep@}%
7541 }%
7542 %
```

XXX Arrays and tables

XXX.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread ‘`eq` and `amstex`’, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsgen` package.

```

7543 \newtoks\@emptytoks
7544
7545 %

```

The rest is from amsmath.

\l@denbody A token register to contain the body.

```

7546 \newtoks\l@denbody
7547
7548 %

```

\addtol@denbody `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```

7549 \newcommand{\addtol@denbody}[1]{%
7550   \global\l@denbody\expandafter{\the\l@denbody#1}}
7551
7552 %

```

\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`.

```

7553 \newcommand{\l@dcollect@body}[1]{%
7554   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}}%
7555   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}}%
7556   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
7557   \begingroup
7558     \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
7559     \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
7560     \processl@denbody%
7561   }%
7562
7563 %

```

\l@dpush@begins When adding a piece of the current environment's contents to `\l@denbody`, we scan it to check for additional `\begin` tokens, and add a 'b' to the stack for any that we find.

```

7564 \def\l@dpush@begins#1\begin#2{%
7565   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
7566
7567 %

```

\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.

```

7568 \def\l@dcollect@@body#1\end#2{%
7569   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
7570                       \expandafter\@gobble\l@dbegin@stack}%
7571   \ifx\@empty\l@dbegin@stack
7572     \endgroup
7573     \@checkend{#2}%
7574     \addtol@denbody{#1}%
7575   \else
7576     \addtol@denbody{#1\end{#2}}%
7577   \fi
7578   \processl@denbody % A little tricky! Note the grouping
7579 }
7580
7581 %

```

There was a question on CTT about how to use `\collect@body` for a macro taking an argument. The following is part of that thread.

From: Heiko Oberdiek <oberdiek@uni-freiburg.de>
 Newsgroups: comp.text.tex
 Subject: Re: Using `\collect@body` with commands that take >1 argument
 Date: Fri, 08 Aug 2003 09:03:20 +0200

eed132@psu.edu (Evan) wrote:
 > I'm trying to make a new Latex environment that acts like the
 > `\colorbox` command that is part of the color package. I looked through
 > the FAQ and ran across this bit about using the `\collect@body` command
 > that is part of AMSLaTeX:
 > <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=cmdasenv>
 >
 > It almost works. If I do something like the following:
 > `\newcommand{\redbox}[1]{\colorbox{red}{#1}}`
 >
 > `\makeatletter`
 > `\newenvironment{redbox}{\collect@body \redbox}{}`

You will get an error message: Command `\redbox` already defined.
 Thus you must rename either the command `\redbox` or the environment
 name.

```

> \begin{coloredbox}{blue}
>   Yadda yadda yadda... this is on a blue background...
> \end{coloredbox}
> and can't figure out how to make the \collect@body take this.

> \collect@body \colorbox{red}

```

```
> \collect@body {\colorbox{red}}
```

The argument of `\collect@body` has to be one token exactly.

```
\documentclass{article}
\usepackage{color}
\usepackage{amsmath}

\newcommand{\redbox}[1]{\colorbox{red}{#1}}
\makeatletter
\newenvironment{coloredbox}[1]{%
  \def\next@{\colorbox{#1}}%
  \collect@body\next@
}{%

% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}[1]{%
  \def\next@{\mycoloredbox{#1}}%
  \collect@body\next@
}{%
\newcommand{\mycoloredbox}[2]{%
  \colorbox{#1}{\ignorespaces#2\unskip}%
}

% support of optional color model argument
\newcommand\coloredboxIII\endcsname{}
\def\coloredboxIII#1#2{%
  \@coloredboxIII{#1}%
}
\def\@coloredboxIII#1#2{%
  \def\next@{\mycoloredboxIII{#1}{#2}}%
  \collect@body\next@
}
\newcommand{\mycoloredboxIII}[3]{%
  \colorbox{#1}{#2}{\ignorespaces#3\unskip}%
}

\makeatother

\begin{document}
  Black text before
  \begin{coloredbox}{blue}
    Hello World
  \end{coloredbox}
  Black text after

  Black text before
  \begin{coloredboxII}{blue}
    Hello World
  \end{coloredboxII}
```

```

Black text after

Black text before
\begin{coloredboxIII}[rgb]{0,0,1}
  Hello World
\end{coloredboxIII}
Black text after

\end{document}

Yours sincerely
Heiko <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.

```

7582 \newcommand*{\l@dtabnoexpands}{%
7583   \let\rtab=0%
7584   \let\ctab=0%
7585   \let\ltab=0%
7586   \let\rtabtext=0%
7587   \let\ltabtext=0%
7588   \let\ctabtext=0%
7589   \let\edbeforetab=0%
7590   \let\edaftertab=0%
7591   \let\edatleft=0%
7592   \let\edatright=0%
7593   \let\edvertline=0%
7594   \let\edvertdots=0%
7595   \let\edrowfill=0%
7596 }
7597
7598 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in `edtabularx` and `edarrayx` environments.

`\restore@familiarnotes`

```

7599 \newcommand{\disable@familiarnotes}{%
7600   \unless\ifnofamiliar@%
7601     \def\do##1{%
7602       \csletcs{footnote@@##1}{footnote##1}%
7603       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
7604         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
7605         \csuse{@footnotemark##1}%
7606       }%
7607     }%
7608     \dolistloop{\@series}%
7609   \fi%
7610 }%
7611 \newcommand{\restore@familiarnotes}{%
7612   \unless\ifnofamiliar@%
7613     \def\do##1{%
7614       \csletcs{footnote##1}{footnote@@##1}%
7615     }%
7616     \dolistloop{\@series}%
7617   \fi%
7618 }%
7619 %
7620 %

```

\disable@sidenotes The same, for side notes.

\restore@sidenotes

```

7621 \newcommand{\disable@sidenotes}{%
7622   \let\@@ledrightnote\ledrightnote%
7623   \let\@@ledleftnote\ledleftnote%
7624   \let\@@ledsidenote\ledsidenote%
7625   \let\ledrightnote\@gobble%
7626   \let\ledleftnote\@gobble%
7627   \let\ledsidenote\@gobble%
7628 }%
7629 \newcommand{\restore@sidenotes}{%
7630   \let\ledrightnote\@@ledrightnote%
7631   \let\ledleftnote\@@ledleftnote%
7632   \let\ledsidenote\@@ledsidenote%
7633 }%
7634 %

```

\disable@notes Disable/restore side and familiar notes.

\restore@notes

```

7635 \newcommand{\disable@notes}{%
7636   \disable@sidenotes%
7637   \disable@familiarnotes%
7638 }%
7639 \newcommand{\restore@notes}{%
7640   \restore@sidenotes%
7641   \restore@familiarnotes%
7642 }%

```

```
7643 %
```

\EDTEXT We need to be able to modify the `\edtext` macros and also restore their original definitions.

\xedtext

```
7644 \let\EDTEXT=\edtext
7645 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
7646 %
```

\EDLABEL We need to be able to modify and restore the `\edlabel` macro.

\xedlabel

```
7647 \let\EDLABEL=\edlabel
7648 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}
7649 %
```

\xedindex Macros supporting modification and restoration of `\edindex`.

\nulledindex

```
7650 \AtBeginDocument{\let\xedindex\edindex}%
7651 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
7652 %
7653 %
```

\@line@@num Macro supporting restoration of `\linenum`.

```
7654 \let\@line@@num=\linenum
7655 %
```

\l@dgobblearg `\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```
7656 \newcommand*{\l@dgobbleoptarg}[2][\relax]%
7657 %
7658 %
```

\Relax₅₉ `\let\Relax=\relax`

\NEXT₆₀ `\let\NEXT=\next`

```
7661 %
7662 %
```

\l@dmodforedtext Modify and restore various macros for when `\edtext` is used.

\l@drestoreforedtext

```
7663 \newcommand{\l@dmodforedtext}{%
7664   \let\edtext\relax
7665   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
7666   \dolistloop{\@series}%
7667   \let\edindex\nulledindex
7668   \let\linenum@gobble}
7669 \newcommand{\l@drestoreforedtext}{%
7670   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}}
```



```

7671 \dolistloop{\@series}%
7672 \let\edindex\xedindex}
7673 %

```

\l@dnnullfills Nullify and restore some column fillers, etc.

```

\l@drestorefills
7674 \newcommand{\l@dnnullfills}{%
7675 \def\edlabel##1{%
7676 \def\edrowfill##1##2##3{%
7677 }
7678 \newcommand{\l@drestorefills}{%
7679 \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
7680 }
7681 %
7682 %

```

\letsforverteilen Gathers some lets and other code that is common to the **verteilen** macros.

```

7683 \newcommand{\letsforverteilen}{%
7684 \let\edtext\xedtext
7685 \let\edindex\xedindex
7686 \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
7687 \dolistloop{\@series}%
7688 \let\linenum\@line@num
7689 \hilfsskip=\l@dcolwidth%
7690 \advance\hilfsskip by -\wd\hilfsbox
7691 \def\edlabel##1{\xedlabel{##1}}
7692 %
7693 %

```

\disablel@dtabfeet Declarations for using or using `\edtext` inside tabulars. The default at this point is for `\edtext`.

```

7694 \newcommand{\disablel@dtabfeet}{\l@dmodforedtext}%
7695 \newcommand{\enablel@dtabfeet}{\l@drestoreforedtext}%
7696 %

```

XXX.2.2 Counters, boxes and lengths

\l@dampcount `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter for the columns.

```

7697 \newcount\l@dampcount
7698 \l@dampcount=1\relax
7699 \newcount\l@dcolcount
7700 \l@dcolcount=0\relax
7701 %
7702 %

```

```

\hilfsbox Some (temporary) helper items.
\hilfsskip
\Hilfsbox
\hilfscount
7703 \newbox\hilfsbox
7704 \newskip\hilfsskip
7705 \newbox\Hilfsbox
7706 \newcount\hilfscount
7707
7708 %

```

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).

```

7709 \newdimen\dcoli
7710 \newdimen\dcolii
7711 \newdimen\dcoliii
7712 \newdimen\dcoliv
7713 \newdimen\dcolv
7714 \newdimen\dcolvi
7715 \newdimen\dcolvii
7716 \newdimen\dcolviii
7717 \newdimen\dcolix
7718 \newdimen\dcolx
7719 \newdimen\dcolxi
7720 \newdimen\dcolxii
7721 \newdimen\dcolxiii
7722 \newdimen\dcolxiv
7723 \newdimen\dcolxv
7724 \newdimen\dcolxvi
7725 \newdimen\dcolxvii
7726 \newdimen\dcolxviii
7727 \newdimen\dcolxix
7728 \newdimen\dcolxx
7729 \newdimen\dcolxxi
7730 \newdimen\dcolxxii
7731 \newdimen\dcolxxiii
7732 \newdimen\dcolxxiv
7733 \newdimen\dcolxxv
7734 \newdimen\dcolxxvi
7735 \newdimen\dcolxxvii
7736 \newdimen\dcolxxviii
7737 \newdimen\dcolxxix
7738 \newdimen\dcolxxx
7739 \newdimen\dcolerr % added for error handling
7740
7741 %

```

`\l@dcolwidth` This is a cunning way of storing the columnwidths indexed by the column number `\l@dcolcount`, like an array. (was `\Dimenzuordnung`)

```

7742 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???

```

```

7743 \or \dcoli \or \dcolii \or \dcoliii
7744 \or \dcoliv \or \dcolv \or \dcolvi
7745 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
7746 \or \dcolxi \or \dcolxii \or \dcolxiii
7747 \or \dcolxiv \or \dcolxv \or \dcolxvi
7748 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
7749 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
7750 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
7751 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
7752 \else \dcolerr \fi}
7753
7754 %

```

\step1@dcolcount This increments the column counter, and issues an error message if it is too large.

```

7755 \newcommand*{\step1@dcolcount}{\advance\l@dcolcount\@ne
7756 \ifnum\l@dcolcount>30\relax
7757 \led@err@TooManyColumns
7758 \fi}
7759
7760 %

```

\l@dsetmaxcolwidth Sets the column width to the maximum value seen so far.

```

7761 \newcommand{\l@dsetmaxcolwidth}{%
7762 \ifdim\l@dcolwidth < \wd\hilfsbox
7763 \l@dcolwidth = \wd\hilfsbox
7764 \else \relax \fi}
7765
7766 %

```

\measurecell Measure (recursively) the width required for a math cell.

```

7767 \def\measurecell #1{%
7768 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
7769 \else\l@dcheckcols%
7770 \l@dcolcount=0%
7771 \let\NEXT\measurecell%
7772 \fi%
7773 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7774 \step1@dcolcount%
7775 \l@dsetmaxcolwidth%
7776 \let\NEXT\measurecell%
7777 \fi\NEXT}
7778
7779 %

```

\measuretcell Measure (recursively) the width required for a text cell.

```

7780 \def\measuretcell #1{%
7781   \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
7782     \else\l@dcheckcols%
7783       \l@dcolcount=0%
7784       \let\NEXT\measuretcell%
7785     \fi%
7786   \else\setbox\hilfsbox=\hbox{#1}%
7787     \step\l@dcolcount%
7788     \l@dsetmaxcolwidth%
7789     \let\NEXT\measuretcell%
7790   \fi\NEXT}
7791
7792 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

7793 \def\measuremrow #1\{%
7794   \ifx #1&\let\NEXT\relax%
7795   \else\measuremcell #1\&\&\&%
7796     \let\NEXT\measuremrow%
7797   \fi\NEXT}
7798 %

```

\measuretrrow Measure (recursively) the width required for a text row.

```

7799 \def\measuretrrow #1\{%
7800   \ifx #1&\let\NEXT\relax%
7801   \else\measuretcell #1\&\&\&%
7802     \let\NEXT\measuretrrow%
7803   \fi\NEXT}
7804
7805 %

```

\edtabcolsep The length \edtabcolsep controls the distance between columns.

```

7806 \newskip\edtabcolsep
7807 \global\edtabcolsep=10pt
7808
7809 %

```

\variab₁₀ \newcommand{\variab}{\relax}

```

7811
7812 %

```

\l@dcheckcols Check that the number of columns is consistent.

```

7813 \newcommand*{\l@dcheckcols}{%
7814   \ifnum\l@dcolcount=1\relax

```

```

7815 \else
7816 \ifnum\l@dampcount=1\relax
7817 \else
7818 \ifnum\l@dcolcount=\l@dampcount\relax
7819 \else
7820 \l@d@err@UnequalColumns
7821 \fi
7822 \fi
7823 \l@dampcount=\l@dcolcount
7824 \fi}
7825
7826 %

```

\edfilldimen A length.

```

7827 \newdimen\edfilldimen
7828 \edfilldimen=0pt
7829
7830 %

```

\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

```

7831 \newcounter{addcolcount}
7832 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
7833 %

```

XXX.2.3 Tabular typesetting

\setmcellright Typeset (recursively) cells of display math right justified.

```

7834 \def\setmcellright #1&{\def\edlabel##1{}%
7835 \let\edindex\nulledindex
7836 \ifx #1\\ \ifnum\l@dcolcount=0%\removeelastskip
7837 \let\Next\relax%
7838 \else\l@dcolcount=0%
7839 \let\Next=\setmcellright%
7840 \fi%
7841 \else%
7842 \disablel@dtabfeet%
7843 \step1@dcolcount%
7844 \disable@notes%
7845 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7846 \restore@notes%
7847 \letsforverteilen%
7848 \hskip\hilfsskip$\displaystyle{#1}$%
7849 \hskip\edtabcolsep%
7850 \let\Next=\setmcellright%

```

```

7851 \fi\Next}
7852
7853 %

```

\settccllright Typeset (recursively) cells of text right justified.

```

7854 \def\settccllright #1{\def\edlabel##1{}%
7855 \let\edindex\nulledindex
7856 \ifx #1\\ \ifnum\l@dcclcount=0\removelastskip
7857 \let\Next\relax%
7858 \else\l@dcclcount=0%
7859 \let\Next=\settccllright%
7860 \fi%
7861 \else%
7862 \disablel@dtabfeet%
7863 \step1@dcclcount%
7864 \disable@notes%
7865 \setbox\hilfsbox=\hbox{#1}%
7866 \restore@notes%
7867 \letsforverteilen%
7868 \hskip\hilfsskip#1%
7869 \hskip\edtabcolsep%
7870 \let\Next=\settccllright%
7871 \fi\Next}
7872 %

```

\setmcellleft Typeset (recursively) cells of display math left justified.

```

7873 \def\setmcellleft #1{\def\edlabel##1{}%
7874 \let\edindex\nulledindex
7875 \ifx #1\\ \ifnum\l@dcclcount=0 \let\Next\relax%
7876 \else\l@dcclcount=0%
7877 \let\Next=\setmcellleft%
7878 \fi%
7879 \else \disablel@dtabfeet%
7880 \step1@dcclcount%
7881 \disable@notes%
7882 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7883 \restore@notes%
7884 \letsforverteilen%
7885 $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
7886 \let\Next=\setmcellleft%
7887 \fi\Next}
7888
7889 %

```

\settcclleft Typeset (recursively) cells of text left justified.

```

7890 \def\settcclleft #1{\def\edlabel##1{}%
7891 \let\edindex\nulledindex

```

```

7892 \ifx #1\ \ifnum\l@dc colcount=0 \let\Next\relax%
7893 \else\l@dc colcount=0%
7894 \let\Next=\settc lleft%
7895 \fi%
7896 \else \disablel@dt abfeet%
7897 \stepl@dc colcount%
7898 \disable@notes%
7899 \setbox\hilfsbox=\hbox{#1}%
7900 \restore@notes%
7901 \letsforverteilen%
7902 #1\hskip\hilfsskip\hskip\edtabcolsep%
7903 \let\Next=\settc lleft%
7904 \fi\Next}
7905 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

7906 \def\setmcellcenter #1{\def\edlabel##1}%
7907 \let\edindex\nulledindex
7908 \ifx #1\ \ifnum\l@dc colcount=0 \let\Next\relax%
7909 \else\l@dc colcount=0%
7910 \let\Next=\setmcellcenter%
7911 \fi%
7912 \else \disablel@dt abfeet%
7913 \stepl@dc colcount%
7914 \disable@notes%
7915 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7916 \restore@notes%
7917 \letsforverteilen%
7918 \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
7919 \hskip\edtabcolsep%
7920 \let\Next=\setmcellcenter%
7921 \fi\Next}
7922 %
7923 %

```

\settc lcenter Typeset (recursively) cells of text centered.

```

7924 \def\settc lcenter #1{\def\edlabel##1}%
7925 \let\edindex\nulledindex
7926 \ifx #1\ \ifnum\l@dc colcount=0 \let\Next\relax%
7927 \else\l@dc colcount=0%
7928 \let\Next=\settc lcenter%
7929 \fi%
7930 \else \disablel@dt abfeet%
7931 \stepl@dc colcount%
7932 \disable@notes%
7933 \setbox\hilfsbox=\hbox{#1}%
7934 \restore@notes%
7935 \letsforverteilen%

```

```

7936         \hskip 0.5\hlfsskip #1\hskip 0.5\hlfsskip%
7937         \hskip\edtabcolsep%
7938         \let\Next=\settcellcenter%
7939     \fi\Next}
7940
7941 %

```

```

\NEXT142 \let\NEXT=\relax
7943
7944 %

```

\setmrowright Typeset (recursively) rows of right justified math.

```

7945 \def\setmrowright #1\{%
7946     \ifx #1& \let\NEXT\relax
7947     \else \centerline{\setmcellright #1&\\&\\&}
7948         \let\NEXT=\setmrowright
7949     \fi\NEXT}
7950 %

```

\settroright Typeset (recursively) rows of right justified text.

```

7951 \def\settroright #1\{%
7952     \ifx #1& \let\NEXT\relax
7953     \else \centerline{\settcellright #1&\\&\\&}
7954         \let\NEXT=\settroright
7955     \fi\NEXT}
7956
7957 %

```

\setmrowleft Typeset (recursively) rows of left justified math.

```

7958 \def\setmrowleft #1\{%
7959     \ifx #1& \let\NEXT\relax
7960     \else \centerline{\setmcellleft #1&\\&\\&}
7961         \let\NEXT=\setmrowleft
7962     \fi\NEXT}
7963 %

```

\settrorleft Typeset (recursively) rows of left justified text.

```

7964 \def\settrorleft #1\{%
7965     \ifx #1& \let\NEXT\relax
7966     \else \centerline{\settcellleft #1&\\&\\&}
7967         \let\NEXT=\settrorleft
7968     \fi\NEXT}
7969
7970 %

```


\setmrowcenter Typeset (recursively) rows of centered math.

```

7971 \def\setmrowcenter #1\{\%
7972   \ifx #1& \let\NEXT\relax%
7973   \else \centerline{\setmcellcenter #1&\&\&}
7974     \let\NEXT=\setmrowcenter
7975   \fi\NEXT}
7976 %

```

\settextcenter Typeset (recursively) rows of centered text.

```

7977 \def\settextcenter #1\{\%
7978   \ifx #1& \let\NEXT\relax
7979   \else \centerline{\settextcellcenter #1&\&\&}
7980     \let\NEXT=\settextcenter
7981   \fi\NEXT}
7982 %
7983 %

```

\nullsetzen \newcommand{\nullsetzen}{%

```

7985   \step1@dcolcount%
7986   \l@dcolwidth=0pt%
7987   \ifnum\l@dcolcount=30\let\NEXT\relax%
7988     \l@dcolcount=0\relax
7989   \else\let\NEXT\nullsetzen%
7990   \fi\NEXT}
7991 %
7992 %

```

\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.

```

7993 \newcommand{\edatleft}[3][\@empty]{%
7994   \ifx#1\@empty
7995     \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
7996       depth 0pt \right. $\hss}\vfil}
7997   \else
7998     \vbox to 4pt{\vss\hbox{#1\left#2\vrule width0pt height #3
7999       depth 0pt \right. $\}\vfil}
8000   \fi}
8001 %

```

\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.

```

8002 \newcommand{\edatright}[3][\@empty]{%
8003   \ifx#1\@empty
8004     \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
8005       depth 0pt \right#2 $\hss}\vfil}

```

```

8006 \else
8007   \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
8008             depth 0pt \right#2 #1 $\}\vfil}
8009 \fi}
8010
8011 %

```

\edvertline `\edvertline{⟨len⟩}` vertical line ⟨len⟩ high.

```

8012 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
8013
8014 %

```

\edvertdots `\edvertdots{⟨len⟩}` vertical dotted line ⟨len⟩ high.

```

8015 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
8016   {\cleaders\hbox{$\math\hbox{.}\vbox to 0.5em{ }$}\vfil}}}
8017
8018 %

```

\l@dtabaddcols `\l@dtabaddcols{⟨startcol⟩}{⟨endcol⟩}` adds the widths of the columns ⟨startcol⟩ through ⟨endcol⟩ to `\edfilldimen`. It is a \LaTeX style reimplementation of the original `\@add@`.

```

8019 \newcommand{\l@dtabaddcols}[2]{%
8020   \l@dcheckstartend{#1}{#2}%
8021   \ifl@dstartendok
8022     \setcounter{addcolcount}{#1}%
8023     \@whilenum \value{addcolcount}<#2\relax \do
8024       {\advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
8025        \advance\edfilldimen by \edtabcolsep
8026        \stepcounter{addcolcount}}%
8027     \advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
8028   \fi
8029 }
8030
8031 %

```

\ifl@dstartendok `\l@dcheckstartend{⟨startcol⟩}{⟨endcol⟩}` checks that the values of ⟨startcol⟩ and `\l@dcheckstartend` ⟨endcol⟩ are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```

8032 \newif\ifl@dstartendok
8033 \newcommand{\l@dcheckstartend}[2]{%
8034   \l@dstartendoktrue
8035   \ifnum #1<\@ne
8036     \l@dstartendokfalse
8037     \led@err@LowStartColumn
8038   \fi
8039   \ifnum #2>30\relax
8040     \l@dstartendokfalse

```

```

8041 \led@err@HighEndColumn
8042 \fi
8043 \ifnum #1>#2\relax
8044 \l@startendokfalse
8045 \led@err@ReverseColumns
8046 \fi
8047 }
8048
8049 %

```

\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.

```

8050 \newcommand*{\edrowfill}[3]{%
8051 \l@dtabaddcols{#1}{#2}%
8052 \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
8053 \let\@edrowfill=\edrowfill
8054 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
8055
8056 %

```

\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`.

```

8057 \newcommand{\leftltab}[1]{%
8058 \hb@xt@ \z@{\vbox{\edtabindent%
8059 \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8060
8061 %

```

\leftrtab `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

8062 \newcommand{\leftrtab}[2]{%
8063 #2\hb@xt@ \z@{\vbox{\edtabindent%
8064 \advance\Hilfsskip by\dcoli%
8065 \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8066
8067 %

```

\leftctab `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

8068 \newcommand{\leftctab}[2]{%
8069 \hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%

```

```

8070 \advance\Hilfsskip by 0.5\dcoli%
8071 \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8072 \disablel@dtabfeet$\displaystyle{#2}$}%
8073 \advance\Hilfsskip by -0.5\wd\hilfsbox%
8074 \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
8075 #2}
8076
8077 %

```

\rightctab `\rightctab{<math>}<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

8078 \newcommand{\rightctab}[2]{%
8079 \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8080 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8081 #1\hb@xt@{\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8082 \advance\Hilfsskip by 0.5\l@dcolwidth%
8083 \advance\Hilfsskip by -\wd\hilfsbox%
8084 \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8085 \disablel@dtabfeet$\displaystyle{#1}$}%
8086 \advance\Hilfsskip by -0.5\wd\hilfsbox%
8087 \advance\Hilfsskip by \edtabcolsep%
8088 \moveright\Hilfsskip\hbox{ #2}}\hss}%
8089 }
8090
8091 %

```

\rightltab `\rightltab{<math>}<math>}{<text>}` for `\edaftertab` in `\ltab`.

```

8092 \newcommand{\rightltab}[2]{%
8093 \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8094 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8095 #1\hb@xt@{\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8096 \advance\Hilfsskip by\l@dcolwidth%
8097 \advance\Hilfsskip by-\wd\hilfsbox%
8098 \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8099 \disablel@dtabfeet$\displaystyle{#1}$}%
8100 \advance\Hilfsskip by-\wd\hilfsbox%
8101 \advance\Hilfsskip by\edtabcolsep%
8102 \moveright\Hilfsskip\hbox{ #2}}\hss}%
8103 }
8104
8105 %

```

\rightrtab `\rightrtab{<math>}<math>}{<text>}` for `\edaftertab` in `\rtab`.

```

8106 \newcommand{\rightrtab}[2]{%
8107 \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8108 \disablel@dtabfeet#2}%
8109 #1\hb@xt@{\z@\vbox{\edtabindent%
8110 \advance\Hilfsskip by-\wd\hilfsbox%

```

```

8111 \advance\Hilfsskip by\edtabcolsep%
8112 \moveright\Hilfsskip\hbox{ #2}\hss}%
8113 }
8114
8115 %

```

\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.

```

8116 \newcommand{\rtab}[1]{%
8117 \l@nullfills
8118 \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
8119 \def\edaftertab##1##2{\righttab{##1}{##2}}%
8120 \measurebody{#1}%
8121 \l@drestorefills
8122 \variab
8123 \setmrowright #1\&\%
8124 \enablel@dtabfeet}
8125
8126 %

```

\measurebody `\measurebody{<body>}` measures the array `<body>`.

```

8127 \newcommand{\measurebody}[1]{%
8128 \disablel@dtabfeet%
8129 \l@dcolcount=0%
8130 \nullsetzen%
8131 \l@dcolcount=0
8132 \measuremrow #1\&\%
8133 \global\l@dampcount=1}
8134
8135 %

```

\rtabtext `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```

8136 \newcommand{\rtabtext}[1]{%
8137 \l@nullfills
8138 \measuretbody{#1}%
8139 \l@drestorefills
8140 \variab
8141 \setthrowright #1\&\%
8142 \enablel@dtabfeet}
8143
8144 %

```

\measuretbody `\measuretbody{<body>}` measures the tabular `<body>`.

```

8145 \newcommand{\measuretbody}[1]{%
8146   \disable@notes%
8147   \disablel@dtabfeet%
8148   \l@dcolcount=0%
8149   \nullsetzen%
8150   \l@dcolcount=0
8151   \measuretrrow #1\\&\\%
8152   \restore@notes%
8153   \global\l@dampcount=1}
8154
8155 %

```

\ltab Array with entries left justified.

```

\edbeforetab \newcommand{\ltab}[1]{%
\edaftertab  \l@dnullfills
8156
8157   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
8158   \def\edaftertab##1##2{\rightltab{##1}{##2}}%
8159   \measuretbody{#1}%
8160   \l@drestorefills
8161   \variab
8162   \setmrowleft #1\\&\\%
8163   \enablel@dtabfeet}
8164
8165 %
8166 %

```

\ltabtext Tabular with entries left justified.

```

8167 \newcommand{\ltabtext}[1]{%
8168   \l@dnullfills
8169   \measuretbody{#1}%
8170   \l@drestorefills
8171   \variab
8172   \settrrowleft #1\\&\\%
8173   \enablel@dtabfeet}
8174
8175 %

```

\ctab Array with centered entries.

```

\edbeforetab \newcommand{\ctab}[1]{%
\edaftertab  \l@dnullfills
8176
8177   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
8178   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
8179   \measuretbody{#1}%
8180   \l@drestorefills
8181   \variab
8182   \setmrowcenter #1\\&\\%
8183   \enablel@dtabfeet}
8184
8185 %
8186 %

```

`\ctabtext` Tabular with entries centered.

```

8187 \newcommand{\ctabtext}[1]{%
8188   \l@dnnullfills
8189   \measuretbody{#1}%
8190   \l@drestorefills
8191   \variab
8192   \settrowcenter #1\\&\\%
8193   \enablel@dtabfeet}
8194
8195 %

```

`\spreadtext`⁹⁶ `\newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%`
`\hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}`
`%`

`\spreadmath`⁹⁹ `\newcommand{\spreadmath}[1]{%`
`\hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}`
`%`

`\HILFSskip` More helpers.

`\Hilfsskip`
`\newskip\HILFSskip`
`\newskip\Hilfsskip`
`%`

`\EDTABINDENT`¹⁰⁷ `\newcommand{\EDTABINDENT}{%`
`\ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%`
`\else\step1@dcolcount%`
`\advance\Hilfsskip by\l@dcolwidth%`
`\ifdim\l@dcolwidth=0pt\advance\hilfscount\@ne`
`\else\advance\Hilfsskip by \the\hilfscount\edtabcolsep%`
`\hilfscount=1\fi%`
`\let\NEXT=\EDTABINDENT%`
`\fi\NEXT}%`
`%`

`\edtabindent` (was `\tabindent`)

```

8217 \newcommand{\edtabindent}{%
8218   \l@dcolcount=0\relax
8219   \Hilfsskip=0pt%
8220   \hilfscount=1\relax
8221   \EDTABINDENT%
8222   \hilfsskip=\hspace%

```

```

8223 \advance\hilfsskip -\Hilfsskip%
8224 \Hilfsskip=0.5\hilfsskip%
8225 }%
8226
8227 %

```

\EDTAB (was \TAB)

```

8228 \def\EDTAB #1|#2|{%
8229 \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
8230 \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
8231 \advance\tabelskip -\wd\tabhilfbox%
8232 \advance\tabelskip -\wd\tabHilfbox%
8233 \unhbox\tabhilfbox\hskip\tabelskip%
8234 \unhbox\tabHilfbox}%
8235
8236 %

```

\EDTABtext (was \TABtext)

```

8237 \def\EDTABtext #1|#2|{%
8238 \setbox\tabhilfbox=\hbox{#1}%
8239 \setbox\tabHilfbox=\hbox{#2}%
8240 \advance\tabelskip -\wd\tabhilfbox%
8241 \advance\tabelskip -\wd\tabHilfbox%
8242 \unhbox\tabhilfbox\hskip\tabelskip%
8243 \unhbox\tabHilfbox}%
8244 %

```

\tabhilfbox Further helpers.

\tabHilfbox

```

8245 \newbox\tabhilfbox
8246 \newbox\tabHilfbox
8247
8248 %

```

XXX.2.4 Environments

edarrayl edarrayc edarrayr The ‘environment’ forms for \ltab, \ctab and \rtab.

```

8249 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{\}
8250 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{\}
8251 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{\}
8252
8253 %

```

edtabularl edtabularc edtabularr The ‘environment’ forms for \ltabtext, \ctabtext and \rtabtext.


```

8254 \newenvironment{edtabularl}{\l@ddcollect@body\ltabtext}{\}
8255 \newenvironment{edtabularc}{\l@ddcollect@body\ctabtext}{\}
8256 \newenvironment{edtabularr}{\l@ddcollect@body\rtabtext}{\}
8257
8258 %

```

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
8259 \newcommand{\initnumbering@quote}{
8260 \ifnoquotation@else
8261 \renewcommand{\quotation}{\par\leavevmode%
8262 \parindent=1.5em%
8263 \skipnumbering%
8264 \ifautopar%
8265 \vskip-\parskip%
8266 \else%
8267 \vskip\topsep%
8268 \fi%
8269 \global\leftskip=\leftmargin%
8270 \global\rightskip=\leftmargin%
8271 }
8272 \renewcommand{\endquotation}{\par%
8273 \global\leftskip=0pt%
8274 \global\rightskip=0pt%
8275 \leavevmode%
8276 \skipnumbering%
8277 \ifautopar%
8278 \vskip-\parskip%
8279 \else%
8280 \vskip\topsep%
8281 \fi%
8282 }
8283 \renewcommand{\quote}{\par\leavevmode%
8284 \parindent=0pt%
8285 \skipnumbering%
8286 \ifautopar%
8287 \vskip-\parskip%
8288 \else%
8289 \vskip\topsep%
8290 \fi%
8291 \global\leftskip=\leftmargin%
8292 \global\rightskip=\leftmargin%
8293 }

```

```

8294 \renewcommand{\endquote}{\par%
8295 \global\leftskip=0pt%
8296 \global\rightskip=0pt%
8297 \leavevmode%
8298 \skipnumbering%
8299 \ifautopar%
8300 \vskip-\parskip%
8301 \else%
8302 \vskip\topsep%
8303 \fi%
8304 }
8305 \fi
8306 }
8307 %

```

XXXII Section's title commands

XXXII.1 Commands to disable some feature

\ledsectnotoc The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```

8308 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
8309 %

```

\ledsectnomark The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```

8310 \newcommand{\ledsectnomark}{%
8311 \let\chaptermark\@gobble%
8312 \let\sectionmark\@gobble%
8313 \let\subsectionmark\@gobble%
8314 }
8315 %

```

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 \LaTeX is run a other time, this file is read. That:
 - Adds the `pstart` number to a list of `pstarts` where a sectioning command is used.
 - Defines a command, the name of which contains the `pstart` number, and which calls the normal \LaTeX sectioning command.
4. This last command is called when the `pstart` is effectively printed.

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).

```
8316 \catcode`\#=12
8317 \notbool{@noeled@sec}{%
8318 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
8319 \ifl@dmemoir
8320 \newcommand\beforeeledchapter{%
8321 \clearforchapter%
8322 }
8323 \else
8324 \newcommand\beforeeledchapter{%
8325 \if@openright%
8326 \cleardoublepage%
8327 \else%
8328 \clearpage%
8329 \fi%
8330 }
8331 \fi
8332 %
```

XXXII.4 Auxiliary commands

`\print@leftmargin@eledsection` and `\print@rightmargin@eledsection` are added by `reledmac` inside the code of sectioning command, in order to affix lines numbers. They include tests for RTL languages.

```
8333 \def\print@rightmargin@eledsection{%
8334 \if@eled@sectioning%
8335 \begingroup%
```

```

8336 \if@RTL%
8337 \let\llap\rlap%
8338 \let\leftlinenum\rightlinenum%
8339 \let\leftlinenumR\rightlinenumR%
8340 \let\l@drd@ta\l@dld@ta%
8341 \let\l@drsn@te\l@dlsn@te%
8342 \fi%
8343 \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
8344 \endgroup%
8345 \fi%
8346 }%
8347
8348 \def\print@leftmargin@eledsection{%
8349 \if@eled@sectioning%
8350 \leavevmode%
8351 \begingroup%
8352 \if@RTL%
8353 \let\rlap\llap%
8354 \let\rightlinenum\leftlinenum%
8355 \let\rightlinenumR\leftlinenumR%
8356 \let\l@dld@ta\l@drd@ta%
8357 \let\l@dlsn@te\l@drsn@te%
8358 \fi%
8359 \l@dld@ta\csuse{LR}{\l@dlsn@te}%
8360 \endgroup%
8361 \fi%
8362 }%
8363
8364 %

```

XXXII.5 Patching standard commands

`\M@ssect` We have to patch `LATEX`, `book` and `memoir` sectioning commands in order to:

`\@mem@old@ssect`

`\@makechapterhead`

`\@makechapterhead`

`\@makeschapterhead`

`\@ssect`

`\@ssect`

- 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.

```

8365 \AtBeginDocument{%
8366
8367
8368 \pretocmd{\M@ssect}
8369 {\let\old@edtext=\edtext%
8370 \let\edtext=\dummy@edtext@showlemma%
8371 }

```

```

8372 {}
8373 {}
8374
8375 \apptocmd{\M@sect}
8376 {\let\edtext=\old@edtext}
8377 {}
8378 {}
8379
8380 \patchcmd{\M@sect}
8381 { #9}
8382 { #9%
8383 \print@rightmargin@eledsection%
8384 }
8385 {}
8386 {}
8387
8388 \patchcmd{\M@sect}
8389 {\hskip #3\relax}
8390 {\hskip #3\relax%
8391 \print@leftmargin@eledsection%
8392 }
8393 {}
8394 {}
8395
8396 \patchcmd{\@mem@old@ssect}
8397 {#5}
8398 {#5%
8399 \print@leftmargin@eledsection%
8400 }
8401 {}
8402 {}
8403
8404 \patchcmd{\@mem@old@ssect}
8405 {\hskip #1}
8406 {\hskip #1%
8407 \print@rightmargin@eledsection%
8408 }
8409 {}
8410 {}
8411
8412
8413
8414 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8415 \if@eled@sectioning\else%
8416 \ifl@dprintingpages\else%
8417 \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a
\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
8418 \fi%

```

```

8419 \fi%
8420 }
8421 {}
8422 {}
8423
8424 \patchcmd{\@makechapterhead}
8425   {#1}
8426   {\print@leftmargin@eledsection%
8427     #1%
8428     \print@rightmargin@eledsection%
8429   }
8430 {}
8431 {}
8432
8433 \patchcmd{\@makechapterhead}% For BIDI
8434   {\if@RTL\raggedleft\else\raggedright\fi}%
8435   {\if@eled@sectioning\else%
8436     \if@RTL\raggedleft\else\raggedright\fi%
8437   \fi%
8438   }%
8439 {}%
8440 {}%
8441
8442 \patchcmd{\@makeschapterhead}
8443   {#1}
8444   {\print@leftmargin@eledsection%
8445     #1%
8446     \print@rightmargin@eledsection%
8447   }
8448 {}
8449 {}
8450
8451 \pretocmd{\@sect}
8452   {\let\old@edtext=\edtext
8453     \let\edtext=\dummy@edtext@showlemma%
8454   }
8455 {}
8456 {}
8457
8458 \apptocmd{\@sect}
8459   {\let\edtext=\old@edtext}
8460 {}
8461 {}
8462
8463 \pretocmd{\@ssect}
8464   {\let\old@edtext=\edtext%
8465     \let\edtext=\dummy@edtext@showlemma%
8466   }
8467 {}
8468 {}

```

```

8469 \apptocmd{\@sect}
8470   {\let\edtext=\old@edtext}
8471   {}
8472   {}
8473   {}
8474
8475 %

```

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.

```

8476 \@ifpackageloaded{nameref}{
8477
8478   \patchcmd{\NR@sect}
8479     {#8}
8480     {#8%
8481       \print@rightmargin@eledsection%
8482     }
8483     {}
8484     {}
8485
8486   \patchcmd{\NR@sect}
8487     {\hskip #3\relax}
8488     {\hskip #3\relax%
8489       \print@leftmargin@eledsection%
8490     }
8491     {}
8492     {}
8493
8494   \patchcmd{\NR@ssect}
8495     {#5}
8496     {#5%
8497       \print@rightmargin@eledsection%
8498     }
8499     {}
8500     {}
8501
8502   \patchcmd{\NR@ssect}
8503     {\hskip #1}
8504     {\hskip #1%
8505       \print@leftmargin@eledsection%
8506     }
8507     {}
8508     {}
8509   }%
8510   {
8511     \patchcmd{\@sect}
8512       {#8}
8513       {#8%
8514         \print@rightmargin@eledsection%
8515       }

```

```

8516 {}
8517 {}
8518
8519 \patchcmd{\@sect}
8520 {\hskip #3\relax}
8521 {\hskip #3\relax%
8522 \print@leftmargin@eledsection%
8523 }
8524 {}
8525 {}
8526
8527 \patchcmd{\@ssect}
8528 {#5}
8529 {#5%
8530 \print@rightmargin@eledsection%
8531 }
8532 {}
8533 {}
8534
8535 \patchcmd{\@ssect}
8536 {\hskip #1}
8537 {\hskip #1%
8538 \print@leftmargin@eledsection%
8539 }
8540 {}
8541 {}
8542 }%
8543 }%
8544 %

```

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 `#`.

```

8545 {}}%
8546 \protect\catcode`\#=6 %Space NEEDS by \catcode
8547 %

```

\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.

```

8548 \AtBeginDocument{%
8549 \patchcmd{\chapter}{\clearforchapter}{%
8550 \if@eled@sectioning\else%
8551 \ifl@dprintingpages\else%
8552 \clearforchapter%
8553 \fi%
8554 \fi%
8555 }%
8556 {}%
8557 {}%

```



```

8558
8559 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8560   \if@eled@sectioning\else%
8561   \ifl@dprintingpages%
8562     \endgraf%
8563   \else%
8564     \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
8565   \fi%
8566 \fi%
8567 }%
8568 {}%
8569 {}%
8570 }%
8571 %

```

\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.

```

8572 \newif\if@eled@sectioning%
8573 %

```

We reopen a new \notbool{@noeled@sec} statement, as we will define the \elesection commands.

```

8574 \notbool{@noeled@sec}{%
8575 %

```

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.

```

8576 \newwrite\eled@sectioning@out
8577 %

```

\eledchapter **\eledsection** And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsubsection
\eledsubsubsection
8578 \newcommand{\eledchapter}[2] [] {%
8579   #2%
8580   \ifl@Rcol%
8581     \immediate\write\eled@sectioningR@out{%
8582       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
8583     }%
8584   \else%
8585     \immediate\write\eled@sectioning@out{%

```

```

8586     \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
8587   }%
8588   \fi%
8589 }
8590
8591 \newcommand{\eledsection}[2] [] {%
8592   #2%
8593   \ifledRcol%
8594     \immediate\write\eled@sectioningR@out{%
8595       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
8596     }%
8597   \else%
8598     \immediate\write\eled@sectioning@out{%
8599       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
8600     }%
8601   \fi%
8602 }
8603
8604 \newcommand{\eledsubsection}[2] [] {%
8605   #2%
8606   \ifledRcol%
8607     \immediate\write\eled@sectioningR@out{%
8608       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
8609     }%
8610   \else%
8611     \immediate\write\eled@sectioning@out{%
8612       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
8613     }%
8614   \fi%
8615 }
8616 \newcommand{\eledsubsubsection}[2] [] {%
8617   #2%
8618   \ifledRcol%
8619     \immediate\write\eled@sectioningR@out{%
8620       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
8621     }-}{R}
8622   }%
8623   \else%
8624     \immediate\write\eled@sectioning@out{%
8625       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}
8626     }-}{-}
8627   }%
8628   \fi%
8629 }
8630 \WithSuffix\newcommand{\eledchapter*}[2] [] {%
8631   #2%
8632   \ifledRcol%

```

```

8633 \immediate\write\eled@sectioningR@out{%
8634 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8635 }%
8636 \else%
8637 \immediate\write\eled@sectioning@out{%
8638 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{L}
8639 }%
8640 \fi%
8641 }
8642
8643 \WithSuffix\newcommand\eledsection*[2][]{%
8644 #2%
8645 \ifledRcol%
8646 \immediate\write\eled@sectioningR@out{%
8647 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8648 }%
8649 \else%
8650 \immediate\write\eled@sectioning@out{%
8651 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{L}
8652 }%
8653 \fi%
8654 }
8655
8656 \WithSuffix\newcommand\eledsubsection*[2][]{%
8657 #2%
8658 \ifledRcol%
8659 \immediate\write\eled@sectioningR@out{%
8660 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8661 }%
8662 \else%
8663 \immediate\write\eled@sectioning@out{%
8664 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}
8665 }{*}{L}
8666 }%
8667 \fi%
8668 }
8669
8670 \WithSuffix\newcommand\eledsubsubsection*[2][]{%
8671 #2%
8672 \ifledRcol%
8673 \immediate\write\eled@sectioningR@out{%
8674 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
8675 }{*}{R}
8676 }%
8677 \else%
8678 \immediate\write\eled@sectioning@out{%
8679 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}
8680 }{*}{L}
8681 }%

```

```

8679 \fi%
8680 }
8681 %

```

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.

```

8682 \def\eled@chapter#1#2#3#4#5{%
8683     \ifstrempy{#4}%
8684     {%
8685         \ifstrempy{#1}%
8686         {%
8687             \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}%
8688             \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%
8689             }%Need for \pairs, because of using parbox.
8690             {%
8691                 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
8692                 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}}%Need for \pairs, because of using parbox.
8693             }%
8694             }%
8695             {%
8696                 \ifstrempy{#1}%
8697                 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
8698                 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*[#1]{#2}}}%Bug in LaTeX!
8699             }%
8700             \listcsgadd{eled@sections#5@@}{#3}%
8701             }
8702 \def\eled@section#1#2#3#4#5{%
8703     \ifstrempy{#4}%
8704     {\ifstrempy{#1}%
8705         {%
8706             \csgdef{eled@sectioning@#3#5}{\section{#2}}%

```

```

8707 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
      {#2}}%Need for \pairs, because of using parbox.
8708 }%
8709 {%
8710 \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
8711 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
      {#1}}%Need for \pairs, because of using parbox.
8712 }%
8713 }%
8714 {\ifstrepty{#1}%
8715   {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
8716   {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
8717 }
8718 \listcsadd{eled@sections#5@@}{#3}%
8719 }
8720 \def\eled@subsection#1#2#3#4#5{%
8721   \ifstrepty{#4}%
8722     {\ifstrepty{#1}%
8723       {%
8724         \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
8725         \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)
8726       }%
8727       {%
8728         \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
8729         \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)
8730       }%
8731     }%
8732     {\ifstrepty{#1}%
8733       {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
8734       {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
8735     }
8736   \listcsadd{eled@sections#5@@}{#3}%
8737 }
8738 \def\eled@subsubsection#1#2#3#4#5{%
8739   \ifstrepty{#4}%
8740     {\ifstrepty{#1}%
8741       {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
8742       {\csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
8743     }%
8744     {\ifstrepty{#1}%
8745       {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
8746       {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in
      LaTeX!
8747     }
8748   \listcsadd{eled@sections#5@@}{#3}%
8749 }

```

```
8750
8751 %
```

End of the conditional test about noeledsec option.

```
8752 }{}
8753 %
```

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.

```
8754 \def\normal@page@break{}
8755 %
```

`\l@prev@pb` The `\l@prev@pb` macro is a etoolbox list, which contains the lines in which page breaks occur (before or after). The `\l@prev@nopb` macro is a etoolbox list, which contains the lines with NO page break before or after.

```
8756 \def\l@prev@pb{}
8757 \def\l@prev@nopb{}
8758 %
```

`\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.

```
8759 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
8760 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
8761 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
8762 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
8763 %
```

`\led@pb` The `\led@pb` adds the absolute line number in the `\l@prev@pb` list. The `\led@pbnum` adds the argument in the `\l@prev@pb` list. The `\led@nopb` adds the absolute line number in the `\l@prev@nopb` list. The `\led@nopbnum` adds the argument in the `\l@prev@nopb` list.

```

8764 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
8765 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
8766 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
8767 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
8768 %

```

\ledpbsetting The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is `before`.

\led@pb@setting

```

8769 \def\led@pb@setting{before}
8770 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
8771 %

```

\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`.

\led@check@nopb

```

8772 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}{}}
8773 \newcommand{\led@check@nopb}{%
8774   \IfStrEq{\led@pb@setting}{before}{%
8775     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
8776       {\numdef{\abs@prevline}{\the\absline@num-1}}%
8777       \xifinlist{\abs@prevline}{\normal@page@break}%
8778       {\nopagebreak[4]\enlargethispage{\baselineskip}}%
8779     }%
8780   }%
8781 }%
8782 \IfStrEq{\led@pb@setting}{after}{%
8783   \xifinlist{\the\absline@num}{\l@prev@nopb}{%
8784     \xifinlist{\the\absline@num}{\normal@page@break}%
8785     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
8786   }%
8787 }%
8788 }%
8789   {}%
8790 {}%
8791 {}%
8792 }
8793 %

```

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.

```

8794 \newcommand{\check@pb@in@verse}{%
8795   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
8796   \ifnum\page@num=\last@page@num\else%If we have change page
8797   \IfStrEq{\led@pb@setting}{before}{%
8798     \numgdef{\abs@line@verse}{\the\absline@num-1}%
8799     \ledpbnum{\abs@line@verse}%
8800   }{}%
8801   \IfStrEq{\led@pb@setting}{after}{%
8802     \numgdef{\abs@line@verse}{\the\absline@num-1}%
8803     \lednopbnum{\abs@line@verse}%
8804   }{}%
8805   \fi%
8806 \fi\fi\fi%
8807 }
8808 %

```

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

```

8809 \def\Hy@raisedlink@left#1{%
8810   \ifvmode
8811     #1%
8812   \else
8813     \Hy@SaveSpaceFactor
8814     \llap{\smash{%
8815       \begingroup
8816         \let\HyperRaiseLinkLength\@tempdima
8817         \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
8818         \HyperRaiseLinkHook
8819       \expandafter\endgroup

```

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-coupure-de-ligne/801>.


```

8820 \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
8821 \Hy@RestoreSpaceFactor
8822 #1%
8823 \Hy@SaveSpaceFactor
8824 }%
8825 }}%
8826 \Hy@RestoreSpaceFactor
8827 \penalty\@M\hskip\z@\relax
8828 \fi
8829 }
8830 %

```

XXXVI Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

8831 \ifeledmaccompat@%
8832
8833 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
8834 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
8835 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
8836 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
8837
8838 \unless\ifnocritical@
8839 \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
8840 \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
8841 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
8842 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
8843 \let\hsizetwocol\Xhsizetwocol
8844 \let\hsizethreecol\Xhsizethreecol
8845 \let\bhookXnote\Xbhooknote
8846 \let\boxsymlinenum\Xboxsymlinenum
8847 \let\symlinenum\Xsymlinenum
8848 \let\beforenumberinfootnote\Xbeforenumber
8849 \let\afternumberinfootnote\Xafternumber
8850 \let\beforeXsymlinenum\XbeforeXsymlinenum
8851 \let\afterXsymlinenum\XafterXsymlinenum
8852 \let\inplaceofnumber\Xinplaceofnumber
8853 \let\Xlemmaseparator\lemmaseparator
8854 \let\afterlemmaseparator\Xafterlemmaseparator
8855 \let\beforelemmaseparator\Xbeforelemmaseparator
8856 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
8857 \let\txbeforeXnotes\Xtxbeforenotes
8858 \let\afterXrule\Xafterrule
8859 \let\numberonlyfirstinline\Xnumberonlyfirstinline
8860 \let\numberonlyfirstintwolines\Xnumberonlyfirstintwolines
8861 \let\nonumberinfootnote\Xnonumberinfootnote
8862 \let\pstartinfootnote\Xpstart
8863 \let\pstartinfootnoteeverytime\Xpstarteverytime

```

```

8864 \let\onlyXpstart\Xonlypstart
8865 \let\Xnonnumberinfootnote\Xnonnumber
8866 \let\Xnonbreakableafternumber\Xnonbreakableafternumber
8867 \let\XmaxhXnotes\Xmaxhnotes
8868 \let\XbeforeXnotes\Xbeforenotes
8869 \let\Xboxlinenum\Xboxlinenum
8870 \let\Xboxlinenumalign\Xboxlinenumalign
8871 \let\Xboxstartlinenum\Xboxstartlinenum
8872 \let\Xboxendlinenum\Xboxendlinenum
8873 \let\Xtwolines\Xtwolines
8874 \let\Xmorethantwolines\Xmorethantwolines
8875 \let\Xtwolinesbutnotmore\Xtwolinesbutnotmore
8876 \let\Xtwolinesonlyinsamepage\Xtwolinesonlyinsamepage
8877 \fi
8878
8879 \unless\ifnofamiliar@
8880 \let\XnotesXwidthliketwocolumns\XnoteswidthliketwocolumnsX
8881 \fi
8882 \newcommandX{\parafootsep}[2][1,usedefault]{%
8883 \Xparafootsep[#1]{#2}%
8884 \parafootsepX[#1]{#2}
8885 }%
8886
8887 \newcommandX{\afternote}[2][1,usedefault]{%
8888 \Xafternote[#1]{#2}%
8889 \afternoteX[#1]{#2}%
8890 }%
8891
8892 \unless\ifnoend@
8893 \let\XendXtwolines\Xendtwolines
8894 \let\XendXmorethantwolines\Xendmorethantwolines
8895 \let\XhookXendnote\Xendhooknote
8896 \let\XboxXendlinenum\Xendboxlinenum%
8897 \let\XboxXendlinenumalign\Xendboxlinenumalign%
8898 \let\XboxXendstartlinenum\Xendboxstartlinenum%
8899 \let\XboxXendendlinenum\Xendboxendlinenum%
8900 \let\XendXlemmaseparator\Xendlemmaseparator
8901 \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
8902 \let\XendXafterlemmaseparator\Xendafterlemmaseparator
8903 \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
8904 \fi
8905
8906 \AtBeginDocument{%
8907 \ifdef\lineref{}\let\lineref\edlineref}%
8908 }%
8909
8910
8911 \fi%
8912 %

```

</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:

<code>I saw my friend \critext{Smith}</code>	1 I saw my friend
<code>\Afootnote{Jones C, D.}/</code>	2 Smith on Tuesday.
on Tuesday.	<u>2 Smith]</u> Jones C, D.

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

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

<code>\critext{I saw my friend</code>	1 I saw my friend
<code>\critext{Smith}{\Afootnote{Jones</code>	2 Smith on Tuesday.
<code>C, D.}/ on Tuesday.}</code>	<u>2 Smith]</u> Jones C, D.
<code>\Bfootnote{The date was</code>	<u>1–2 I saw my friend</u>
<code>July 16, 1954.}</code>	Smith on Tuesday.] The
<code>/</code>	date was July 16, 1954.

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

The second argument of the `\critext` macro, `⟨commands⟩`, is the same as the second argument to the `\edtext` macro.

It is possible to define aliases for `\critext`, which can be easier to type. You can make a single character substitute for `\critext` by saying this:

```
\catcode`\<=\active
```

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. 119 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. 33), 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. 34) 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 `stanzaindentrepitition` (cf. 9.3 p. 48). This bug had two consequences:

1. `stanzaindentrepitition` did not work when its value was greater than 2.
2. `stanzaindentrepitition` worked wrong when its value was equal to 2.

So, if you used `stanzaindentrepitition` with a value equal to 2, you had to change your `\setstanzaindent`s. Explanation:

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

This code, in versions prior to 1.5.1, made the first line have an indentation of 0, the second line of 1, the third verse of 0, the fourth verse of 1 and so forth.

But this code should have instead achieved quite the contrary: the first line would have an indentation of 1, the second line of 0, the third line of 1, the fourth line of 0 and so forth.

So version 1.5.1 corrected this bug. If you want to keep the former presentation, you must change:

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

to:

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

Appendix A.4 Migration to eledmac 1.12.0

The migration to eledmac 1.12.0 is easy:

- You must first delete all the auxiliary files, then compile your document three times as usual.
- If you have modified `\l@reg`, which is not advisable, you must rename it to `\@nl@reg`.

There is an additional problem. If you have put text into brackets just after `\pstart` or `\pend`, this text will be considered to be an optional argument of `\pstart` or `\pend` (see 5.2.3 p. 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. 64.

Appendix A.5 Migration to eledmac 17.1

This version changes the default setting of `\Xpstart`. Henceforth, `pstart` numbers will be printed in footnotes within the section of text where you have called `\numberpstarttrue`.

We do not see any reason to print them in the other sections. However, if you want to print the `pstart` numbers in all of the footnotes, whatever the section, without having to use `\numberpstarttrue`, you can use `\Xpstarteverytime`.

Appendix A.6 Migration to eledmac 1.21.0

Appendix A.6.1 `\Xledsetnormalparstuff` and `\ledsetnormalparstuffX`

The `\ledsetnormalparstuff` has been split into two different commands:

- `\Xledsetnormalparstuff` for critical notes;
- `\ledsetnormalparstuffX` for familiar notes.

Both commands can take an optional argument which is the series letter. If you have redefined `\ledsetnormalparstuff` or any of the commands which call them, you must change them accordingly.

Appendix A.6.2 Endnotes

In any case, delete the `.end` file before the next run.

The previous version of Eledmac had a bug: there were two spaces between the starting page number and the starting line number, but only one space between the ending page number and the ending line number.

As a matter of fact, a spurious space was added after the first `\printnpnum`. This spurious space has been deleted. However, if you want to keep the previous spurious space, you may load the package with the `oldprintnpnumspace` option.

If you have redefined `\endprint`, you must:

- Contact us and ask for the feature that required your hack, in order to avoid such a hack in the future.
- Use the new fifth argument.
- Add `\xdef\@currentseries{#4}` at the beginning of your own command.

Appendix A.7 Migration to eledmac 1.22.0

The `\ledinnote` command now takes a first optional argument, which is the label for the hyperreference. If you have redefined it, change your redefinition, and check whether you can avoid this redefinition by only redefining `\ledinnotemark`.

Appendix A.8 Migration to eledmac 1.23.0

You must delete the numbered auxiliary files before compiling with the new version of eledmac.

Appendix A.9 Migration from eledmac to reledmac

There are many changes in reledmac which require the user to make modifications.

Appendix A.9.1 Risk of ‘no room for a new’

The risk to obtain a ‘no room for a new something’ error is greater in reledmac than it is in eledmac. See 19.1.3 p. 66 in order to know how to limit it.

Appendix A.9.2 Multiple indices with memoir

Eledmac and ledmac used the specific indexing tools of the memoir class designed to produce multiple indices. However, eledmac could also use imakeidx or indextools tools independently of the memoir class. This system forced to maintain redundant code. Since reledmac, we use only the imakeidx or indextools tools.

Consequently: Users of memoir are invited to use indextool or imakeidx to produce multiple indices.

Appendix A.9.3 Deprecated commands and options

The table of deprecated commands and their alternatives follows. Note that the way some commands must be used may have changed. Please read the handbook.

<i>Deprecated command</i>	<i>Replaced with</i>
<code>\addfootins</code>	<code>\newseries</code>
<code>\addfootinsX</code>	<code>\newseries</code>
<code>\critext</code>	<code>\edtext</code>
<code>\falseverse</code>	<code>\newverse</code>
<code>\interparanoteglue</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\ledchapter</code>	<code>\eledchapter</code>
<code>\ledsection</code>	<code>\eledsection</code>
<code>\ledsetnormalparstuff</code>	<code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code>
<code>\ledsubsection</code>	<code>\eledsubsection</code>
<code>\ledsubsubsection</code>	<code>\eledsubsubsection</code>
<code>\noeledsec</code>	Package option <code>noeledsec</code>
<code>\noendnotes</code>	Package option <code>noendnotes</code>
<code>\pageparbreak</code>	<code>\ledpb</code>

The `ledsecnolinenumber` option has been removed, because it was related to deprecated commands.

The `oldprintnpnumspace` option has been removed too, because it was related to a historical bug. The `\usingedtext` and `\usingcritext` commands are also deprecated.

Appendix A.9.4 \renewcommand replaced by command

Many uses of \renewcommand have been replaced with uses of specific commands. Please read handbook about specific commands.

<i>Deprecated \renewcommand</i>	<i>Replaced with</i>
\@led@extranofeet	\newseries
\apprefprefixmore	\setapprefprefixmore
\apprefprefixsingle	\setapprefprefixsingle
\endstanzaextra	Optional argument of \&
\hangingsymbol	\sethangingsymbol
\ledfootinsdim	\Xmaxhnotes and \maxhnotesX
\parafootftmsep	\Xparafootsep and \parafootsepX
\notenumfont	\Xnotenumfont, \Xendnotenumfont and \notenumfontX
\notefontsetup	\Xnotefontsize, \Xendnotefontsize and \notefontsizeX
\sidenotesep	\setsidenotsep
\startstanzahook	Optional argument of \stanza
\symplinenum	\Xsymplinenum

Appendix A.9.5 Commands the names of which have been changed

In order to help the migration from eledmac to reledmac, you may load reledmac with eledmac-compat option. However, it is advised not to, and to change the command names themselves instead. In many cases, you use only a few of them, except the \footparagraph command.

<i>Old command</i>	<i>New command</i>
\footparagraph	\Xarrangement
\footnormal	\Xarrangement
\foottwocol	\Xarrangement
\footthreecol	\Xarrangement
\footparagraphX	\arrangementX
\footnormalX	\arrangementX
\foottwocolX	\arrangementX
\footthreecolX	\arrangementX
\afterlemmaseparator	\Xafterlemmaseparator
\afternote	\Xafternote and \afternoteX
\afternumberinfootnote	\Xafternumber
\afterXrule	\Xafterrule
\afterXsymplinenum	\Xaftersymplinenum
\beforelemmaseparator	\Xbeforelemmaseparator
\beforenumberinfootnote	\Xbeforenumber
\beforeXnotes	\Xbeforenotes
\beforeXsymplinenum	\Xbeforesymplinenum

<i>Old command</i>	<i>New command</i>
<code>\bhookXnote</code>	<code>\Xbhookendnote</code>
<code>\bhookXnote</code>	<code>\Xbhooknote</code>
<code>\boxendlinenum</code>	<code>\Xboxendlinenum</code>
<code>\boxlinenum</code>	<code>\Xboxlinenum</code>
<code>\boxlinenumalign</code>	<code>\Xboxlinenumalign</code>
<code>\boxstartlinenum</code>	<code>\Xboxstartlinenum</code>
<code>\boxsymlinenum</code>	<code>\Xboxsymlinenum</code>
<code>\boxXendlinenum</code>	<code>\Xendboxlinenum</code>
<code>\boxXendlinenumalign</code>	<code>\Xendboxlinenumalign</code>
<code>\boxXendstartlinenum</code>	<code>\boxXendstartlinenum</code>
<code>\letboxXendendlinenum</code>	<code>\Xendletboxendlinenum</code>
<code>\hsizetwocol</code>	<code>\Xhsizetwocol</code>
<code>\hsizethreecol</code>	<code>\Xhsizethreecol</code>
<code>\inplaceoflemmaseparator</code>	<code>\Xinplaceoflemmaseparator</code>
<code>\inplaceofnumber</code>	<code>\Xinplaceofnumber</code>
<code>\lemmaseparator</code>	<code>\Xlemmaseparator</code>
<code>\maxhXnotes</code>	<code>\Xmaxhnotes</code>
<code>\morethantwolines</code>	<code>\Xmorethantwolines</code>
<code>\nonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\notesXwidthliketwocolumns</code>	<code>\noteswidthliketwocolumnsX</code>
<code>\noXlemmaseparator</code>	<code>\Xnolemmaseparator</code>
<code>\numberonlyfirstinline</code>	<code>\Xnumberonlyfirstinline</code>
<code>\numberonlyfirstintwolines</code>	<code>\Xnumberonlyfirstintwolines</code>
<code>\nonbreakableafternumber</code>	<code>\Xnonbreakableafternumber</code>
<code>\onlyXpstart</code>	<code>\Xonlypstart</code>
<code>\parafootsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\pstartinfootnote</code>	<code>\Xpstart</code>
<code>\pstartinfootnoteeverytime</code>	<code>\Xpstarteverytime</code>
<code>\symlinenum</code>	<code>\Xsymlinenum</code>
<code>\twolines</code>	<code>\Xtwolines</code>
<code>\twolinesbutnotmore</code>	<code>\Xtwolinesbutnotmore</code>
<code>\twolinesonlyinsamepage</code>	<code>\Xtwolinesonlyinsamepage</code>
<code>\txtbeforeXnotes</code>	<code>\Xtxtbeforenotes</code>
<code>\XendXafterlemmaseparator</code>	<code>\Xendafterlemmaseparator</code>
<code>\XendXbeforelemmaseparator</code>	<code>\Xendbeforelemmaseparator</code>
<code>\XendXinplaceoflemmaseparator</code>	<code>\Xendinplaceoflemmaseparator</code>
<code>\XendXlemmaseparator</code>	<code>\Xendlemmaseparator</code>
<code>\XendXmorethantwolines</code>	<code>\Xendmorethantwolines</code>
<code>\XendXtwolines</code>	<code>\Xendtwolines</code>
<code>\Xnonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\lineref</code>	<code>\edlineref</code>

Appendix A.9.6 Endnotes

With *reledmac*, there is now one auxiliary file for every endnotes set (*.Aend*, *.Bend*, *.Cend* etc.). If you have overridden `\doendnotes` (which you would not have done) you must adapt your code.

Appendix A.9.7 Z Series

The ‘Z’ series of notes has been removed. Only five series are provided now by default: A, B, C, D, E.

Appendix A.9.8 Internal commands

Users who have overridden internal commands, which is wrong, must adapt according to the following. Or better, they should not override any of such commands and use *reledmac* options instead.

- If you have modified `\Xfootfmt`, note that the fourth argument is now mandatory.
- `\unvxh` has been replaced with `\Xunvxh` and `\unvxhX` with two mandatory arguments.

Appendix A.10 Migration to *reledmac* 2.1.0

Reledmac 2.1.0 fix some bugs when using `\Xbhooknote` and `\bhooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }.`
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` in the argument of `\Xbhooknote/\bhooknoteX`.

Appendix A.11 Migration to *reledmac* 2.1.3

Reledmac 2.1.3 fix an historical bug, (style in *ledmac* 0.7!) which doubled the space before the rules of paragraphed familiar footnotes. Consequently, if you use paragraphed familiar footnotes, you should maybe adapt it, playing with `\beforenotesX`.

Appendix A.12 Migration to *reledmac* 2.3.0

Before *reledmac* 2.3.0, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use optional argument of `\stanza` or `\endverse`.

Appendix A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{\langle...}\rangle}` to the use of `\Xlinerangeseparator` or `/` and `\Xendlinerangeseparator` (7.2.4 p. 36).

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. 40.

`\xlineref` does not print anymore the side flag (R for right side), because it is incompatible with numerical test. Use `\xflagref` to obtain it.

The `\printlines` and `\printendlines` commands take now an eighth argument, which is the side flag. It is strongly recommended to NEVER redefine these two commands and to use the setting commands instead (or to ask for new setting commands if the actual does not answer to your needs). However, if you have done it, just change your redefinition to have a new argument.

It is strongly recommended to stop redefining `\fullstop` and to use `\Xsublinesep` instead.

Appendix A.15 Migration to reledmac 2.7.0

`\SErefonlypage` (introduced in reledmac 2.5.0) added an parenthesis after the page number. This was just an error, linked to a bad imitation of `\SErefwithpage`. That has been deleted. And so, the `\XendafterpagenumberSErefonlypage` to set it was also deleted.

`\rigidbalance` is split to two new commands: `\Xrigidbalance` for critical footnotes and `\rigidbalanceX` for familiar footnotes. If you have redefined it — but why should you have ?—, you should split your single redefinition in two redefinitions.

Appendix A.16 Migration to reledmac 2.7.2

`\Xhsize` is already defined in the `floatrow` package. It becomes `\Xwidth`, and, consequently, `\hsizeX` becomes `\widthX`.

The ancient names are temporarily maintained as aliases.

Appendix A.17 Migration to reledmac 2.8.0

Reledmac 2.8.0 fix spurious indents for paragraphed critical and familiar footnotes in `ledgroup` and `minipage`. You can re-establish the indent with `\Xparinden` and `\parindentX`.

Appendix A.18 Migration to reledmac 2.13.1

Reledmac 2.5.0 added a bug, which makes the right flag to be printed on the right side of critical footnotes, even if not explicitly requested by using `\Xlineflag`.

Version 2.13.1 solves this issue. Please use `\Xlineflag` if you want to add the right flag.

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>	48
<code>\@EDROWFILL@</code>	1
<code>\@adv</code>	1
<code>\@advancestanzanumber</code>	1
<code>\@beforeinsertofthisedtext</code>	1
<code>\@doclearpage</code>	1
<code>\@doreinfeetX</code>	1
<code>\@edindex@fornote@</code>	1
<code>\@edindex@hyperref</code>	1
<code>\@edrowfill@</code>	1
<code>\@edtext@level</code>	1

\@emptytoks	1
\@endmsd	1
\@fnpos	1
\@footnotemark	1
\@footnotetext	1
\@getfirstseries	1
\@gobblefive	1
\@gobblefour	1
\@gobblethree	1
\@h	1
\@hangingsymbol	1
\@iiiminipage	1
\@insertstanza	1
\@k	1
\@l@dttempcnta	1
\@l@dttempcntb	1
\@lab	1
\@led@testifnofoot	1
\@lemma	1
\@line@num	1
\@lock	1
\@lopL	1
\@lopR	1
\@makechapterhead	1
\@makeschapterhead	1
\@mem@extranofeet	1
\@mem@old@ssect	1
\@mpfnpos	1
\@msd	1
\@msd@c	1
\@msd@options@iffullpage	1
\@msdata@list	1
\@nl	1
\@nl@reg	1
\@opXfeet	1
\@pend	1
\@pendR	1
\@ref	1
\@ref@reg	1
\@ref@reg@parse	1
\@sect	1
\@series	1
\@set	1
\@sidenotesep	1
\@ssect	1
\@startstanza	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1

\@xloop	1
\@xympar	1
CLASSarticle	66
CLASSbook	66, 332
CLASSmemoir	192, 252, 253, 285, 332, 352, 410, 414
CLASSscrbook	414
COMMAND*footnote	68
COMMAND\...@footnotemark...	195
COMMAND\...d@ta	143
COMMAND\<hook	
@<series	240
COMMAND\<hookname	
<pseudoseries	242
COMMAND\<type	
footfmt	182
COMMAND\@@line	173
COMMAND\@MM	160, 411
COMMAND\@Rlineflag	287, 411
COMMAND\@Serefprefix	264
COMMAND\@Serefprefixmore	264
COMMAND\@add@	322
COMMAND\@adv	104
COMMAND\@apprefprefixmore	264
COMMAND\@apprefprefixsingle	264
COMMAND\@beforeinsertofthisedtext	126
COMMAND\@bsphack	255
COMMAND\@doclearpage	253, 404, 414
COMMAND\@doreinfeetX	414
COMMAND\@dprintingcolumns	411
COMMAND\@edindex@hyperref	287, 288
COMMAND\@edtext@level	123
COMMAND\@esphack	255
COMMAND\@fnpos	213, 249
COMMAND\@footnotemark	192, 404, 414
COMMAND\@footnotetext	192, 193, 404
COMMAND\@gobble	121, 122, 235
COMMAND\@gobblefive	237, 413
COMMAND\@gobblefour	410
COMMAND\@gobblethree	403
COMMAND\@h	176
COMMAND\@hangingsymbol	292
COMMAND\@iiiminipage	276, 278, 403, 414
COMMAND\@iiiminpage	277
COMMAND\@l	409
COMMAND\@l@tempcnta	145, 147, 155
COMMAND\@l@tempcntb	147
COMMAND\@l@reg	409
COMMAND\@lab	101, 254, 257, 261, 403
COMMAND\@ldunboxmpfoot	279
COMMAND\@led@extranofeet	353

COMMAND\@ledinnote@command	283, 284
COMMAND\@lemma	125, 127
COMMAND\@lock	95, 292
COMMAND\@lopL	404
COMMAND\@lopR	404
COMMAND\@makecol	248, 249, 252, 414
COMMAND\@mpfnpos	213
COMMAND\@msd	300
COMMAND\@msd@c	300
COMMAND\@msd@options@iffullpage	306
COMMAND\@msdata@list	300, 301
COMMAND\@nl	101–104, 106, 115, 257, 403, 404
COMMAND\@nl@reg	102, 350, 404, 409
COMMAND\@opXfeet	404
COMMAND\@opfeetX	414
COMMAND\@opxtrafeeti	414
COMMAND\@page	103, 257
COMMAND\@pend	404
COMMAND\@pendR	404
COMMAND\@ref	101, 111–113, 116, 121
COMMAND\@ref@later	112, 116
COMMAND\@ref@reg	111, 404
COMMAND\@ref@reg@parsearg	112
COMMAND\@reinserts	248, 250, 252, 414
COMMAND\@secondoftwo	68
COMMAND\@sect	335
COMMAND\@series	238
COMMAND\@set	105
COMMAND\@sidenotesep	275
COMMAND\@stopmsd	301
COMMAND\@sw	112, 113, 129, 132
COMMAND\@tag	122, 123, 127
COMMAND\@tempcnta	81
COMMAND\@tempcntb	81
COMMAND\@toksa	87
COMMAND\@toksb	87
COMMAND\@xloop	156
COMMAND\@xympar	269, 414
COMMAND\Aendnote	15, 24
COMMAND\Afootfmt	159
COMMAND\Afootgroup	159
COMMAND\Afootnote	8, 15, 23, 24, 27, 124, 168, 191, 214, 230, 413
COMMAND\Afootstart	159
COMMAND\AtBeginDocument	252
COMMAND\AtEveryPend	18, 50, 138, 411, 412, 414
COMMAND\AtEveryPstart	18, 50, 411, 412, 414, 417
COMMAND\AtEveryStanza	50, 418
COMMAND\AtEveryStopStanza	50, 418
COMMAND\Bendnote	15, 23
COMMAND\Bfootnote	8, 15, 191, 214, 230

COMMAND\Centering	44
COMMAND\Cfootnote	191
COMMAND\Columns	82, 165
COMMAND\Dfootnote	191
COMMAND\Efootnote	191
COMMAND\Gls	60
COMMAND\Hy@raisedlink	344
COMMAND\Hy@raisedlink@left	344
COMMAND\LTR	43
COMMAND\NR	335
COMMAND\Pages	82, 250, 251
COMMAND\ProcessOptionsX	72
COMMAND\RL	42
COMMAND\RaggedLeft	44
COMMAND\RaggedRight	44
COMMAND\SEonlypage	262, 416
COMMAND\SEref	54–56, 262, 265, 417, 419
COMMAND\SErefonlypage	54–56, 356, 416
COMMAND\SErefwithpage	54–56, 262, 265, 356, 416, 418
COMMAND\Stanza	409
COMMAND\Waklam	323
COMMAND\X@doreinfeet	250, 414
COMMAND\XR@prefix	268
COMMAND\XR@test	268
COMMAND\XR@test@mac	268
COMMAND\XR@test@mac@test	268
COMMAND\XXXXXXfmt	349
COMMAND\XXXXXXfmt	349
COMMAND\Xafterlemmaseparator	41, 353
COMMAND\Xafternote	44, 45, 352, 353
COMMAND\Xafternumber	38, 39, 353
COMMAND\Xafterrule	46, 214, 353, 409, 412
COMMAND\Xaftersymlinenum	39, 353
COMMAND\Xarrangement	35, 45, 68, 160, 161, 240, 353
COMMAND\Xarrangement@footparagraph	166
COMMAND\Xarrangement@normal	161
COMMAND\Xarrangement@paragraph	166
COMMAND\Xbeforeinserting	43
COMMAND\Xbeforelemmaseparator	40, 353
COMMAND\Xbeforenotes	45, 214, 353, 409, 412
COMMAND\Xbeforenumber	36, 38, 39, 353
COMMAND\Xbeforesymlinenum	39, 353
COMMAND\Xbhookendnote	354
COMMAND\Xbhookgroup	45, 416, 417
COMMAND\Xbhooknote	43, 354, 355, 414, 415
COMMAND\Xboxendlinum	39, 40, 354, 413
COMMAND\Xboxlinenum	39, 40, 354
COMMAND\Xboxlinenumalign	39, 40, 354, 413
COMMAND\Xboxstartlinenum	39, 40, 354, 413
COMMAND\Xboxsymlinenum	39, 354

COMMAND\Xcolalign	43, 412
COMMAND\Xdo@feet	414, 419
COMMAND\Xend	237
COMMAND\XendXafterlemmaseparator	354
COMMAND\XendXbeforelemmaseparator	354
COMMAND\XendXinplaceoflemmaseparator	354
COMMAND\XendXlemmaseparator	354
COMMAND\XendXmorethantwolines	354
COMMAND\XendXtwolines	354
COMMAND\Xendafterenumber	38, 415
COMMAND\Xendafterlemmaseparator	41, 354
COMMAND\Xendafternote	47, 417
COMMAND\Xendafternumber	40
COMMAND\Xendafterpagenumbe	418
COMMAND\Xendafterpagenumber	40, 55
COMMAND\XendafterpagenumberSErefonlypage	356
COMMAND\Xendaftersymlinenum	39, 40, 415
COMMAND\Xendahookinplaceofnumber	40, 415
COMMAND\Xendahooklinenumber	40, 415
COMMAND\Xendbeforelemmaseparator	41, 354
COMMAND\Xendbeforelinenumber	40
COMMAND\Xendbeforenumber	38, 415
COMMAND\Xendbeforepagenumber	40, 55
COMMAND\XendbeforepagenumberSErefonlypage	55
COMMAND\Xendbeforesymlinenum	39, 40, 415
COMMAND\Xendbhookinplaceofnumber	40, 415
COMMAND\Xendbhooklinenumber	40, 415
COMMAND\Xendbhooknote	43
COMMAND\Xendboxendlinum	40, 413
COMMAND\Xendboxlinenum	40, 354, 411
COMMAND\Xendboxlinenumalign	40, 354, 413
COMMAND\Xendboxstartlinenum	40, 413
COMMAND\Xendboxsymlinenum	39, 415
COMMAND\Xendhangindent	43, 415, 417
COMMAND\Xendinplaceoflemmaseparator	24, 41, 354
COMMAND\Xendinplaceofnumber	39, 414
COMMAND\Xendinplaceofpagenumber	36, 419
COMMAND\Xendinsertsep@	221
COMMAND\Xendlemmadisablefontselection	42
COMMAND\Xendlemmafont	42, 416
COMMAND\Xendlemmaseparator	25, 41, 354
COMMAND\Xendletboxendlinum	354
COMMAND\Xendlineflag	55
COMMAND\Xendlineprefixmore	40, 55
COMMAND\Xendlineprefixsingle	40, 56
COMMAND\Xendlinerangeseparator	36, 56, 180, 356, 415
COMMAND\Xendmorethantwolines	24, 37, 56, 354, 412, 413
COMMAND\Xendnonumber	37, 414
COMMAND\Xendnote	217, 235–237, 412
COMMAND\Xendnotefontsize	42, 353

COMMAND\Xendnotenumfont	40, 41, 353
COMMAND\Xendnotes	220
COMMAND\Xendnumberonlyfirstinline	35, 415
COMMAND\Xendnumberonlyfirstintwoline	35, 415
COMMAND\Xendpagenumberonlyfirst	36, 419
COMMAND\Xendpagenumberonlyfirstifsingle	36, 419
COMMAND\Xendpagenumberonlyfirstintwo	36, 419
COMMAND\Xendparagraph	47, 409
COMMAND\Xendsep	47
COMMAND\Xendsublinesep	38, 56, 180
COMMAND\Xendsymlinenum	35, 415
COMMAND\Xendsympagenum	36, 419
COMMAND\Xendtwoline	24, 37, 56, 354, 412, 413
COMMAND\Xendtwolinebutnotmore	37, 56, 412, 413
COMMAND\Xendtwolineonlyinsamepage	37, 56, 412, 413
COMMAND\Xendwrapcontent	42, 418
COMMAND\Xendwraplemma	42, 418
COMMAND\Xfootfmt	355
COMMAND\Xfootgroup	165
COMMAND\Xfootins	164
COMMAND\Xfootnote	52, 58, 122, 349, 406, 410, 412, 416, 418
COMMAND\Xfootstarts	165
COMMAND\Xgroupbyline	45, 153, 190
COMMAND\Xgroupbylines	419
COMMAND\Xgroupbylineseparatetwoline	45
COMMAND\Xhangindent	43, 415
COMMAND\Xhsize	356, 416, 417
COMMAND\Xhsizethreecol	44, 46, 354
COMMAND\Xhsizetwocol	44, 46, 242, 354
COMMAND\Xinplaceoflemmaseparator	24, 41, 354
COMMAND\Xinplaceofnumber	39, 354, 412, 413
COMMAND\Xinsertparafootsep	171, 172
COMMAND\Xledsetnormalparstuff	351, 352, 412
COMMAND\Xlemmadisablefontselection	41
COMMAND\Xlemmafnt	42, 416
COMMAND\Xlemmaseparator	40, 41, 180, 243, 246, 248, 354
COMMAND\Xlineflag	55, 356, 357, 418
COMMAND\Xlinerangeseparator	36, 55, 180, 356, 415
COMMAND\Xmaxhnotes	46, 66, 68, 353, 354, 409, 411
COMMAND\Xmorethantwoline	24, 37, 55, 354, 411
COMMAND\Xnoindent	415
COMMAND\Xnolemmaseparator	41, 248, 354
COMMAND\Xnonbreakableafternumber	38, 354, 407
COMMAND\Xnonumber	37, 354
COMMAND\Xnonumberinfootnote	354
COMMAND\Xnotefontsize	42, 353
COMMAND\Xnotefontsize@<s>	171, 175, 176
COMMAND\Xnotenumfont	41, 353
COMMAND\Xnoteswidthliketwocolumns	47, 410
COMMAND\Xnumberonlyfirstinline	35, 36, 45, 98, 242, 243, 246, 354, 406, 411

COMMAND\Xnumberonlyfirstintwolines	35, 45, 354, 406
COMMAND\Xonlypstart	38, 354, 406, 411
COMMAND\Xpagelinesep	38, 419
COMMAND\Xparafootsep	44, 45, 98, 353, 354, 418
COMMAND\Xparafootsep@series	171
COMMAND\Xparinden	356
COMMAND\Xparindent	42, 412, 415, 417
COMMAND\Xprenotes	45, 214, 418
COMMAND\Xprenotes@	164, 214, 406
COMMAND\Xpstart	37, 38, 351, 354, 406, 411
COMMAND\Xpstarteverytime	38, 351, 354, 411
COMMAND\Xragged	44
COMMAND\Xrigidbalance	173, 356, 416
COMMAND\Xstanza	38, 50
COMMAND\Xstanzaseparator	38
COMMAND\Xsublinesep	21, 38, 55, 180, 356
COMMAND\Xsublinesepside	21, 38
COMMAND\Xsymlinenum	35, 44, 353, 354, 413
COMMAND\Xtoendnotes	25, 236
COMMAND\Xtwolines	24, 37, 55, 187, 188, 242, 354, 411
COMMAND\Xtwolinesappref	242
COMMAND\Xtwolinesbutnotmore	37, 55, 354, 412
COMMAND\Xtwolinesbutnotmoreappref	242
COMMAND\Xtwolinesonlyinsamepage	37, 55, 354, 412
COMMAND\Xtxtbeforenotes	45, 354, 418, 419
COMMAND\Xunvxh	168, 355
COMMAND\Xwidth	46, 356, 417
COMMAND\Xwrapcontent	42, 418
COMMAND\Xwraplemma	42, 43, 418
COMMAND\&	353
COMMAND\(\XXX)vfootnote	190
COMMAND\absline@num	95, 144
COMMAND\accent	121
COMMAND\actionlines@list	96, 146
COMMAND\actions@list	96
COMMAND\add@Xgroupbyline	153
COMMAND\add@inserts	96, 152, 153
COMMAND\add@inserts@next	152, 153
COMMAND\add@msd@	300
COMMAND\add@msdata	300, 301
COMMAND\add@msdata@firstlineofpage	303
COMMAND\add@msddata	300
COMMAND\add@penalties	144, 154
COMMAND\addcontentsline	330
COMMAND\addfootins	349, 352
COMMAND\addfootinsX	349, 352
COMMAND\addtoenotes	236
COMMAND\advancelabel@refs	256
COMMAND\advanceline	21, 22, 97, 104, 117, 414
COMMAND\affixlin@num	275

COMMAND\affixline@num	147, 150, 151, 404
COMMAND\affixpstart@num	151
COMMAND\afterXrule	353
COMMAND\afterXsymlinenum	353
COMMAND\afterenumber	38
COMMAND\aftergroup	120, 125
COMMAND\afterlemmaseparator	353
COMMAND\afternote	353
COMMAND\afternoteX	44, 352, 353
COMMAND\afternumberinfootnote	353
COMMAND\afterruleX	46, 409, 412
COMMAND\applabel	54, 258, 259, 412, 418
COMMAND\appref	52, 54–56, 262, 265, 416, 417
COMMAND\apprefprefixmore	54, 353
COMMAND\apprefprefixsingle	54, 353
COMMAND\apprefwithpage	54–56, 262, 265, 413, 416
COMMAND\arrangementX	35, 68, 195, 240, 353
COMMAND\arrangementX@normal	201
COMMAND\article	14
COMMAND\at@every@pend	138
COMMAND\autopar	17, 135, 138, 139, 211, 405, 407, 408, 412
COMMAND\ballast	67
COMMAND\ballast@count	144, 155
COMMAND\baselineskip	35, 167, 171
COMMAND\beforeXnotes	353
COMMAND\beforeXsymlinenum	353
COMMAND\beforeelectedchapter	10, 64, 331
COMMAND\beforeinsertingX	43
COMMAND\beforelemmaseparator	353
COMMAND\beforenotesX	45, 355, 408, 409, 412
COMMAND\beforenumberinfootnote	353
COMMAND\begin	307, 308
COMMAND\beginnumbering	16, 18, 19, 82, 83, 85, 94, 99, 114, 138, 217, 299, 406, 409, 413, 414, 419
COMMAND\bf	406
COMMAND\bfseries	41, 406
COMMAND\bhookXnote	354
COMMAND\bhookgroupX	45, 416
COMMAND\bhooknoteX	43, 355, 414, 415
COMMAND\body	292
COMMAND\bodyfootmarkA	32
COMMAND\book	14
COMMAND\boxXendlinenum	354
COMMAND\boxXendlinenumalign	354
COMMAND\boxXendstartlinenum	354
COMMAND\boxendlinenum	354
COMMAND\boxlinefootnote	184
COMMAND\boxlinenum	354
COMMAND\boxlinenumalign	354
COMMAND\boxstartlinenum	354
COMMAND\boxsymlinenum	354

COMMAND\break	35, 169
COMMAND\brokenpenalty	154
COMMAND\centering	44
COMMAND\ch@ck@l@ck	404
COMMAND\ch@cksub@l@ck	150, 404
COMMAND\chapter	63, 332, 336, 409, 412, 414, 418
COMMAND\chaptermark	330
COMMAND\check@pb@in@verse	344
COMMAND\colalignX	43, 412
COMMAND\collect@body	308
COMMAND\color	418
COMMAND\colorbox	68
COMMAND\columns	47
COMMAND\columnwidth	167, 410
COMMAND\command names	242
COMMAND\copyright	121
COMMAND\correct@Xfootins@box	411
COMMAND\correct@footinsX@box	411
COMMAND\count	174
COMMAND\critex	405
COMMAND\critext	127, 348, 349, 352
COMMAND\curname	73, 131
COMMAND\ctab	323, 324, 328
COMMAND\ctabtext	328
COMMAND\dcoll	317
COMMAND\def	70
COMMAND\detokenize	131
COMMAND\dimen	174
COMMAND\dimexpr	46
COMMAND\discretionary	168
COMMAND\displaywidowpenalty	154
COMMAND\do@Xfeet	249, 404, 414, 419
COMMAND\do@actions	144–146, 404
COMMAND\do@actions@fixedcode	404
COMMAND\do@actions@next	145, 146
COMMAND\do@ballast	144, 155
COMMAND\do@feet@custom@order	249
COMMAND\do@insidelinehook	407
COMMAND\do@line	96, 120, 137, 140, 143, 152, 154, 292, 404, 405, 407, 409
COMMAND\do@linehook	404
COMMAND\do@lockoff	97
COMMAND\do@lockon	97
COMMAND\dodoreintrafeet	403
COMMAND\doendnotes	25, 221, 355, 413
COMMAND\doendnotesbysection	25, 221, 237, 413
COMMAND\doennotes	420
COMMAND\doinsidelinehook	22, 410
COMMAND\dolinehook	22, 410
COMMAND\doreintrafeeti	414
COMMAND\doreintrafeetii	414

COMMAND\doxtrafeet	248, 403
COMMAND\doxtrafeeti	414
COMMAND\doxtrafeetii	414
COMMAND\dummy@ref	121
COMMAND\edaftertab	63, 323, 324
COMMAND\edatleft	62, 321
COMMAND\edatright	62, 63, 321
COMMAND\edbeforetab	63, 323
COMMAND\edfilldimen	322
COMMAND\edfont@info	127
COMMAND\edgls	60, 282
COMMAND\edgls...	417
COMMAND\edindex	58–60, 282, 283, 286, 288, 312, 407, 410, 411, 414, 415, 419
COMMAND\edindexlab	59, 60
COMMAND\edlabel	52–54, 56, 121, 254, 256, 257, 259, 260, 267, 282, 312, 344, 403, 406–408, 411, 416
COMMAND\edlabelE	54, 259
COMMAND\edlabelS	54, 259
COMMAND\edlabelSE	54
COMMAND\edlineref	52, 254, 354, 411, 413, 416, 420
COMMAND\edmakelabel	53, 267, 268
COMMAND\edpageref	52, 254, 259, 267
COMMAND\edrowfill	323
COMMAND\edtabcolsep	316
COMMAND\edtext	6, 23, 24, 26–29, 47, 52–54, 58, 61, 68, 95, 96, 111, 113, 116, 119–127, 129, 130, 132, 133, 258, 259, 262, 312, 313, 332, 344, 348, 349, 352, 403, 405, 407, 409–413, 418, 419
COMMAND\edtext@level	413
COMMAND\edtextlater	113
COMMAND\edvertdots	63, 322
COMMAND\edvertline	63, 322
COMMAND\elechapter	64
COMMAND\eled@sectioning@out	337
COMMAND\eledchapter	64, 352, 410, 414
COMMAND\eledchapter*	64
COMMAND\eledmac@error	403
COMMAND\eledsection	6, 15, 64, 121, 142, 331, 352, 412
COMMAND\eledsection*	64
COMMAND\eledsubsection	64, 352
COMMAND\eledsubsection*	64
COMMAND\eledsubsubsection	64, 352
COMMAND\eledsubsubsection*	64
COMMAND\eledxxx	10, 65, 337, 409
COMMAND\eledxxxx	330
COMMAND\eledsection	337
COMMAND\else	281, 331
COMMAND\empty	81, 148, 254
COMMAND\end	307, 308
COMMAND\end@lemmas	120
COMMAND\endashchar	180
COMMAND\endgraf	137, 170, 211
COMMAND\endlock	21, 97, 118, 297

COMMAND\endminipage	276, 277, 279, 403, 414
COMMAND\endmsdata	31
COMMAND\endnotes	412, 416
COMMAND\endnumbering	16, 19, 82–85, 404, 413, 419
COMMAND\endprint	217, 219, 237, 351
COMMAND\endstanzaextra	353
COMMAND\endsub	21, 97, 117
COMMAND\endverse	355
COMMAND\everypar	139
COMMAND\extensionchars	66, 82
COMMAND\externaldocument	56, 268
COMMAND\fix@l@cks	404
COMMAND>falseverse	352, 407, 409
COMMAND\fi	331
COMMAND\firstlinenum	19, 147, 405
COMMAND\firstsublinenum	19, 405
COMMAND\fix@page	102, 103, 404
COMMAND\flag@end	116, 126, 409
COMMAND\flag@end@later	116
COMMAND\flag@start	116, 126, 409, 410
COMMAND\flag@start@later	116
COMMAND\flagstanza	51
COMMAND\floatingpenalty	160, 411
COMMAND\flush@notes	155, 156
COMMAND\fnpos	33, 213, 408, 419
COMMAND\footfmt	159, 162
COMMAND\footfmt...	196
COMMAND\footfootmarkA	32
COMMAND\footfudgefactor	169
COMMAND\footfudgefiddle	68, 167, 403
COMMAND\footgroup	159
COMMAND\footins	164
COMMAND\footnormal	241, 353, 403
COMMAND\footnormalX	353
COMMAND\footnote	32, 67, 191–193, 350, 404
COMMAND\footnote@lang	180
COMMAND\footnoteA	16, 32
COMMAND\footnoteB	16
COMMAND\footnoteC	23
COMMAND\footnoteE	32
COMMAND\footnoteX	8, 233, 235, 418
COMMAND\footnoteX@reading	235
COMMAND\footnoteXmk	248
COMMAND\footnote⟨X⟩	122
COMMAND\footnotelang@lua	158
COMMAND\footnotelang@poly	158
COMMAND\footnoteoption@	157, 415
COMMAND\footnoterule	174
COMMAND\footnotesize	42
COMMAND\footparagraph	167, 241, 353, 409

COMMAND\footparagraphX	206, 353, 409
COMMAND\footsplitskips	404, 411
COMMAND\footstart	159, 164, 174
COMMAND\footstrut	170
COMMAND\footthreecol	353
COMMAND\footthreecolX	353, 413
COMMAND\foottwocol	353
COMMAND\foottwocolX	353, 413
COMMAND\foreignlanguage	42
COMMAND\fullstop	356
COMMAND\get@edindex@hyperref	287
COMMAND\get@edindex@ledinnote@command	283
COMMAND\get@fnmark	193
COMMAND\get@index@command	408
COMMAND\get@linelistfile	404
COMMAND\get@thisfootnote	200
COMMAND\getline@num	144, 145
COMMAND\gl@p	87
COMMAND\global	101
COMMAND\globaldefs	101
COMMAND\gls	60, 290
COMMAND\hangindentX	43, 412, 415
COMMAND\hangingsymbol	353, 405
COMMAND\hbox	168
COMMAND\hfill	408
COMMAND\hidenumbering	22, 109, 412
COMMAND\hidenumberingonleftpage	22, 109, 418
COMMAND\hidenumberingonrightpage	22, 109, 418
COMMAND\hline	61
COMMAND\hrulefill	323
COMMAND\hsize	35, 164, 167, 169, 175, 178, 212, 404, 410
COMMAND\hsizeX	356, 416, 417
COMMAND\hsizethreecol	354
COMMAND\hsizethreecolX	44, 46
COMMAND\hsizetwocol	354
COMMAND\hsizetwocolX	44, 46
COMMAND\hyperlinkR	287
COMMAND\hyperlinkformat	286
COMMAND\hyperlinkformatR	287
COMMAND\if@RTL	74
COMMAND\if@edtext@	410, 413
COMMAND\if@eled@sectioning	337
COMMAND\if@firstlineofpage	74
COMMAND\if@firstlineofpageR	74
COMMAND\if@msd@options@fullpage	306
COMMAND\if@msdata@insertedfrompreviouspage	303
COMMAND\if@nobreak	138
COMMAND\if@noneed@Footnote	116
COMMAND\ifXnote@	82
COMMAND\ifbypage@	88

COMMAND\ifbypage@R	88
COMMAND\ifbypstart@	88
COMMAND\ifbypstart@R	88
COMMAND\iffirst@linenum@out@	114
COMMAND\ifindtl@innote	82
COMMAND\ifindtl@notenumber	82
COMMAND\ifinserthangingsymbol	292
COMMAND\ifinstanza	292
COMMAND\ifstwofollowinglines	188
COMMAND\ifl@d@Xmorethantwolines	185, 411
COMMAND\ifl@d@Xtwolines	185
COMMAND\ifl@d@dash	185
COMMAND\ifl@d@elin	185
COMMAND\ifl@d@esl	185
COMMAND\ifl@d@pnum	185
COMMAND\ifl@d@ssub	185
COMMAND\ifl@dend@X	235
COMMAND\ifl@dmemoir	403
COMMAND\ifl@dpaging	410
COMMAND\ifl@dpairing	81, 405
COMMAND\ifl@dprintingpages	411
COMMAND\ifl@dskipnumber	147
COMMAND\ifl@dstartendok	322
COMMAND\ifl@imakeidx	74
COMMAND\ifledRcol	82, 405
COMMAND\ifledRcol@	82, 409
COMMAND\iflemmacommand@	410
COMMAND\ifnoend@	221
COMMAND\ifnoedgroup@	281
COMMAND\ifnoteschanged@	98
COMMAND\ifnumberedpar@	135
COMMAND\ifnumbering	83, 85
COMMAND\ifnumberingR	82, 405
COMMAND\ifnumberline	126, 147
COMMAND\ifpst@rted	405
COMMAND\ifpst@rtedL	83
COMMAND\ifseriesbefore	239
COMMAND\ifstopmsdata@inserted@	299
COMMAND\ifsublines@	94, 107
COMMAND\iftrue	413
COMMAND\ifvmode	256
COMMAND\ifxxx	331
COMMAND\ignorespaces	124
COMMAND\imki@wrindexentry	74
COMMAND\immediate	114, 216
COMMAND\indent	18, 139, 355
COMMAND\index	289, 419
COMMAND\indtl@wrindexentry	74
COMMAND\initnumbering@quote	329, 414
COMMAND\initnumbering@reg	404

COMMAND\initnumbering@sectcmd	414
COMMAND\inplaceoflemmaseparator	354
COMMAND\inplaceofnumber	354
COMMAND\insert	126, 152, 159, 162, 176, 190, 196
COMMAND\insert@Xtxtbeforenotes	156, 176
COMMAND\insert@count	111, 116, 124
COMMAND\insert@countR	124
COMMAND\insert@msdata	300, 306
COMMAND\inserthangingsymbol	408
COMMAND\insertlines@list	96, 111
COMMAND\insertparafootsepX	210
COMMAND\inserts@list	120, 135, 152, 153, 168
COMMAND\interAfootnotelinepenalty	405
COMMAND\interfootnotelinepenalty	405
COMMAND\interlinepenalty	159
COMMAND\interparanoteglua	352
COMMAND\justifying	43
COMMAND\l@advance@parledegroupp@beforenormalnotes	414
COMMAND\l@d@@wrindexhyp	410
COMMAND\l@d@add	128
COMMAND\l@d@end	217, 235
COMMAND\l@d@nums	124, 126, 128, 185
COMMAND\l@d@section	217
COMMAND\l@d@set	106, 118
COMMAND\l@dampcount	313
COMMAND\l@dbfnote	193, 404
COMMAND\l@dcheckstartend	322
COMMAND\l@dchset@num	106
COMMAND\l@dcolcount	313, 314
COMMAND\l@dcollect@@body	307
COMMAND\l@dcollect@body	307
COMMAND\l@dcsnote	409
COMMAND\l@dcsnotetext	143, 273
COMMAND\l@dcsnotetext@l	143, 273
COMMAND\l@dcsnotetext@r	143, 273
COMMAND\l@ddodoreinxtrafeet	250, 403
COMMAND\l@ddoxtrafeet	250, 403
COMMAND\l@demptyd@ta	405
COMMAND\l@dend@close	216
COMMAND\l@dend@open	216
COMMAND\l@dend@stuff	217
COMMAND\l@denvbody	307
COMMAND\l@dfeetbeginmini	403
COMMAND\l@dfeetendmini	403
COMMAND\l@dgetline@margin	405
COMMAND\l@dgetlock@disp	405
COMMAND\l@dgetref@num	261
COMMAND\l@dgetsidenote@margin	269, 405
COMMAND\l@dgobbeloptarg	410
COMMAND\l@dgobblearg	410

COMMAND\l@dobbleoptarg	312
COMMAND\l@dlabel@parse	261
COMMAND\l@dld@ta	147, 149
COMMAND\l@dlp@rbox	274
COMMAND\l@dlsn@te	405
COMMAND\l@dlsnote	409
COMMAND\l@dmake@labels	256, 257, 268
COMMAND\l@dmake@labelsR	268
COMMAND\l@dnumstartsL	83, 405
COMMAND\l@dp@rsefootspec	185
COMMAND\l@dparsefootspec	185
COMMAND\l@dpush@begins	308
COMMAND\l@drd@ta	147, 149
COMMAND\l@dref@undefined	261
COMMAND\l@drsn@te	405
COMMAND\l@drsnote	409
COMMAND\l@dtabaddcols	322
COMMAND\l@dtabnoexpands	403
COMMAND\l@dumboxmpfoot	414
COMMAND\l@dunboxmpfoot	405
COMMAND\l@dzeropenalties	405, 410
COMMAND\l@pb	343
COMMAND\l@prev@nopb	342
COMMAND\l@prev@pb	342
COMMAND\l@reg	350
COMMAND\label	18, 54, 56, 59, 60, 255, 262
COMMAND\label@refs	254
COMMAND\labelstarttrue	18, 406
COMMAND\labelref@list	254, 257
COMMAND\language	168
COMMAND\last@page@num	404
COMMAND\lastbox	139
COMMAND\lastskip	117
COMMAND\latex@makecol	252
COMMAND\leavevmode	18, 139
COMMAND\led@check@nopb	343
COMMAND\led@check@pb	343
COMMAND\led@nopb	342, 344
COMMAND\led@nopbnum	342
COMMAND\led@pb	342, 344
COMMAND\led@pb@macro	343
COMMAND\led@pbnum	342
COMMAND\led@reinit@index@fornote	289
COMMAND\led@set@index@fornote	289
COMMAND\ledRflag	287
COMMAND\ledchapter	352, 407
COMMAND\ledfootinsdim	353
COMMAND\ledinnernote	56, 270, 409, 418
COMMAND\ledinnote	284, 351, 413
COMMAND\ledinnotemark	58, 351, 412

COMMAND\ledleftnote	57, 270
COMMAND\ledlinenum	93, 405
COMMAND\ledllfill	143
COMMAND\ledlsnotefontsetup	418
COMMAND\ledlsnotesep	57
COMMAND\ledlsnotewidth	57
COMMAND\lednopb	65, 342
COMMAND\lednopbinverse	343
COMMAND\lednopbinversetrue	50, 65
COMMAND\lednopbnum	342
COMMAND\ledouternote	56, 270, 409, 418
COMMAND\ledpb	65, 342, 352
COMMAND\ledpbnum	342
COMMAND\ledpbsetting	66, 343, 415
COMMAND\ledrightnote	57, 270
COMMAND\ledrsnotefontsetup	418
COMMAND\ledrsnotesep	57
COMMAND\ledrsnotewidth	57
COMMAND\ledsection	352
COMMAND\ledsectnomark	330
COMMAND\ledsectnotoc	330
COMMAND\ledsetnormalparstuff	351, 352, 412
COMMAND\ledsetnormalparstuff@common	211
COMMAND\ledsetnormalparstuffX	351, 352, 412
COMMAND\ledsidenote	56, 57, 270, 273
COMMAND\ledsubsection	352
COMMAND\ledsubsubsection	352
COMMAND\ledxxx	409
COMMAND\left	62
COMMAND\leftctab	323
COMMAND\leftheadline	93
COMMAND\leftlinenum	20, 93, 403, 405
COMMAND\leftltab	323
COMMAND\leftnoteupfalse	57
COMMAND\leftpstartnum	151
COMMAND\leftrtab	323
COMMAND\leftsidenote	273
COMMAND\leftskip	164, 167, 169
COMMAND\lemma	3, 24, 26–29, 119, 123, 124, 127, 129, 348, 405, 406, 413, 414, 416
COMMAND\lemmaseparator	354
COMMAND\let	27, 48, 122, 126, 235, 252, 297, 403
COMMAND\letboxXendendlinenum	354
COMMAND\line	173, 176
COMMAND\line@list	95, 112, 126
COMMAND\line@list@stuff	83, 99, 114, 403, 405
COMMAND\line@list@version	101
COMMAND\line@margin	89, 149, 269
COMMAND\line@num	94–96, 147, 403
COMMAND\line@set	128
COMMAND\lineation	20, 88

COMMAND\linebreak	35
COMMAND\linenum	24, 26, 27, 53, 54, 119, 128, 259, 262, 268, 348, 417
COMMAND\linenum@out	113, 114, 254, 257
COMMAND\linenumberlist	19, 20, 81, 148, 403
COMMAND\linenumberstyle	22, 93, 403, 420
COMMAND\linenumincrement	19, 405
COMMAND\linenummargin	20, 89, 269
COMMAND\linenumr@p	92, 403, 405
COMMAND\linenumrep	92, 93, 405
COMMAND\linenumsep	20, 57, 93, 270
COMMAND\linerangesep@	248
COMMAND\lineref	254, 260, 267, 354, 411
COMMAND\list@clear	87
COMMAND\list@clearing@reg	405
COMMAND\list@create	86, 87
COMMAND\lock@disp	91
COMMAND\lock@off	108
COMMAND\lock@on	107
COMMAND\lockdisp	21, 91
COMMAND\loop	156, 292
COMMAND\ltab	323, 324, 328
COMMAND\ltabtext	328
COMMAND\m@mmf@prepare	192
COMMAND\makeatletter	143
COMMAND\makehboxofhboxes	169, 171
COMMAND\makeindex	58, 286
COMMAND\makelabel	267
COMMAND\managestanza@modulo	294
COMMAND\marginpar	56, 67, 269, 404
COMMAND\marginparwidth	57, 270
COMMAND\markboth	143
COMMAND\mathchardef	293
COMMAND\maxXnotes	354
COMMAND\maxhnotesX	46, 68, 353, 408, 409, 411–413
COMMAND\maxlinesinpar@list	99
COMMAND\measurebody	325
COMMAND\measuredbody	325
COMMAND\memorybreak	19
COMMAND\morenoexpands	68, 69, 120, 121
COMMAND\morethantwolines	354
COMMAND\mpfnpos	33, 213, 408, 419
COMMAND\mpnormalfootgroup	404
COMMAND\mpnormalvfootnote	404
COMMAND\msdata	31, 298–300
COMMAND\multfootsep	32, 192
COMMAND\multiplefootnotemarker	192
COMMAND\musixtex	409
COMMAND\n@num	405, 412
COMMAND\n@num@ref	412
COMMAND\new@line	115, 404

COMMAND\new@series	122
COMMAND\newcommand	27, 70, 192, 257
COMMAND\newcommandx	27
COMMAND\newhookarg@specific	247
COMMAND\newhookcommand@series	242, 243, 412
COMMAND\newhookcommand@series@reload	243
COMMAND\newhookcommand@toggle@reload	243, 410
COMMAND\newhooktoggle@series	242, 243, 412
COMMAND\newhooktoggle@specific	247
COMMAND\newif	412
COMMAND\newline	35
COMMAND\newlinechar	236
COMMAND\newseries	33, 349, 352, 353
COMMAND\newseries@	227, 228, 240
COMMAND\newverse	50, 51, 352, 409
COMMAND\next	292
COMMAND\next@action	100
COMMAND\next@actionline	100
COMMAND\next@insert	153
COMMAND\nl@regR	102
COMMAND\no@expands	68, 122, 127, 235, 403
COMMAND\noXlmmaseparator	354
COMMAND\nobreak	184
COMMAND\nocritical	228
COMMAND\noeledsec	65, 352
COMMAND\noendnotes	352
COMMAND\noexpand	350
COMMAND\nofamiliar	245
COMMAND\noindent	18, 139, 415
COMMAND\noindentX	415
COMMAND\nomk@	248
COMMAND\nonbreakableafternumber	354
COMMAND\nonumberinfootnote	354
COMMAND\normal@footnotemarkX	196
COMMAND\normal@page@break	342
COMMAND\normal@pars	211
COMMAND\normalbfnoteX	405
COMMAND\normalbodyfootmarkX	196
COMMAND\normalfont	419
COMMAND\normalfootfmt	48, 163, 170, 180, 217, 417
COMMAND\normalfootfmtX	197
COMMAND\normalfootfootmarkX	197
COMMAND\normalfootgroup	165
COMMAND\normalfootgroupX	198
COMMAND\normalfootnoterule	160
COMMAND\normalfootstart	164, 167
COMMAND\normalfootstartX	197
COMMAND\normalvfootnote	162, 163
COMMAND\normalvfootnote@inserted	162, 163
COMMAND\normalvfootnoteX	196

COMMAND\notbool	331
COMMAND\notfontsetup	353
COMMAND\notfontsizeX	42, 353
COMMAND\notenumfont	353
COMMAND\notenumfontX	41, 353
COMMAND\notesXwidthliketwocolumns	354
COMMAND\noteswidthliketwocolumnsX	47, 354, 410, 412
COMMAND\num@lines	135, 155
COMMAND\numberlinefalse	19
COMMAND\numberlinetrue	19
COMMAND\numberonlyfirstinline	240, 354
COMMAND\numberonlyfirstintwolines	354
COMMAND\numberpstartfalse	18
COMMAND\numberpstarttrue	18, 37, 351, 406, 414
COMMAND\numberstanza	38
COMMAND\numberstanzafalse	50
COMMAND\numberstanzatrue	50
COMMAND\numlabfont	20, 47, 93
COMMAND\one@line	135
COMMAND\onehalfspacing	415
COMMAND\onlyXpstart	354
COMMAND\onlysideX	234
COMMAND\page@action	96, 106
COMMAND\page@start	96, 405
COMMAND\pagecontents	96
COMMAND\pagelinesep	59
COMMAND\pageparbreak	352
COMMAND\pageref	54, 259
COMMAND\par	26, 35, 138, 139, 211
COMMAND\par@line	135, 155
COMMAND\para@footgroup	167
COMMAND\para@footgroupX	209
COMMAND\para@footsetup	167, 403
COMMAND\para@footsetupX	207, 403, 410
COMMAND\para@vfootnoteX	207
COMMAND\parafootfmt	169, 170, 417
COMMAND\parafootfmtX	208
COMMAND\parafootftm	172
COMMAND\parafootftmX	210
COMMAND\parafootftmsep	353
COMMAND\parafootsep	354, 408, 413
COMMAND\parafootsepX	44, 98, 353, 354, 418
COMMAND\parafootstart	167
COMMAND\parafootstartX	207
COMMAND\paravfootnote	168, 171
COMMAND\parfillskip	170
COMMAND\parindent	415
COMMAND\parindentX	43, 356, 415, 417
COMMAND\parshape	67
COMMAND\parskip	139

COMMAND\pausenumbering	19, 85, 99, 101, 139, 408, 410, 417
COMMAND\penalty	170
COMMAND\pend	2, 6, 17–20, 22, 64, 65, 118, 120, 123, 128, 135–140, 151, 152, 350, 408, 409, 418, 419
COMMAND\preXnotes	412, 418
COMMAND\prenotesX	45, 215, 412
COMMAND\prepare@Xgroupbyline	190
COMMAND\prepare@Xprenotes	214
COMMAND\prev@nopb	342
COMMAND\prev@pb	342
COMMAND\prevlineX	98
COMMAND\prevpageX@num	98
COMMAND\print@Xfootnoterule	412
COMMAND\print@Xnotes	250, 251
COMMAND\print@Xnotes@forpages	411
COMMAND\print@eledsection	142
COMMAND\print@footnoteXrule	412
COMMAND\print@leftmargin@eledsection	331
COMMAND\print@lemma	181
COMMAND\print@line	140
COMMAND\print@notesX@forpages	411
COMMAND\print@rightmargin@eledsection	331
COMMAND\printendlines	222, 227, 265, 356, 403, 405
COMMAND\printlinefootnote	182, 183, 411
COMMAND\printlinefootnotearea	183, 184, 411
COMMAND\printlinefootnotenotenumbers	182
COMMAND\printlines	163, 180, 185, 186, 222, 265, 356, 403, 405, 411, 416
COMMAND\printnpnum	351, 356
COMMAND\printpstart	181
COMMAND\protect	121, 350
COMMAND\providecommand	192, 403
COMMAND\pstart	2, 6, 17–20, 22, 63–65, 106, 118, 123, 128, 135, 137–139, 142, 152, 336, 350, 405–407, 409, 410, 412–414, 417–419
COMMAND\pstartinfootnote	354
COMMAND\pstartinfootnoteeverytime	354
COMMAND\pstartnum	151
COMMAND\pstartref	52, 254, 260, 408
COMMAND\pstarts	406
COMMAND\raggedX	44
COMMAND\raggedleft	44
COMMAND\raggedright	43
COMMAND\raw@text	135
COMMAND\rbracket	40, 41
COMMAND\read@linelist	99–101
COMMAND\ref	54, 56, 59
COMMAND\refformated@	265
COMMAND\refformatedwithpage	265
COMMAND\relax	18, 106, 126, 145, 153, 297, 312, 350
COMMAND\renewcommand	68, 353, 356
COMMAND\reset@msd@options@iffullpage	306
COMMAND\resetprevline@	98

COMMAND\resetprevpage@	98
COMMAND\resumenumbering	19, 82, 85, 99, 101, 139, 405, 409, 410, 417
COMMAND\right	62
COMMAND\rightctab	324
COMMAND\rightlinenum	20, 93, 403, 405
COMMAND\rightltab	324
COMMAND\rightnoteupfalse	57
COMMAND\rightrtab	324
COMMAND\rightrightnote	273
COMMAND\rightrightskip	164, 167, 169, 170
COMMAND\rightstartnum	151
COMMAND\rigidbalance	173, 174, 176, 356, 416
COMMAND\rigidbalanceX	173, 356, 416
COMMAND\robustify	35
COMMAND\roman	317, 416
COMMAND\rtab	323–325, 328
COMMAND\rtabtext	325, 328
COMMAND\sameword	27–29, 113, 129–131, 133, 411, 413, 415, 418
COMMAND\sameword@inedtext	130
COMMAND\saweword	129
COMMAND\scriptsize	93
COMMAND\section	63, 405, 419
COMMAND\section@num	82
COMMAND\sectionmark	330
COMMAND\select@lemmafont	47, 48, 157
COMMAND\series	227
COMMAND\series@	227
COMMAND\seriesatbegin	33, 239, 412
COMMAND\seriesatend	33, 239, 412
COMMAND\set@Xtxtbeforenotes	156
COMMAND\set@line	126
COMMAND\set@line@action	97, 106
COMMAND\setSErefonlypageprefixmore	55, 265, 417
COMMAND\setSErefonlypageprefixsingle	55, 265, 417
COMMAND\setSErefprefixmore	55
COMMAND\setSErefprefixsingle	55
COMMAND\setapprefprefixmore	55, 353
COMMAND\setapprefprefixsingle	55, 353, 416
COMMAND\setcommand@series	241
COMMAND\sethangingsymbol	50, 292, 353, 415
COMMAND\sethangingsymbol	48
COMMAND\setistwofollowinglines	188
COMMAND\setl@dlprbox	274
COMMAND\setline	21, 22, 97, 102, 105, 118, 121, 137, 414
COMMAND\setlinenum	22, 102, 106, 118, 403
COMMAND\setmsdatalabel	32
COMMAND\setmsdataseries	32
COMMAND\setprintendlines	222, 223, 405
COMMAND\setprintlines	186, 187, 222, 405
COMMAND\setsidenotessep	57

COMMAND\setsidenotsep	353
COMMAND\setstanzaindent	294
COMMAND\setstanzaindents	49, 293, 350
COMMAND\setstanzapenalties	293
COMMAND\setstanzavalues	293
COMMAND\settoggle@series	240, 406, 410
COMMAND\showlemma	121, 403, 404
COMMAND\showwordrank	30, 130
COMMAND\sidenote@margin	404
COMMAND\sidenotemargin	56, 404, 409
COMMAND\sidenotesep	353
COMMAND\sidepstartnumtrue	18
COMMAND\skip	164
COMMAND\skipnumbering	22, 109, 118, 405, 412, 413
COMMAND\skipnumbering@reg	412
COMMAND\small	42
COMMAND\special	12
COMMAND\splitmaxdepth	160, 175
COMMAND\splitoff	173
COMMAND\splittopskip	159, 175, 176
COMMAND\stanza	21, 22, 50, 51, 297, 353, 355, 415
COMMAND\stanza@hang	296
COMMAND\stanza@line	296
COMMAND\stanzaindent	49, 294, 411
COMMAND\stanzaindent*	49
COMMAND\stanzaindentbase	293
COMMAND\stanzanumwrapper	51
COMMAND\startlock	21, 97, 118, 297
COMMAND\startstanzahook	353
COMMAND\startsub	21, 97, 117
COMMAND\stopmsd	301
COMMAND\stopmsdata	31, 299
COMMAND\strip@pt	167
COMMAND\strutbox	175
COMMAND\sub@action	97, 107
COMMAND\sub@lock	95
COMMAND\sub@off	104, 257
COMMAND\sub@on	104, 257
COMMAND\subline@num	94–96
COMMAND\sublinenum@rep	403
COMMAND\sublinenumberstyle	22, 93, 403
COMMAND\sublinenumincrement	19
COMMAND\sublinenumr@p	92, 403, 405
COMMAND\sublinenumrep	92, 93, 405
COMMAND\sublineref	52, 254, 260
COMMAND\subsectionmark	330
COMMAND\sw@inthisedtext	123
COMMAND\sw@list@inedtext	127, 133
COMMAND\symlinenum	354
COMMAND\symplinum	353

COMMAND\sza@penalty	296
COMMAND>tag	411
COMMAND\text	348
COMMAND\text<language>	42
COMMAND\textcolor	68
COMMAND\textheight	68
COMMAND\the	403
COMMAND\thefootnoteA	32
COMMAND\thefootnoteX	407
COMMAND\thelabidx	288
COMMAND\thepage	102
COMMAND\thepstart	18
COMMAND\thepstartL	406
COMMAND\thepstartR	406
COMMAND\thestanza	50
COMMAND\this@line@list@version	114
COMMAND\thisfootnote	200
COMMAND\threecol@begin@insert	176
COMMAND\threecolfootfmt	176, 417
COMMAND\threecolfootfmtX	205
COMMAND\threecolfootgroup	174
COMMAND\threecolfootgroupX	205
COMMAND\threecolfootsetup	174
COMMAND\threecolfootsetupX	204
COMMAND\threecolvfootnote	175
COMMAND\threecolvfootnote@inserted	175
COMMAND\threecolvfootnoteX	205
COMMAND\toendnotes	25, 220, 418
COMMAND\twocolfootfmt	417
COMMAND\twocolfootfmtX	203
COMMAND\twocolfootgroupX	203
COMMAND\twocolfootsetupX	202
COMMAND\twocolvfootnoteX	202
COMMAND\twolines	240, 354
COMMAND\twolines@A	240
COMMAND\twolines@B	240
COMMAND\twolines@C	240
COMMAND\twolinesbutnotmore	354
COMMAND\twolinesonlyinsamepage	354
COMMAND\txtbeforeXnotes	354
COMMAND\unhbox	168
COMMAND\unpenalty	169–171
COMMAND\unskip	170
COMMAND\unvxh	170, 355
COMMAND\unvxhX	355
COMMAND\upbracefill	323
COMMAND\usingcritext	349, 352
COMMAND\usingedtext	349, 352
COMMAND\vAfootnote	159
COMMAND\variant	27

COMMAND\ vbox	137, 138, 168, 173, 213
COMMAND\ vfootnote	159, 164, 168, 175
COMMAND\ vl@dbfnote	193, 404
COMMAND\ vl@disnote	273
COMMAND\ vl@dlsnote	273
COMMAND\ vl@dosnote	273
COMMAND\ vl@drsnote	273
COMMAND\ vnumfootnoteX	405
COMMAND\ vsize	46, 68
COMMAND\ vsplit	154
COMMAND\ wklam	323
COMMAND\ wklamec	323
COMMAND\ wapunktel	323
COMMAND\ wastricht	323
COMMAND\ widthX	46, 356, 417
COMMAND\ wrap@edcrossref	259, 410
COMMAND\ wrapcontentX	42, 418
COMMAND\ wrapped@bodyfootmarkX	211
COMMAND\ wrapped@footfootmarkX	210
COMMAND\ x...	53
COMMAND\ xdef	87, 297
COMMAND\ xflagref	53, 260, 356, 416
COMMAND\ xleft@appenditem	87, 120
COMMAND\ xlineref	53, 356, 416
COMMAND\ xpageref	53
COMMAND\ xpstartref	53, 408
COMMAND\ xr	56
COMMAND\ xright@appenditem	87
COMMAND\ xsublineref	53
COMMAND\ xxref	53, 262, 268, 408, 411, 418
COMMAND\ zz@@@	403
ENVIRONMENTastanza	416
ENVIRONMENTedarrayc	328
ENVIRONMENTedarrayl	328
ENVIRONMENTedarrayr	328
ENVIRONMENTedtabularc	328
ENVIRONMENTedtabularl	328
ENVIRONMENTedtabularr	328
ENVIRONMENTledgroup	73, 280, 356, 416
ENVIRONMENTledgroupsize	280
PACKAGE(r)(e)ledmac	33
PACKAGEEledmac	11, 70, 96, 285, 351, 352, 411, 412
PACKAGEEledpar	412
PACKAGEEtoolbox	73
PACKAGEParallel	358
PACKAGEReledmac	355, 356
PACKAGEamsgen	306
PACKAGEamsmath	306, 307
PACKAGEbabel	42, 69, 317, 416
PACKAGEbiblatex	67

PACKAGEbidi	42, 43, 74, 415
PACKAGEccaption	81
PACKAGEcolor	68
PACKAGEedmac	1, 6, 10, 12, 13, 70, 185, 191, 255, 293, 348, 358, 403
PACKAGEedstanza	1, 13, 291
PACKAGEeledmac	1, 10, 13–16, 59, 191, 281, 285, 310, 332, 345, 349, 351–353, 407, 409, 411
PACKAGEeledpar	81, 160, 330, 358, 405, 409–411
PACKAGEetex	415
PACKAGEtoolbox	86, 129, 227, 240, 248, 273, 331, 342
PACKAGEfancyhdr	252, 419
PACKAGEfloatrow	66, 356
PACKAGEfootmisc	32, 69, 74, 191, 192, 358, 419
PACKAGEgeometry	14
PACKAGEglossaries	60, 290, 416
PACKAGEhandout	410
PACKAGEhyperlink	235
PACKAGEhyperref	53, 123, 210, 211, 255, 287, 335, 344, 408–410, 417
PACKAGEifluatex	73
PACKAGEifxetex	73
PACKAGEimakeidx	58, 67, 74, 281, 285, 352, 407–409, 411
PACKAGEindextols	289
PACKAGEindextool	352
PACKAGEindextools	58, 67, 74, 82, 281, 285, 289, 352, 411, 416, 419
PACKAGEinputenc	131
PACKAGEledarab	69
PACKAGEledmac	1, 10, 13, 14, 69, 86, 285, 348, 349, 352, 355
PACKAGEledpar	69
PACKAGEMemoir	73, 285, 352, 358, 410
PACKAGEMorewrites	67
PACKAGEMusixtex	409
PACKAGEperpage	416
PACKAGEpolyglossia	40, 69, 124, 158, 180, 416
PACKAGERagged2e	44, 73
PACKAGEreledmac	1, 2, 10–12, 14–16, 18, 19, 22–24, 26–28, 30, 32, 33, 36, 37, 39, 42, 43, 45–48, 50, 51, 53, 54, 56, 58–60, 65–71, 88, 90, 96, 97, 100, 101, 104, 113, 114, 121, 126, 129, 152, 161, 164, 169, 180, 191, 192, 217, 228, 232, 233, 240, 248, 259, 262, 268, 285, 303, 310, 330, 331, 343, 344, 352, 353, 355, 356, 414, 417, 419
PACKAGEreledpar	1, 4, 6, 8, 15, 18, 47, 53, 55, 65–67, 69, 71, 81, 88, 99, 104, 124, 126, 161, 165, 212, 213, 228, 234, 248, 250, 251, 281, 292, 415, 416, 418, 419
PACKAGESuffix	73
PACKAGETabmac	1, 13, 358
PACKAGEuninormalize	28
PACKAGExargs	27, 73
PACKAGExkeyval	71, 248
PACKAGEXr	5, 56, 268, 417
PACKAGEXref	268
PACKAGEXstring	73, 287

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
\add@Xgroupbyline	1
\addtol@denbody	1
Adelard II	13
\advancelabel@refs	1
\advanceline	1, 21
\Aendnote	24
\affixline@num	1
\affixpstart@num	1
\affixside@note	1
\Afootnote	24
\afternoteX	44
\afterruleX	46
\ampersand	1, 51
\append@notesX	1
\append@Xnotes	1
\applabel	1, 54
\appref	1, 54
\apprefwithpage	1, 54
\arrangementX	1, 34
\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	67
\ballast@count	1
Beeton, Barbara Ann Neuhaus Friend	18
\beforeeledchapter	1
\beforeinsertingX	43
\beforenotesX	45
\beginnumbering	1, 16
\Bendnote	24
\Bfootnote	24
\bhookgroupX	45
\bhooknoteX	43
\bodyfootmarkA	32

<code>\boxfootnotenumbers</code>	1
Bredon, Simon	13
Breger, Herbert	13, 310
Brey, Gerhard	13
Busard, Hubert L. L.	13
<code>\bypage@false</code>	1
<code>\bypage@true</code>	1
<code>\bypstart@false</code>	1
<code>\bypstart@true</code>	1

C

<code>\c@addcolcount</code>	1
<code>\c@ballast</code>	1
<code>\c@firstlinenum</code>	1
<code>\c@firstsublinenum</code>	1
<code>\c@labidx</code>	1
<code>\c@linenumincrement</code>	1
<code>\c@sublinenumincrement</code>	1
<code>\Cendnote</code>	24
<code>\Cfootnote</code>	24
<code>\ch@ck@l@ck</code>	1
<code>\ch@cksub@l@ck</code>	1
<code>\chapter</code>	1
<code>\check@pb@in@verse</code>	1
Chester, Robert of	13
Claassens, Geert H. M.	13
<code>\colalignX</code>	43
Copernicus, Nicolaus	13
<code>\critext</code>	348
<code>\ctab</code>	1
<code>\ctabtext</code>	1

D

Dekker, Dirk-Jan	68
<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@feet@custom@order</code>	1
<code>\do@feetX</code>	1
<code>\do@insidelinehook</code>	1
<code>\do@line</code>	1
<code>\do@linehook</code>	1
<code>\do@lockoff</code>	1

<code>\do@lockoffL</code>	<u>1</u>
<code>\do@lockon</code>	<u>1</u>
<code>\do@lockonL</code>	<u>1</u>
<code>\do@Xfeet</code>	<u>1</u>
<code>\doedindexlabel</code>	<u>1</u>
<code>\doendnotes</code>	<u>1</u> , 25
<code>\doendnotesbysection</code>	<u>1</u> , 25
<code>\doinsidelinehook</code>	<u>1</u> , 22
<code>\dolinehook</code>	<u>1</u> , 22
<code>\dosplits</code>	<u>1</u>
Downes, Michael	67, 168, 170
<code>\doxtrafeet</code>	<u>1</u>
<code>\dummy@edtext</code>	<u>1</u>
<code>\dummy@edtext@showlemma</code>	<u>1</u>
<code>\dummy@ref</code>	<u>1</u>

E

<code>\edaftertab</code>	<u>1</u> , 63, 323
<code>edarrayc</code> (environment)	60
<code>edarrayl</code> (environment)	60
<code>edarrayr</code> (environment)	60
<code>\edatleft</code>	<u>1</u> , 62
<code>\edatright</code>	<u>1</u> , 62
<code>\edbeforetab</code>	<u>1</u> , 63, 323
<code>\edfilldimen</code>	<u>1</u>
<code>\edfont@info</code>	<u>1</u>
<code>\edindex</code>	<u>1</u> , 57
<code>\edindexlab</code>	<u>1</u> , 59
<code>\EDLABEL</code>	<u>1</u>
<code>\edlabel</code>	<u>1</u> , 52
<code>\edlabelE</code>	<u>1</u> , 54
<code>\edlabels</code>	<u>1</u> , 54
<code>\edlabelSE</code>	<u>1</u> , 54
<code>\edlineref</code>	<u>1</u> , 52
<code>\edmakelabel</code>	<u>1</u> , 53
<code>\edpageref</code>	<u>1</u> , 52
<code>\edrowfill</code>	<u>1</u> , 61
<code>\EDTAB</code>	<u>1</u>
<code>\edtabcolsep</code>	<u>1</u> , 61
<code>\EDTABINDENT</code>	<u>1</u>
<code>\edtabindent</code>	<u>1</u>
<code>\EDTABtext</code>	<u>1</u>
<code>edtabularc</code> (environment)	60
<code>edtabularl</code> (environment)	60
<code>edtabularr</code> (environment)	60
<code>\EDTEXT</code>	<u>1</u>
<code>\edtext</code>	<u>1</u> , 23
<code>\edvertdots</code>	<u>1</u> , 63
<code>\edvertline</code>	<u>1</u> , 63
<code>\Endnote</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>	60
<code>edarrayl</code>	60
<code>edarrayr</code>	60
<code>edtabularc</code>	60
<code>edtabularl</code>	60
<code>edtabularr</code>	60
<code>ledgroup</code>	51
<code>ledgroupsize</code>	51
<code>minipage</code>	51
<code>Euclid</code>	13
<code>\extensionchars</code>	1, 66

F

<code>\f@x@l@cks</code>	1
Fairbairns, Robin	32
<code>\first@linenum@out@false</code>	1
<code>\first@linenum@out@true</code>	1
<code>\firstlinenum</code>	1, 19
<code>\firstseriesX@</code>	1
<code>\firstsublinenum</code>	1, 19
<code>\firstXseries@</code>	1
<code>\fix@page</code>	1

<code>\flag@end</code>	1
<code>\flag@end@later</code>	1
<code>\flag@start</code>	1
<code>\flag@start@later</code>	1
<code>\flagstanza</code>	1, 51
<code>\flush@notes</code>	1
<code>\fnpos</code>	1, 33
Folkerts, Menso	13
<code>\footfootmarkA</code>	32
<code>\footfudgefiddle</code>	1, 68
<code>\footnote</code>	1
<code>\footnoteA</code>	32
<code>\footnoteB</code>	32
<code>\footnoteC</code>	32
<code>\footnoteD</code>	32
<code>\footnoteE</code>	32
<code>\footnotelang@lua</code>	1
<code>\footnotelang@poly</code>	1
<code>\footnoteoptions@</code>	1
<code>\footsplitskips</code>	1
<code>\fullstop</code>	1

G

Gädeke, Nora	13
<code>\get@edindex@hyperref</code>	1
<code>\get@edindex@ledinnote@command</code>	1
<code>\get@fnmark</code>	1
<code>\get@fnmarkX</code>	1
<code>\get@index@command</code>	1
<code>\get@linelistfile</code>	1
<code>\get@sw@txt</code>	1
<code>\get@thisfootnote</code>	1
<code>\get@thisfootnoteX</code>	1
<code>\getline@num</code>	1
<code>\gl@p</code>	1

H

<code>\h@num</code>	1
<code>\hangindentX</code>	43
<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>	44
<code>\hsizetwocolX</code>	44

\Hy@raisedlink@left	1
\hyperlinkformat	1
\hyperlinkformatR	1
\hyperlinkR	1

I

\if@addsw	1
\if@edtext@secondarg@	1
\if@eled@sectioning	1
\if@firstlineofpage	1
\if@led@nofoot	1
\if@ledgroup	1
\if@lemmacommand@	1
\if@msdata@insertedfrompreviouspage	1
\if@noeled@sec	1
\if@noneed@Footnote	1
\if@RTL	1
\ifautopar@pause	1
\ifbypage@	1
\ifbypage@R	1
\ifbypstart@	1
\ifbypstart@R	1
\ifeledmacompat@	1
\iffirst@linenum@out@	1
\ifindtl@innote	1
\ifindtl@notenumber	1
\ifinserthangingsymbol	1
\ifinstanza	1
\ifl@d@dash	1
\ifl@d@elin	1
\ifl@d@esl	1
\ifl@d@pnum	1
\ifl@d@ssub	1
\ifl@d@Xmorethantwolines	1
\ifl@d@Xtwolines	1
\ifl@dend@X	1
\ifl@dhidenummer	1
\ifl@dmemoir	1
\ifl@dpaging	1
\ifl@dpairing	1
\ifl@dprintingcolumns	1
\ifl@dprintingpages	1
\ifl@dskipnumber	1
\ifl@dskipversenumber	1
\ifl@dstartendok	1
\ifl@footmisc	1
\ifl@imakeidx	1
\ifl@indextools	1
\ifledfinal	1, 66
\ifledgroupnotesL@	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	156
---------------------------	-----

L

<code>\l@advance@parledgroup@beforenormalnotes</code>	1
<code>\l@d@add</code>	1
<code>\l@d@nums</code>	1
<code>\l@d@section</code>	1
<code>\l@d@set</code>	1
<code>\l@d@Xend</code>	1
<code>\l@dampcount</code>	1

\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@ddodoreintrafeet	1
\l@dedbeginmini	1
\l@dedendmini	1
\l@emptyd@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
\l@dlsn@te	1
\l@dlsnote	1
\l@dmake@labels	1
\l@dmodforedtext	1
\l@dnullfills	1
\l@dnumpstartsL	1
\l@doldold@footnotetext	1
\l@dp@rsefootspec	1
\l@dparsedendline	1
\l@dparsedendpage	1
\l@dparsedendsub	1
\l@dparsedstartline	1
\l@dparsedstartpage	1
\l@dparsedstartsub	1

<code>\ldparsefootspec</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>	54
<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@BadAction</code>	1
<code>\led@err@edtextoutsidepstart</code>	1
<code>\led@err@EdtextWithoutFootnote</code>	1
<code>\led@err@FootnoteNotInSecondArgEdtext</code>	1
<code>\led@err@HighEndColumn</code>	1
<code>\led@err@LineationInNumbered</code>	1
<code>\led@err@LowStartColumn</code>	1
<code>\led@err@ManyLeftnotes</code>	1
<code>\led@err@ManyRightnotes</code>	1
<code>\led@err@ManySidenotes</code>	1
<code>\led@err@NumberingNotStarted</code>	1
<code>\led@err@NumberingShouldHaveStarted</code>	1
<code>\led@err@NumberingStarted</code>	1
<code>\led@err@NumberingWithoutPstart</code>	1
<code>\led@err@PendNoPstart</code>	1
<code>\led@err@PendNotNumbered</code>	1
<code>\led@err@PstartInPstart</code>	1
<code>\led@err@PstartNotNumbered</code>	1
<code>\led@err@ReverseColumns</code>	1
<code>\led@err@toendnotes@outsidenumbering</code>	1
<code>\led@err@TooManyColumns</code>	1
<code>\led@err@UnequalColumns</code>	1

<code>\led@error@fail@patch@@doclearpage</code>	1
<code>\led@error@fail@patch@@iiiminipage</code>	1
<code>\led@error@fail@patch@@makecol</code>	1
<code>\led@error@fail@patch@@reinserts</code>	1
<code>\led@error@fail@patch@endminipage</code>	1
<code>\led@error@PackageAfterEledmac</code>	1
<code>\led@mess@NotesChanged</code>	1
<code>\led@mess@SectionContinued</code>	1
<code>\led@nopb</code>	1
<code>\led@nopbnum</code>	1
<code>\led@pb</code>	1
<code>\led@pb@setting</code>	1
<code>\led@pbnum</code>	1
<code>\led@reinit@index@fornote</code>	1
<code>\led@set@index@fornote</code>	1
<code>\led@toksa</code>	1
<code>\led@toksb</code>	1
<code>\led@warn@AppLabelOutSecondArgEdtext</code>	1
<code>\led@warn@BadAction</code>	1
<code>\led@warn@BadAdvancelineLine</code>	1
<code>\led@warn@BadAdvancelineSubline</code>	1
<code>\led@warn@BadLineation</code>	1
<code>\led@warn@BadLinenummargin</code>	1
<code>\led@warn@BadLockdisp</code>	1
<code>\led@warn@BadSetline</code>	1
<code>\led@warn@BadSetlinenum</code>	1
<code>\led@warn@BadSidenotemargin</code>	1
<code>\led@warn@BadSubblockdisp</code>	1
<code>\led@warn@DuplicateLabel</code>	1
<code>\led@warn@edinde@outsidenumbering</code>	1
<code>\led@warn@LineFileObsolete</code>	1
<code>\led@warn@NoIndexFile</code>	1
<code>\led@warn@NoLineFile</code>	1
<code>\led@warn@NoMarginpars</code>	1
<code>\led@warn@RefUndefined</code>	1
<code>\led@warn@SeriesStillExist</code>	1
<code>\led@warning@hsizeX@deprecated</code>	1
<code>\led@warning@msdatawithoutstop</code>	1
<code>\led@warning@preXnotes@deprecated</code>	1
<code>\led@warning@Xhsize@deprecated</code>	1
<code>ledgroup (environment)</code>	51
<code>ledgroupsize (environment)</code>	51
<code>\ledinnernote</code>	1, 56
<code>\ledinnote</code>	1
<code>\ledinnotehyperpage</code>	1
<code>\ledinnotemark</code>	1
<code>\ledleftnote</code>	1, 56
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 57

<code>\ledlsnotesep</code>	1, 57
<code>\ledlsnotewidth</code>	1, 57
<code>\lednopb</code>	1, 65
<code>\lednopbinversetrue</code>	65
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	56
<code>\ledouterote</code>	1
<code>\ledpb</code>	1, 65
<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 65
<code>\ledrightnote</code>	1, 56
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 57
<code>\ledrsnotesep</code>	1, 57
<code>\ledrsnotewidth</code>	1, 57
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 56
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 20
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	57
<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>\linerrangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1

\lock@off	1
\lock@on	1
\lockdisp	1, 21
Lorch, Richard	13
\ltab	1
\ltabtext	1
Luecking, Dan	72

M

\m@mmf@check	1
\m@mmf@prepare	1
\M@sect	1
\makehboxofhboxes	1
\managestanza@modulo	1
\maxhnotesX	46
Mayer, Gyula	13
\measurebody	1
\measurecell	1
\measuremrow	1
\measuretbody	1
\measuretcell	1
\measuretrow	1
Middleton, Thomas	13, 95
minipage (environment)	51
Mittelbach, Frank	12
\morenoexpands	1, 68
\mp@append@notesX	1
\mp@append@Xnotes	1
\mpfnpos	1, 33
\mpnormalfootgroup	1
\mpnormalfootgroupX	1
\mpnormalvfootnote	1
\mpnormalvfootnote@inserted	1
\mpnormalvfootnoteX	1
\mppara@footgroupX	1
\mppara@vfootnoteX	1
\mpparafootgroup	1
\mpparavfootnote	1
\mpthreecolfootgroup	1
\mpthreecolfootgroupX	1
\mpthreecolfootsetup	1
\mpthreecolfootsetupX	1
\mptwocolfootgroup	1
\mptwocolfootgroupX	1
\mptwocolfootsetup	1
\mptwocolfootsetupX	1
\msdata	1, 31
\multfootsep	1, 32
\multiplefootnotemarker	1

N

<code>\n@num</code>	1
<code>\n@num@stanza</code>	1
<code>\new@line</code>	1
<code>\newhookarg@specific</code>	1
<code>\newhookcommand@series</code>	1
<code>\newhookcommand@series@reload</code>	1
<code>\newhooktoggle@series</code>	1
<code>\newhooktoggle@series@reload</code>	1
<code>\newhooktoggle@specific</code>	1
<code>\newseries@</code>	1
<code>\newverse</code>	1
<code>\NEXT</code>	1
<code>\no@expands</code>	1
<code>\noeledsec</code>	65
<code>\nomk@</code>	1
<code>\normal@footnotemarkX</code>	1
<code>\normal@page@break</code>	1
<code>\normal@pars</code>	1
<code>\normalbfnoteX</code>	1
<code>\normalbodyfootmarkX</code>	1
<code>\normalfootfmt</code>	1
<code>\normalfootfmtX</code>	1
<code>\normalfootfootmarkX</code>	1
<code>\normalfootgroup</code>	1
<code>\normalfootgroupX</code>	1
<code>\normalfootnoterule</code>	1
<code>\normalfootnoteruleX</code>	1
<code>\normalfootstart</code>	1
<code>\normalfootstartX</code>	1
<code>\normalvfootnote</code>	1
<code>\normalvfootnote@inserted</code>	1
<code>\normalvfootnoteX</code>	1
<code>\notefontsizeX</code>	42
<code>\notenumfontX</code>	41
<code>\noteschanged@false</code>	1
<code>\noteschanged@true</code>	1
<code>\noteswidthliketwocolumnsX</code>	47
<code>\nulledindex</code>	1
<code>\nullsetzen</code>	1
<code>\num@lines</code>	1
<code>\numberedpar@false</code>	1
<code>\numberedpar@true</code>	1
<code>\numberingfalse</code>	1
<code>\numberingtrue</code>	1
<code>\numberlinefalse</code>	19
<code>\numberlinetrue</code>	19
<code>\numberpstartfalse</code>	1, 18
<code>\numberpstarttrue</code>	1, 18
<code>\numberstanzafalse</code>	50

\numberstanzatrue	50
\numlabfont	1, 47

O

\old@hsize	1
\one@line	1
optionauxdir	15, 418
optioncontinuousnumberingwithcolumns	417
optioninnnote	416
optioninnote	416
optionlinrangesep	248
optionnocritical	416
optionnoeledsec	336, 418
optionnoend	416
optionnopenalties	67
optionnotenumber	416

P

\page@action	1
\page@num	1
\pagelinesep	1, 58
\pageref	54
\par@line	1
\para@footgroupX	1
\para@footsetup	1
\para@footsetupX	1
\para@vfootnoteX	1
\parafootfmt	1
\parafootfmtX	1
\parafootgroup	1
\parafootsepX	44
\parafootstart	1
\parafootstartX	1
\paravfootnote	1
\parindentX	42
\pausenumbering	1, 18
\pend	1, 16
Plato of Tivoli	13
\postbodyfootmark	1
\prebodyfootmark	1
\prenotesX	45
\prepare@edindex@fornote	1
\prepare@prenotesX	1
\prepare@Xgroupbyline	1
\prepare@Xprenotes	1
\prev@nopb	1
\prev@pb	1
\prevpage@num	1
\print@eledsection	1
\print@footnoteXrule	1

<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>	52

Q

<code>\quotation</code>	1
<code>\quote</code>	1

R

<code>\raggedX</code>	44
<code>\raw@text</code>	1
<code>\rbracket</code>	1
<code>\read@linelist</code>	1
<code>\ref</code>	54
<code>\ref@reg@later</code>	1
<code>\Relax</code>	1
<code>\reledmac@error</code>	1
<code>\reledmac@warning</code>	1
<code>\removehboxes</code>	1
<code>\reset@msd@options@iffullpage</code>	1
<code>\resetprevline@</code>	1, 98
<code>\resetprevpage@</code>	1
<code>\resetprevpage@num</code>	98
<code>\restore@familiarnotes</code>	1
<code>\restore@notes</code>	1
<code>\restore@sidenotes</code>	1
<code>\resumenumbering</code>	1, 18
<code>\rightctab</code>	1
<code>\rightlinenum</code>	1, 20

\rightltab	1
\rightnoteupfalse	57
\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@lemmafnt	1
\select@lemmafnt	1, 47
\Seref	1, 54
\Serefonlypage	54
\Serefwithpage	1, 54
\series	1
\seriesatbegin	1, 33
\seriesatend	1, 33
\set@line	1
\set@line@action	1
\set@Xtxtbeforenotes	1
\setapprefprefixmore	54
\setapprefprefixsingle	54
\setcommand@series	1
\sethangingsymbol	1, 50
\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, 32
\setmsdataseries	1, 32
\setnoteswidthliketwocolumnsX@	1
\setnotesXpositionliketwocolumns@	1
\setprintendlines	1
\setprintlines	1
\setSerefonlypageprefixmore	55
\setSerefonlypageprefixsingle	55
\setSerefprefixmore	55

<code>\setSErefprefixsingle</code>	55
<code>\setsidenotesep</code>	57
<code>\setstanzaindents</code>	1, 48
<code>\setstanzapenalties</code>	1, 49
<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, 66
<code>\showwordrank</code>	1, 30
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 56
<code>\sidepstartnumtrue</code>	18
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 22
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 61
<code>\spreadtext</code>	1, 61
<code>\stanza</code>	1, 48
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 49
<code>\stanzaindent*</code>	1, 49
<code>\stanzaindentbase</code>	1, 48
<code>\stanzanumwrapper</code>	1, 51
<code>\startlock</code>	1, 21
<code>\startsub</code>	1, 21
<code>\stepl@dcolcount</code>	1
<code>\stopmsdata</code>	1, 31
<code>\strip@szacnt</code>	1
<code>\sub@action</code>	1
<code>\sub@lock</code>	1
<code>\sub@off</code>	1
<code>\sub@on</code>	1
<code>\subline@num</code>	1
<code>\sublinenumberstyle</code>	1, 22
<code>\sublinenumincrement</code>	1, 19
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 52
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\subblock@disp</code>	1

\subblockdisp	1
Sullivan, Wayne	12, 13, 67, 81, 85, 168, 169, 254, 291
\sza@penalty	1

T

\tabHilfbox	1
\tabhilfbox	1
\theadcolcount	1
\theadtext	1
\theendpageline	1
\thefootnoteA	32
Theodosius	13
\thepageline	1
\thepstart	1, 18
\thestanza	1, 50
\thestartpageline	1
\this@line@list@version	1
\threecol@begin@insert	1
\threecolfootfmt	1
\threecolfootfmtX	1
\threecolfootgroup	1
\threecolfootgroupX	1
\threecolfootsetup	1
\threecolfootsetupX	1
\threecolvfootnote	1
\threecolvfootnote@inserted	1
\threecolvfootnoteX	1
\toendnotes	25
\toendnotes*	1
\twocolfootfmt	1
\twocolfootfmtX	1
\twocolfootgroup	1
\twocolfootgroupX	1
\twocolfootsetup	1
\twocolfootsetupX	1
\twocolvfootnote	1
\twocolvfootnote@inserted	1
\twocolvfootnoteX	1

U

\unvxhX	1
---------	---

V

Vamana	13
\variab	1
\vbfnoteX	1
\vl@dbfnote	1
\vl@dcfnote	1
\vl@disnote	1
\vl@dlsnote	1

\vl@dosnote	1
\vl@drsnote	1
\vnumfootnoteX	1

W

Whitney, Ron	12
\widthX	46
\wrap@edcrossref	1
\wrapcontentX	42
\wrapped@bodyfootmarkX	1
\wrapped@footfootmarkX	1
Wujastyk, Dominik	12

X

\X@atbegininsertion	1
\X@beforeinsertion	1
\X@doreinfeet	1
\Xafterlemmaseparator	41
\Xafternote	44
\Xafternumber	38
\Xafterrule	46
\Xaftersymlinenum	39
\Xarrangement	1, 34
\Xarrangement@normal	1
\Xarrangement@paragraph	1
\Xarrangement@threecol	1
\Xarrangement@twocol	1
\Xbeforeinserting	43
\Xbeforelemmaseparator	40
\Xbeforenotes	45
\Xbeforenumber	36, 38
\Xbeforesymlinenum	39
\Xbhookgroup	45
\Xbhooknote	43
\Xboxlinenum	39
\Xboxlinenumalign	39
\Xboxsymlinenum	39
\Xcolalign	43
\xedindex	1
\xedlabel	1
\xedtext	1
\Xendafterenumber	38
\Xendafterlemmaseparator	41
\Xendafternote	47
\Xendafterpagenumber	40
\Xendaftersymlinenum	39
\Xendahookinplaceofnumber	40
\Xendahooklinenum	40
\Xendbeforelemmaseparator	41
\Xendbeforenumber	38

\Xendbeforepagenumber	40
\Xendbeforesymlinenum	39
\Xendbhookinplaceofnumber	40
\Xendbhooklinenum	40
\Xendbhooknote	43
\Xendboxendlinenumalign	40
\Xendboxlinenum	40
\Xendboxlinenumalign	40
\Xendboxstartlinenumalign	40
\Xendboxsymlinenum	39
\Xendhangindent	43
\Xendinplaceoflemmaseparator	41
\Xendinplaceofnumber	39
\Xendinplaceofpagenumber	36
\Xendlemmadisablefontselection	42
\Xendlemmafont	42
\Xendlemmaseparator	41
\Xendlineprefixmore	40
\Xendlineprefixsingle	40
\Xendlinerrangeseparator	36
\Xendmorethantwolines	37
\Xendnonumber	37
\Xendnotefontsize	42
\Xendnotenumfont	41
\Xendnumberonlyfirstinline	35
\Xendnumberonlyfirstintwolines	35
\Xendpagenumberonlyfirst	36
\Xendparagraph	47
\Xendsep	47
\Xendsublinesep	38
\Xendsymlinenum	35
\Xendsympagenum	36
\Xendtwolines	37
\Xendtwolinesbutnotmore	37
\Xendwrapcontent	42
\xflagref	1
\Xgroupbyline	45
\Xgroupbylineseparetwolines	45
\Xhangindent	43
\Xhsizethreecol	44
\Xhsizetwocol	44
\Xinplaceoflemmaseparator	41
\Xinplaceofnumber	39
\Xinsertparafootsep	1
\Xledsetnormalparstuff	1
\xleft@appenditem	1
\Xlemmadisablefontselection	41
\Xlemmafont	42
\Xlemmaseparator	40
\Xlinerrangeseparator	36

\xlineref	<u>1</u> , 53
\Xmaxhnotes	46
\Xmorethantwolines	37
\Xnolemmaseparator	<u>1</u> , 41
\Xnonbreakableafternumber	38
\Xnonumber	37
\Xnotefontsize	42
\Xnotenumfont	41
\Xnoteswidthliketwocolumns	47
\Xnumberonlyfirstinline	35
\Xnumberonlyfirstintwolines	35
\Xonlypstart	38
\Xpagelinesep	38
\xpageref	<u>1</u> , 53
\Xparafootsep	44
\Xparindent	42
\Xprenotes	<u>1</u> , 45
\Xprenotes@	<u>1</u>
\Xpstart	37
\Xpstarteverytime	37
\xpstartref	<u>1</u> , 53
\XR@test	<u>1</u>
\XR@test@mac	<u>1</u>
\XR@test@mac@test	<u>1</u>
\Xragged	44
\xright@appenditem	<u>1</u>
\Xrigidbalance	<u>1</u>
\Xstanza	38
\Xstanzaseparator	38
\xsublineref	<u>1</u> , 53
\Xsublinesep	38
\Xsymlinenum	35
\Xtoendnotes	25
\Xtwolines	37
\Xtwolinesonlyinsamepage	37
\Xtxtbeforenotes	45
\Xunvxh	<u>1</u>
\Xwidth	46
\Xwrapcontent	42
\Xwrapendlemma	42
\Xwraplemma	42
\xxref	<u>1</u> , 53
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	73
\morenoexpands: Added \l@dtabnoexpands to \no@expands	122
\reledmac@error: Added \eledmac@error and replaced error messages	75
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	257
\doxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	248
\edlabel: Tweaked \edlabel to get correct page numbers	255
\l@ddodoreinxtrafeet: Renamed \dodoreinxtrafeet to \l@ddodoreinxtrafeet	250
\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	122
\zz@@@: Minor change to \zz@@@	254
General: Added text about normal labeling	54
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	252
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	167
\line@list@stuff: Added initial write of page number in \line@list@stuff	114
\para@footsetup: Added \footfudgefiddle to \para@footsetup	167
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	207
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	81
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	257
\@nloreg: Added a bunch of code to \@nl for handling \setlinenum	102
\ledlinenum: Added \linenumr@p and \sublinenum@rep to \leftlinenum and \rightlinenum	93
\linenumberlist: Added \linenumberlist mechanism	81
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	223
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	189
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	93
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	278
\Xarrangement@normal: Added minpage footnote setup to \footnormal	161
\edtext: Added \showlemma to \edtext	123
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	277

\mpnormalfootgroup: Added \mpnormalfootgroup	165
\mpnormalvfootnote: Added \mpnormalvfootnote	163
\showlemma: Added \showlemma	81
General: Added final/draft options	71
Added ledgroup environment	280
Added ledgroupsize environment	280
Added minipage, etc., support	1
v0.4.1.	
\do@Xfeet: Changed \do@Xfeet code for easier extensions	249
\edindex: Leteledmac take advantage of memoir's indexing	285
\print@Xnotes: Added \opXfeet	249
General: Added code for changing \doclearpage	253
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	192
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	193
\@xympar: Eliminated \marginpar disturbance	269
General: Added left and right side notes	269
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	147
\l@getsidenote@margin: Added \sidenotemargin and \sidenote@margin	269
General: Added moveable side note	269
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	280
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	104
\@nl@reg: Added \fix@page to \@nl	102
Extended \@nl to include the page number	102
\fix@page: Added \last@page@num and \fix@page	103
\get@thisfootnote: Changed \l@dbfnote and \v@dbfnote as originals could give incorrect markers in the footnotes	193
\new@line: Extended \new@line to output page numbers	115
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared foreledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	102
\@ref@reg: Added \@ref@reg	111
\affixline@num: Added skipnumering to \affixline@num	147
\do@actions@fixedcode: Added \do@actions@fixedcode	146
\do@actions@next: Added number skipping to \do@actions	145
\do@insidelinehook: Added \do@linehook for use in \do@line	143
\endnumbering: Changed \endnumbering foreledpar	84
\fix@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fix@l@cks	150
\footplitskips: Added \footplitskips for use in many footnote styles	160
\get@linelistfile: Added \get@linelistfile	100
\initnumbering@reg: Added \initnumbering@reg	83

\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	279
\l@dcsnotetext@r: Added \l@emptyd@ta	143
\l@dgetline@margin: Added \l@dgetline@margin	89
\l@dgetlock@disp: Added \l@dgetlock@disp	91
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	269
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	81
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	143
\l@dzeropenalties: Added \l@dzeropenalties	138
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum ..	93
\line@list@stuff: Deleted \page@start from \line@list@stuff	114
\list@clearing@reg: Added \list@clearing@reg	100
\n@num: Added \n@num	109
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	199
\resumenumbering: Changed \resumenumbering for eledpar	85
\setprintendlines: Added \setprintendlines for use by \printendlines ...	222
\setprintlines: Added \setprintlines for use by \printlines	186
\skipnumbering: Added \skipnumbering and supports	118
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	91
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	93
Using \sublinenumrep instead of \sublinenumr@p	93
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	201
General: eledmac having been available for 2 years, deleted the commented out original edmac texts	1
Maïeul Rouquette new maintainer	1
Made macros of all messages	74
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	81
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.	294
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	48
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.	255
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.	147
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	292
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code> .	188
General: Line numbering can be reset at each <code>pstart</code> .	88
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.	292
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	122
General: <code>\lemma</code> can contain commands.	26
Debug in lineation command	20
New generic commands to customize footnote display.	34, 240
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code> .	24
Options of <code>\Xfootnotes</code> .	157
Possibility to have commands in sidenotes.	56
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.	35
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code>	214
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	240
General: Add <code>\labelpstarttrue</code> .	18
Add <code>\Xnumberonlyfirstintwolines</code>	35
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	37
New hook to add arbitrary code at the beginning of the notes	43
New options for block of notes.	45
New package option: <code>parapparatus</code> .	1
New tools to change order of series	239
Sectioning commands.	63

v1.2.0.	
<code>\Xprenotes</code> :	Debug in familiar footnotes (bug introduced by v1.1). 214
<code>\endquote</code> :	Compatibility of <code>\ledchapter</code> with the <i>memoir</i> class. 329
v1.3.0.	
<code>\endquote</code> :	<i>Quotation</i> and quote environment inside numbered sections. 329
v1.4.0.	
<code>\edtext</code> :	Compatibility of <code>\edtext</code> with the right-to-left direction (with Polyglossia). 123
<code>\ledsetnormalparstuffX</code> :	Direction of footnotes with polyglossia. 211
<code>\newseries@</code> :	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. 230
<code>\rbracket</code> :	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). 180
General:	Compatibility with LuaTeX of RTL notes. 1
v1.4.1.	
<code>\affixside@note</code> :	Remove spurious spaces. 275
<code>\endquote</code> :	New option <i>noquotation</i> 329
<code>\get@thisfootnote</code> :	Compatibility of standard footnotes with <code>eledmac</code> when these footnotes contain any commands. 193
<code>\labelrefsparsesubline</code> :	Fix bug with <code>\edlabel</code> 256
v1.4.2.	
General:	Debug with some special classes. 1
v1.4.3.	
General:	Add <code>\Xnonbreakableafternumber</code> 38
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> 329
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. 199
v1.5.0.	
<code>\do@insidelinehook</code> :	Added <code>\do@insidelinehook</code> for use in <code>\do@line</code> 143
<code>\edindex</code> :	Compatibility with <code>imakeidx</code> package, and possibility to use multiple index with <code>\edindex</code> 285
General:	Correct indexing when the call is made in critical notes. 281
v1.5.1.	
<code>\managestanza@modulo</code> :	Correct <code>stanzaindentsrepetition</code> counter 294
<code>\normalvfootnoteX</code> :	Fix bug with normal familiar footnotes when mixing RTL and LTR text. 196
v1.6.0.	
<code>\newverse</code> :	Add <code>\falseverse</code> macro. 297
v1.6.1.	
<code>\AtEveryPstart</code> :	Spurious space in <code>\pstart</code> 135

\ifinserthangingsymbol: Hang verse is now not automatically flush right.	292
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill inside.	140
\pend: Spurious space in \pend.	136
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.	65
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts.	329
\get@index@command: Debug \get@index@command and compatibility with hyperref package.	284
\newhookcommand@series@reload: Debug \beforenotesX and \maxnotesX which did not work.	243
\prevpage@num: Correct \parafootsep when using with ledgroup.	172
General: Compatibility with parledgroup option ofeledpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	281
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	216
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.	281
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes.	248
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup.	277
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	329
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.	41
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 \@1. Consequently, \@l@reg becomes \@nl@reg.	102
\AtEveryPstart: New optional argument for \pstart, to execute code before it. . .	135
\edindex: Use correctly default index when imakeidx is loaded.	285
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	329
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar. .	83
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	269
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in eledmac, including needs for eledpar case.	271
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	81
\l@dunhbox@line: \do@line is split in more little commands.	140
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	243
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	243
\pend: New optional argument for \pend, to execute code after it.	136
\stanza: &can have an optional argument: content to be printed after.	297
\Stanza can have an optional argument: content to be printed before.	297
Add \newverse macro, \falseverse deprecated.	297
General: Add \ledinnernote and \ledouternote commands.	56
Add \Xendparagraph and related settings.	47
Add hyperlink to crossref (needs hyperref package).	52
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	281
New hooks: \Xafterrule and \afterruleX	46
New options for ragged-paragraph notes	44
New sectioning commands.	63
Optional arguments for \pstart and \pend.	18

v1.12.1.

\wrap@edcrossref: Fix spurious spaces.	259
--	-----

v1.12.2.

\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0) .	140
--	-----

v1.12.3.

\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	116
\flag@start send a error message when a \edtext is done without insert (note) .	116
\reledmac@error: Replaced error messages	75
General: Add macros for new messages since v0.7	74
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.	56
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	287
\@ssect: Debug \eledchapter in parallel with memoir	332
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	143
\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	84
\l@dgobblearg: \l@dgobblearg becomes \l@dgobbeloptarg	312
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	312
\resumenummering: Debug \resumenummering	85
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	259
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	116
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	243
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	207
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	240
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	47
Added widthliketwocolumns option	71
v1.13.1.	
\thepstart: Add \l@dzeropenalties in \pstart	135
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	81
General: Fix bug with normal footnotes, added by v1.13.0.	1
v1.13.3.	
General: Fix extra spaces with paragraphed footnotes, added by v1.13.0.	1
v1.13.4.	
General: Fix bug with index when memoir class is used without hyperref	1
v1.14.0.	
\edindex: Let \eledmac take advantage of \makeidx even when memoir class is used	285
General: Debug spurious characters before endnotes.	216
Delete previous override of \l@d@@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	289
Move gobbling command	81
Provide \@gobblefour	81
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref	
package.	335
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	332
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	122
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	206
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	117
\if@lemmacommand@: New boolean \iflemmacommand@.	127

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.	114
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.	122
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.	260
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph. . . .	123
v1.18.0.	
<code>\@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools + hyperref</code>	287
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code>	255
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes.	217
<code>\l@dnumppstartsl</code> : Add <code>\ifl@dprintingpages</code> and <code>\dprintingcolumns</code> for <code>eledpar</code>	81
<code>\print@line</code> : Compatibility with Lua \TeX RTL languages.	140
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code>	182
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns (<code>eledpar</code>)	181
General: Add <code>\Xpstarteverytime</code>	37
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.	37
v1.19.0.	
<code>\footsplitskips</code> : <code>\footsplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages.	160
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty. . .	262
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>	189
<code>\stanza</code> : <code>&</code> and <code>&</code> can be preceded by spaces.	297
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (fix bug added in 1.19.0).	262
General: Add <code>\Xendboxlinenum</code>	40
Add <code>\Xtwolines</code> and <code>\Xmorethantwolines</code> hooks	37
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>	287
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	217
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code>	284
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\Xledsetnormalparstuff</code>	211
<code>\n@num</code> : <code>\n@num@ref</code> deleted	109
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	109
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	242
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	242
<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>	213
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	239
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	118
Correct <code>\skipnumbering</code> for stanza.	118
Delete <code>\skipnumbering@reg</code>	118
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	239
Add <code>\applabel</code> and related	54
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column.	1
Add <code>\hidenumbering</code>	22
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks	43
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code>	37
Add <code>\Xparindent</code> and <code>\hangindentX</code>	42
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>	81
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	284
General: Add <code>\doendnotesbysection</code> command.	25
Add option for lemma separator inside endnotes	41
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>	172
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>	122
<code>\Serefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmoreethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code>	265
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	127
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	230
<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.	131
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code>	39
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code>	39
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.	278
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	269
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	279
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	329
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some code of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	279
<code>\newseries@</code> : One endnotes file by series.	235
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined	252
<code>\doxtrafeeti</code> becomes <code>\Xdo@feet</code> ; <code>\doxtrafeetii</code> becomes <code>\do@Xfeet</code> ; <code>\@opxtrafeeti</code> becomes <code>\@opfeetX</code> ; <code>\doreinxtrafeetii</code> becomes <code>\X@doreinfeet</code> ; <code>\doreinxtrafeeti</code> becomes <code>\@doreinfeetX</code>	252
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	192
Suppress indexing command specific to <code>memoir</code>	285
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.	211
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>bidir</code> v17.9 and following.	1
Warning message when using some setting commands inside <code>rightside</code> environment	
(deprecated behavior)	1
v2.3.1.	
General: Fix spurious space when using optional argument of <code>\stanza</code> (introduced in	
v2.3.0).	1
v2.4.0.	
<code>\footnoteoptions@</code> : First argument of <code>\footnoteoption@</code> is now mandatory, not	
optional.	157
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>	36

Option linerangesep for critical footnotes and endnotes.	36
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	264
\edlabel: Fix bug when calling \edlabel in a footnotes of the rightside	255
\l@d@section: \endnotes take six arguments.	217
\printlines: \printlines takes an eighth argument: the line flag	188
\xlineref: \xlineref does not include anymore the side flag. Use \xflagref to get it. Not that \edlineref still contains the flag.	260
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	54
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.	173
\l@d@section: \endnotes take seven arguments.	217
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	46
New hooks: \Xlemmafont and \Xendlemmafont	42

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>\Xhookgroup</code> hooks executed on columnar footnotes (moved to a larger group, to take effect).	1
v2.7.2.	
General: <code>\Xhsize</code> and <code>\hsizeX</code> become <code>\Xwidth</code> and <code>\widthX</code>	46
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.	217
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>	181

General: Add <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> .	1
Fix bug in <code>\ledlsnotefontsetup</code> and <code>\ledrsnotefontsetup</code> which could not handle <code>\color</code> command properly.	1
More specific error messages.	1
New hooks: <code>\Xwrapcontent</code> , <code>\Xendwrapcontent</code> and <code>\wrapcontentX</code> .	42
New hooks: <code>\Xwraplemma</code> and <code>\Xendwraplemma</code> .	42
v2.10.1.	
General: Add ‘nopenalties’ option.	1
Fix bug introduced in v. 1.4: not paragraphed critical footnotes could prevent marginal line number from being displayed	1
v2.11.0.	
<code>\do@actions@fixedcode</code> : Add action 1010	146
General: Add new tools to produce an apparatus of manuscripts	1
Fix bug in <code>\Xparafootsep</code> in parallel typesetting	1
Make <code>\parafootsepX</code> work	1
Prevent <code>\Xtxtbeforenotes</code> hook from causing notes to go beyond the bottom margin	1
v2.12.0.	
General: <code>\preXnotes</code> becomes <code>\Xprenotes</code> (naming convention)	1
Add <code>\hidenumberingonleftpage</code> and <code>\hidenumberingonrightpage</code>	1
Add <code>\toendnotes</code> and related.	1
Add <code>auxdir</code> option.	1
Fix bug in critical and familiar footnotes when using uppercase letters with accent mark	1
Fix bug when using <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting in combination with the <code>noeledsec</code> option.	1
Fix bug with <code>\ledinnernote</code> and <code>\ledouternote</code> in parallel typesetting	1
Fix bug with familiar footnote number in optional argument of <code>\pstart</code> or <code>\pend</code> in parallel typesetting	1
Fix spurious vertical space in <code>\chapter</code> when used as optional argument of <code>\pstart</code> in parallel typesetting.	1
Make endnote compatible with <code>\sameword</code> mechanism	1
More accurate message to control the position of <code>\Xfootnote</code> and <code>\applabel</code> in the \TeX code	1
v2.13.0.	
General: Version 2.13.0 never existed.	1
v2.13.1.	
General: In critical footnotes, the right side flag is printed only if requested explicitly with <code>\Xlineflag</code> (bug added in v. 2.5.0).	1
v2.13.2.	
General: Fix bug added in v. 11.2 which could make parallel typesetting not work.	1
v2.13.3.	
General: Makes <code>\Xendafterpagenumbe</code> affecting <code>\Serefwithpage</code>	1
v2.14.0.	
General: Hyperref with the line number inside critical footnotes is correct when using <code>\xxref</code>	1
Some internal changes for new features of <code>reledpar</code> .	1
v2.14.1.	
General: Fix bug when using <code>\footnoteX</code> in the first argument of <code>\edtext</code> .	1
v2.14.1a.	
General: Fix problematic typos in the handbook.	1

v2.15.0.	
General: Add ‘byline’ arrangement.	1
Fix <code>\Xtxtbeforenotes</code> in <code>ledgroup</code>	1
v2.15.1.	
General: Fix <code>\edindex</code> in tabular environments.	1
v2.15.2.	
General: Fix bug with <code>fancyhdr</code> package 3.8 and later.	1
v2.15.3.	
General: Fix bug with <code>\section</code> in optional argument of <code>\pstart</code> and empty line before <code>\pend</code> (bug added in v2.8.2).	1
Simplification of the sectioning command code.	1
v2.16.0.	
General: <code>\Xdo@feet</code> becomes <code>\do@Xfeet</code>	252
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1
Add <code>\Xpagelinesep</code> hook.	38
Compatibility with new features of <code>reledpar</code>	1
Deleting dead code.	1
Display warning message if using a version of \TeX that is too old.	1
Fix bug with <code>\Xgroupbylines</code> for notes in two columns	1
Fix bug with <code>\Xtxtbeforenotes</code> for notes in three or two columns	1
Fix bug with ‘notenumber’ option of <code>indextools</code> package when indexing texts in familiar footnotes.	1
Fix potential bug when using <code>\edindex</code> in critical footnotes.	1
More explicit error message when missing to define stanza indent.	252
New options of <code>\fnpos</code> and <code>\mpfnpos</code> to set a customized order for familiar and critical footnotes.	1
When <code>\edindex</code> is called outside of a <code>\beginnumbering... \endnumbering</code> structure, it is automatically switched to <code>\index</code> , with a warning message.	1
When indexing texts in familiar footnotes with <code>\edtext</code> , referring to the line number where the footnote is called.	1
When indexing texts in sidenotes with <code>\edtext</code> , referring to the line number where the sidenote is called.	1
v2.16.1.	
General: Fix bug with redefinition of the style of the footnote number (bug added in v2.12.0)	1
v2.16.2.	
General: Error message if <code>footmisc</code> is loaded after <code>reledmac</code>	1
Fix bug introduced by v2.16.1 when using not expandable control sequence, like <code>\normalfont</code> , in the footnote number style.	1
v2.16.3.	
General: Fix bug with <code>\Seref</code> (bug added in v2.7.0).	1
v2.16.4.	
General: Fix bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.15.3).	1
v2.16.5.	
General: Fix potential spurious spaces in endnotes.	1

v2.16.6.

General: Fix bug with the line number style in `\doenotes` when referring to right side
line in parallel typesetting. 1
Take account of `\linenumberstyle` when using `\edlineref`. 1