

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	14
1.2.3 eledmac	14

*This file (**reledmac.dtx**) has version number v2.26.7, last revised 2018/04/07.

[†]maieul at maieul dot net

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

6.3.2 Case setting	29
6.3.3 Notes about input encoding with UTF-8 processor	29
6.3.4 Use with \lemma command	30
6.3.5 Sameword for a group of words	31
6.3.6 Customizing	32
6.3.7 Problems with some macros	32
6.3.8 Automatic sameword annotation	33
6.4 Apparatus of manuscripts	33
6.4.1 Marking sections of text	34
6.4.2 Layout of the apparatus of manuscripts	34
6.4.3 Settings	35
6.5 Familiar notes	35
6.5.1 Basic use	35
6.5.2 Customizing mark	35
6.5.3 Separator for multiple footnotes	36
6.6 Printing the footnote mark without printing the footnote text	36
6.7 Changing series	36
6.7.1 Create a new series	36
6.7.2 Delete series	36
6.7.3 Series order	36
6.8 Position of critical and familiar footnotes	37
7 Apparatus customization	38
7.1 Notes arrangement in a series	38
7.2 Control line number printing	39
7.2.1 Print line number only at first time	39
7.2.2 Print page number only at first time	39
7.2.3 Arbitrary text before line number	40
7.2.4 Separator for line range	40
7.2.5 Abbreviate line range	40
7.2.6 Disable line number	41
7.2.7 Printing pstart number	41
7.2.8 Printing stanza number	42
7.2.9 Separator between line and subline numbers	42
7.2.10 Separator between page and line numbers	42
7.2.11 Space around number	42
7.2.12 Space around line symbol	42
7.2.13 Space in place of number	43
7.2.14 Boxing line number and line symbol	43
7.3 Arbitrary code around line number	44
7.4 Separator between the lemma and the note	44
7.4.1 For footnotes	44
7.4.2 For endnotes	44
7.5 Font style	45
7.5.1 For line number	45
7.5.2 For the lemma	45

7.5.3 For all notes	45
7.6 Wrapping notes	46
7.6.1 Wrapping lemmas	46
7.6.2 Wrapping contents	46
7.7 Indent of notes content	46
7.8 Arbitrary code at the beginning of notes	46
7.9 Arbitrary code before inserting note	47
7.10 Options for footnotes in columns	47
7.10.1 Alignment	47
7.10.2 Size of the columns	47
7.11 Options for paragraphed footnotes and notes grouped by line	48
7.11.1 Mark separation of notes	48
7.11.2 Ragged text	48
7.12 Options for block of notes	48
7.12.1 Grouping notes by line	48
7.12.2 Text before notes	49
7.12.3 Code before notes	49
7.12.4 Spacing	49
7.12.5 Rule	49
7.12.6 Maximum height	50
7.12.7 Width	50
7.13 Footnotes and the <code>reledpar</code> columns	50
7.14 Line number annotation	50
7.15 Endnotes in one paragraph	51
8 Fonts	51
9 Verse	51
9.1 Basic	51
9.2 Define stanza indents	52
9.3 Repeating stanza indents	52
9.4 Manual stanza indent	53
9.5 Stanza breaking	53
9.6 Hanging symbol	53
9.7 Long verse and page break	54
9.8 Content before/after verses	54
9.9 Numbering stanza	55
9.10 Adding annotations to line numbers	55
9.10.1 Modifying annotation associated with a specific note	57
9.10.2 Changing the position of the annotation	57
9.10.3 Changing the macro in which annotations are wrapped	57
9.11 Various tools	58
9.12 Notes on empty lines	58
10 Grouping	58

11 Cross referencing	59
11.1 Basic use	59
11.2 Cross-referencing to a critical note	60
11.3 Cross-referencing which return a number in any case	60
11.3.1 Cross-referencing in order to define line number of a critical note	60
11.4 Not automatic cross-referencing	61
11.5 Normal \LaTeX cross-referencing	61
11.6 References to start and end lines	61
11.6.1 Reference to main text lines	61
11.6.2 References to lines that are commented on in the apparatus	62
11.6.3 Settings	62
11.7 Compatibility with <code>xr</code> package	64
12 Sidenotes	64
12.1 Basics	64
12.2 Setting	64
12.2.1 Width	64
12.2.2 Vertical position	65
12.2.3 Distance to the main text	65
12.2.4 Font	65
12.2.5 Separator between notes	65
13 Indexing	65
13.1 Basics	65
13.2 Use with <code>imakeidx</code> or <code>indextools</code>	65
13.3 Referring to critical notes	66
13.4 Separator between page and line numbers	66
13.5 Using <code>xindy</code>	67
13.6 Advanced setting	67
14 Glossary	67
14.1 Preamble setting	68
14.2 Commands	68
15 Tabular material	68
16 Sectioning commands	71
16.1 Sectioning commands without line numbers or critical notes	71
16.2 Sectioning commands with line numbering and critical notes	72
16.3 Optimization	72
17 Quotation environments	73
18 Page breaks	73
18.1 Control page breaking	73
18.2 Prevent page break in a long verses	73

19 Miscellaneous	74
19.1 Known and suspected limitations	74
19.1.1 Non-standard geometry	74
19.1.2 floatrow package compatibility	74
19.1.3 ‘No room for a new’	74
19.1.4 Marginal notes	75
19.1.5 Paragraph shape	75
19.1.6 Paragraphed footnotes	75
19.1.7 Use with other packages	76
19.2 Parallel typesetting	77
I Implementation overview	78
II Preliminaries	78
II.1 Links with original edmac	78
II.2 Package declaration	78
II.3 Package options	79
II.3.1 Options of reledpar	79
II.3.2 Options of reledmac	79
II.4 Loading packages	81
II.5 Compatibility with LuaTeX	81
II.6 Boolean flags	82
II.7 Messages	83
II.8 Gobbling	89
II.9 Wrapping content	89
II.10 Miscellaneous commands	90
II.11 Prepare reledpar	90
II.12 Booleans provided by other optional packages which are required in any case	91
III Sectioning commands	91
IV List macros	96
V Line counting	98
V.1 Choosing the system of lineation	98
V.2 Line number margin	99
V.3 Line number initialization and increment	100
V.4 Line number locking	102
V.5 Line number style	103
V.6 Line number printing	103
V.7 Line number counters and lists	104
V.8 Line number locking counter	106
V.9 Line number associated to lemma	106
V.10 Reading the line-list file	109
V.11 Commands within the line-list file	112

V.12 Writing to the line-list file	125
VI Marking text for notes	133
VI.1 <code>\edtext</code> itself	133
VI.2 Substitute lemma	141
VI.3 Substitute line numbers	142
VI.4 Lemma disambiguation	143
VII Paragraph decomposition and reassembly	150
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	150
VII.2 Processing one line	157
VII.2.1 General process	157
VII.2.2 Process for “normal” line	158
VII.2.3 Process for line containing <code>\eledsection</code> command	159
VII.2.4 Hooks	160
VII.2.5 Sidenotes and marginal line number initialization	161
VIII Line and page number computation	162
VIII.1 Continuous line numbering between parallel typesetting and normal typesetting	165
IX Line number annotation	166
X Line number printing	168
XI Pstart number printing inside	172
XII Restoring footnotes and penalties	173
XII.1 Add insertions to the vertical list	173
XII.2 Penalties	175
XII.3 Printing leftover notes	176
XII.4 Text before notes	177
XIII Critical footnotes	179
XIII.1 Fonts	179
XIII.2 Individual note options	179
XIII.3 Notes language	180
XIII.4 General survey of the way we manage notes	181
XIII.5 General setup	181
XIII.6 Footnotes arrangement	182
XIII.6.1 User level macro	182
XIII.6.2 Normal footnote	183
XIII.6.3 Paragraphed footnotes	188
XIII.6.4 Columnar footnotes	195
XIII.7 Critical notes presentation	202
XIII.7.1 Font tools	202
XIII.7.2 Pstart number in footnote	203

XIII.7.3 Lemma printing	203
XIII.7.4 Line number printing	204
XIII.7.5 Footnote grouped by line	214
XIV Familiar footnotes	215
XIV.1 Adjacent footnotes	215
XIV.2 Regular footnotes for numbered texts	217
XIV.3 Footnote formats	219
XIV.4 Footnote arrangement	219
XIV.4.1 User level macro	219
XIV.4.2 Normal footnotes	220
XIV.4.3 Two columns footnotes	225
XIV.4.4 Three columns footnotes	228
XIV.4.5 Paragraphed footnotes	230
XIV.5 Wrapping footnote marks in hyperlink	234
XV Code common to both critical and familiar footnote in normal arrangement	235
XVI Footnotes' width for two columns	236
XVII Footnotes' order	237
XVIII Footnotes' rule	237
XIX Specific skip for first series of footnotes	238
XIX.1 Overview	238
XIX.2 User level command	238
XIX.3 Internal commands	239
XX Endnotes	240
XX.1 Internal commands	240
XX.2 User level commands	244
XX.2.1 Inserting contents to endnotes	244
XX.2.2 Printing endnotes	245
XXI Generate series of notes	252
XXI.1 Test if series is still existing	253
XXI.2 Init specific to <code>reledpar</code>	253
XXI.3 For critical footnotes	253
XXI.3.1 Options	254
XXI.3.2 Create inserts, needed to add notes in foot	255
XXI.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc.	255
XXI.3.4 Set standard display	258
XXI.4 For familiar footnotes	258
XXI.4.1 Options	258

XXI.4.2 Create tools for familiar footnotes (<code>\footnoteX</code>)	259
XXI.5 The endnotes	261
XXI.5.1 The auxiliary file	261
XXI.5.2 The main macro	262
XXI.5.3 Tools	262
XXI.5.4 Internal commands	263
XXI.5.5 The options	263
XXI.6 Init standards series (A,B,C,D,E)	265
XXII Setting series display	265
XXII.1 Change series order	265
XXII.2 Test series order	265
XXII.2.1 Get the first series	266
XXII.3 Series setting	266
XXII.3.1 General way of working	266
XXII.3.2 Tools to set options	266
XXII.3.3 Tools to generate options commands	268
XXII.3.4 Options for critical notes	269
XXII.3.5 Options for familiar notes	271
XXII.3.6 Options for endnotes	272
XXII.4 Hooks for a particular footnote	274
XXII.5 Alias	274
XXIII Output routine	275
XXIII.1 Extra footnotes output	275
XXIII.2 Patching standard output's commands	278
XXIV Page numbering in parallel typesetting	281
XXV Cross referencing	283
XXV.1 Compatibility with <code>xref</code>	299
XXVI Sidenotes	299
XXVII Minipages and such	307
XXVIII Indexing	312
XXVIII.1 Looking on package order	312
XXVIII.2 Auxiliary macros for <code>\edindex</code>	313
XXVIII.3 Code specific to <code>\edindexin</code> critical footnotes	314
XXVIII.4 Analysis of command in indexed text	315
XXVIII.5 Code for the formatted index	315
XXVIII.6 Main code	316
XXVIII.7 Hyperlink	318
XXVIII.8 'innote' and 'notenumber' option of <code>indextols</code> package	320
XXIX Glossaries	321

XXX Verse	323
XXX.1 Hanging symbol management	323
XXX.2 Using & character	324
XXX.3 Code category setting	324
XXX.4 Stanza count and indent	324
XXX.5 Numbering stanza	326
XXX.6 Stanza number in note	327
XXX.7 Main work	328
XXX.8 Restore catcode and penalties	331
XXXI Apparatus of Manuscripts	331
XXXI.1 User level macro	331
XXXI.2 Setting macro	332
XXXI.3 Counters and lists	333
XXXI.4 Auxiliary file macros	333
XXXI.5 Action macro	334
XXXI.6 Inserting footnote	339
XXXI.7 Other	339
XXXII Arrays and tables	340
XXXII.1 Preamble: macro as environment	340
XXXII.2 Tabular environments	343
XXXII.2.1 Disabling and restoring commands	344
XXXII.2.2 Counters, boxes and lengths	347
XXXII.2.3 Tabular typesetting	351
XXXII.2.4 Environments	362
XXXIII Quotation's commands	362
XXXIV Section's title commands	364
XXXIV.1 Commands to disable some feature	364
XXXIV.2 General overview	364
XXXIV.3 \beforeeledchapter command	365
XXXIV.4 Auxiliary commands	365
XXXIV.5 Patching standard commands	366
XXXIV.6 Main code of \eledxxx commands	371
XXXIV.7 Macros written in the auxiliary file	374
XXXV Page breaking or no page breaking depending of specific lines	376
XXXVI Long verse: prevents being separated by a page break	378
XXXVII Tools for hyperref package	378
XXXVIII Compatibility with eledmac	379

Appendix A Things to do when changing versions	382
Appendix A.1 Migrating from edmac to ledmac	382
Appendix A.2 Migration from ledmac to eledmac	383
Appendix A.3 Migration to eledmac 1.5.1	384
Appendix A.4 Migration to eledmac 1.12.0	384
Appendix A.5 Migration to eledmac 17.1	385
Appendix A.6 Migration to eledmac 1.21.0	385
Appendix A.6.1 \Xledsetnormalparstuff and \ledsetnormalparstuffX	385
Appendix A.6.2 Endnotes	385
Appendix A.7 Migration to eledmac 1.22.0	385
Appendix A.8 Migration to eledmac 1.23.0	385
Appendix A.9 Migration from eledmac to reledmac	386
Appendix A.9.1 Risk of ‘no room for a new’	386
Appendix A.9.2 Multiple indices with memoir	386
Appendix A.9.3 Deprecated commands and options	386
Appendix A.9.4 \renewcommand replaced by command	387
Appendix A.9.5 Commands the names of which have been changed	387
Appendix A.9.6 Endnotes	389
Appendix A.9.7 Z Series	389
Appendix A.9.8 Internal commands	389
Appendix A.10 Migration to reledmac 2.1.0	389
Appendix A.11 Migration to reledmac 2.1.3	389
Appendix A.12 Migration to reledmac 2.3.0	389
Appendix A.13 Migration to reledmac 2.4.0	390
Appendix A.14 Migration to reledmac 2.5.0	390
Appendix A.15 Migration to reledmac 2.7.0	390
Appendix A.16 Migration to reledmac 2.7.2	390
Appendix A.17 Migration to reledmac 2.8.0	390
Appendix A.18 Migration to reledmac 2.13.1	390
Appendix A.19 Migration to reledmac 2.18.0	391
Appendix A.20 Migration to reledmac 2.21.0	391
Appendix A.21 Migration to reledmac 2.24.0	391
Appendix A.22 Migration to reledmac 2.26.0	391
References	392
Index	392
Change History	443

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 `reledmac` will take care of the formatting and visual correlation of all the disparate types of information.

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

Any suggestions for new features are welcome.

This manual contains a general description of how to use `reledmac` followed by the complete source code and its extensive documentation (in sections I and following, enumerated with Roman numerals). It ends with a list of actions to do when migrating from one version to other, a change history and an index to the source code.

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

1.2 History

1.2.1 `edmac`

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

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

The present version was developed in the summer of 1990, with the intent of adding necessary features, streamlining and documenting the code, and further generalizing it

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

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

As of 1994, we were pleased to be able to say that edmac was being used for the real-life book production of several interesting editions, such as the Latin texts of Euclid's *Elements*,² an edition of the letters of Nicolaus Copernicus,³ Simon Bredon's *Arithmetica*,⁴ a Latin translation by Plato of Tivoli of an Arabic astrolabe text,⁵ a Latin translation of part II of the Arabic *Algebra* by Abū Kāmil Shujā' b. Aslam,⁶ the Latin *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

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

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

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

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

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

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

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

⁸Geert H. M. Claassens, *De Middelnederlandse Kruisvaartromans*, (Amsterdam: Schiphower 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.

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. 38). These changes can trigger small problems with the old customization. That is why a new name was selected: `eledmac` (extended `ledmac`).

To migrate from `ledmac` to `eledmac`, please read Appendix A.2 p. 383.

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. 386.

1.3 Bibliography

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

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

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

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

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

3 Compatibility warning

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

Please read 7.12.6 p. 50.

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. 387).

swcaseinsensitive make \sameword command case insensitive.

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. 73).

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 (13.5 p. 67).

4.2 Optimizing package performance

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

noledsec disables tools for \eledsection and related commands (16.2 p. 72).

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 reledmac 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. 58). 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 `\autopar` command needs to be called inside a `\beginnumbering... \endnumbering` structure, as follows:

```

\beginnumbering
\autopar

A paragraph of numbered text.

Another paragraph of numbered
text.

\endnumbering

```

`\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`.

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

Note that a `\noindent` will be automatically added before this argument, and, consequently, a `\parskip` will be inserted. You can use a second optional argument in brackets to avoid that `\noindent`.

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

The second optional argument of `\pstart` / `\pend` replaces 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. 51) or `reledpar` (see 19.2 p. 77).

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

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

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

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

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

```
\AtStartEveryPstart{*}
```

Instead of manually doing

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

5.2.5 Numbering paragraphs (`\pstart`)

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

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

On each `\beginnumbering` the numbering restarts.

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

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

5.2.6 Languages written in Right to Left

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

5.2.7 Memory limits

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

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

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

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

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

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

5.3.2 Setting lineation start and step

`\firstlinenum`
`\linenumincrement`

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

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

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

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

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

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

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

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

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

5.3.3 Setting lineation reset

`\lineation`

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

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

5.3.4 Setting line number margin

`\linenummargin`

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

```
\linenummargin{left}
```

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

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

5.3.5 Other settings

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

5.4 Changing the line numbers

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

5.4.1 Sublineation

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

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

`Xsublinesep` You can change the separator between the line number and the subline number ei-
`Xsublinesepside` ther by using `Xsublinesep` without any optional argument (7.2.9 p. 42) or by using `Xsublinesepside`. In the second case, it will change the separator only for line numbers in the margins, not in the footnotes.

5.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current
`\endlock` value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 51.

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

5.4.3 Setting and changing line number

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

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

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

5.4.4 Line number style

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

Alph Uppercase letters (A ... Z).

alph Lowercase letters (a ... z).

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

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

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

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

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

5.4.5 Skipping and hiding number

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

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

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

5.5 Executing code at each line

`\dolinehook`
`\doinsidelinehook`

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

6 Apparatus commands

6.1 Terminology

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

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

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

So :

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

6.2 Critical notes

6.2.1 The lemma

`\edtext`

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

`\edtext{⟨lemma⟩}{⟨commands⟩}`

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

For example:

I am happy :	1	I am happy : I saw my friend Smith on
I saw my friend <code>\edtext{Smith}{</code>	2	Tuesday.
<code>\Afootnote{Jones C, D.}}</code>		
on Tuesday.		
		<hr style="width: 20%; margin-left: 0;"/>
	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:

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

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 6.7.1 p. 36.

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. 40).
- `nonum` disables line numbering for this note. A horizontal blank space is added instead. You can use `\Xinplaceoflemmaseparator` to set it (7.4.1 p. 44).
- `nosep` to disable the lemma separator for this note.
- `linerangesep=<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:

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

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

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 7.2.5 p. 40).
- `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 (7.4.2 p. 45).
- `linrangesep= $\langle c \rangle$` to change to $\langle c \rangle$ 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{ $\langle s \rangle$ }`, where $\langle s \rangle$ 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 $\langle s \rangle$ 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{ $\langle s \rangle$ }`. For each value of $\langle s \rangle$, 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 (7.4.2 p. 44).

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:

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

For example:

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

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

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

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

```
\Attoendnotes*{\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{⟨alternative⟩}` within the second argument to `\edtext` and before the note commands. The most common use of this command is to abbreviate the lemma that’s printed in the notes. For example:

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

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

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.

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

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

This command does not change the marginal line numbers in any way; it just changes the numbers passed to the notes. Its use comes in situations that `\edtext` has trouble dealing with for whatever reason. If you need notes for overlapping passages that aren't nested, for instance, you can use `\lemma` and `\linenum` to generate such notes despite the limitations of `\edtext`. If the *⟨lemma⟩* argument to `\edtext` is extremely long, you may run out of memory; here again you can specify a note with an abbreviated lemma using `\lemma` and `\linenum`. The numbers used in `\linenum` need not be entered manually; you can use the 'x-' symbolic cross-referencing commands below (11 p. 59) 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

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

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

with the `\sameword` command:

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

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

6.3.2 Case setting

By default, `\sameword` is sensitive to the case. E.g. “foo” is considered as a different word to “Foo”.

You can use the `swcaseinsensitive` when loading the package to make `\sameword` insensitive to the case, i.e. to consider “foo” as identical “Foo”.

6.3.3 Notes about input encoding with UTF-8 processor

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

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

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

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

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

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

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

handbook of `xargs` to know more about `\newcommandx`.

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

6.3.4 Use with `\lemma` command

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

For example, in the following example:

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

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

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

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

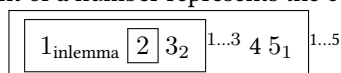
Note that only words that are actually referenced in a `\lemma` need the optional argument. Therefore, the first `\sameword` in the example above should have “1” as its optional argument, to be referenced correctly in the lemma.

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

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

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

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



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)
1	A ¹ ...A ⁵] level 1
1	A ¹ ...A ³] level 2
1	A ² (2)] level 3

6.3.5 Sameword for a group of words

Sometimes a group of words and not only a single word, occurs multiple times. In this case, you have two possibilities.

First, you can consider only the individual words, and not groups of word. For example:

```
\sameword{per} \sameword{causam}
tamen scire
\edtext{\sameword{causam}}{\Bfootnote{fnote}}
est
```

```
\edtext{\sameword{per} \sameword{causam}}{\Bfootnote{causam rei B}}
cognoscere
\edtext{\sameword{causam}}{\Bfootnote{fnote}}
```

1 per causam tamen scire causam est per causam cognoscere causam

```
1 causam2] fnote
1 per2 causam3] causam rei B
1 causam4] fnote
```

In this case it is clear which “per causam” is meant.

However, in the case that “per causam” is the lemma of the second note, there should be only one number for that whole lemma. In this case we can mark all “per causam” groups. But as “causam” is also called as lemma in note 1 and 3, we need to use nested \sameword commands. Consequently, we need to use \lemma for the \edtext linked to “per causam”, as we don’t want to number each individual word.

```
\sameword{per \sameword{causam}} tamen scire
\edtext{\sameword{causam}}{\Bfootnote{fnote}} est
\edtext{\sameword[1]{per \sameword{causam}}}{\lemma{\sameword{per causam}}\Bfootnote{causam
\edtext{\sameword{causam}}{\Bfootnote{fnote}}}
```

1 per causam tamen scire causam est per causam cognoscere causam

```
1 causam2] fnote
1 per causam2] causam rei B
1 causam4] fnote
```

6.3.6 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}%
}
```

6.3.7 Problems with some macros

\swnoexpands Macros inside \sameword that are not fully expandable, mainly macros which manipulate font features, write on full or have optional argument, may cause problems during compilation. Custom commands inside \sameword may therefore result in errors saying that “Use of sameword doesn’t match its definition.” To solve this, include a redefinition of your custom commands in the \swnoexpands macro. In order to not include any content of a macro during comparison, identify the command with \@gobble For example:


```

\makeatletter
\appto{\swnoexpands}{%
  \let\somemacro\@gobble%
}
\makeatother

```

This will drop the content of `\somemacro` during comparison.

To include the content of the first and only one argument of a custom command in `sameword` comparison, use the `\@firstofone` command. For example, this is how `\emph` is handled:

```

\makeatletter
\appto{\swnoexpands}{%
  \let\emph\@firstofone%
}
\makeatother

```

To include command which can take optional argument, use `\RenewExpandableDocumentCommand` of `\xparse`. For example, this is how `\edindex` is handled:

```

\makeatletter
\appto{\swnoexpands}{%
  \RenewExpandableDocumentCommand{\edindex}{om}{}%
}
\makeatother

```

6.3.8 Automatic sameword annotation

All potentially ambiguous apparatus entries must be annotated manually. That annotation process is laborious and includes a risk of errors. *Samewords* is a Python script that can automate this step of the process. It can be installed via the *Python Package Index*, but see <https://github.com/stenskjær/samewords> for more info and documentation. The script is still at a beta stage, so comments and questions as well as error reports are very welcome at <https://github.com/stenskjær/samewords/issues>.

Please note that the maintainer of this script is not identical with the maintainer of `reledmac`.

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 T_EX 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 T_EX two or more times. We strongly recommend to use tools like *latexmk* to do that.

6.4.1 Marking sections of text

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

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

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

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

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

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

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

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

6.4.2 Layout of the apparatus of manuscripts

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

- If the section does not start on the current page, the starting line number will be the line number of the first line on the page.
- If the section does not stop on the current page, the ending line number will be the line number of the last line on the page.

- If the section neither starts nor ends on the current page, no line number will be printed. The same is true in case both `\msdata` is called at the very beginning of the page and `\endmsdata` is called at the very end of the page.

6.4.3 Settings

As the apparatus of manuscripts technically consists of first-level critical notes ('A' series), any setting available for critical notes can be applied (7 p. 38). However, the following *additional* commands are available.

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

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

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

6.5 Familiar notes

6.5.1 Basic use

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

6.5.2 Customizing mark

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

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

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

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

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

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

There are similar command triples for the other series.

6.5.3 Separator for multiple footnotes

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

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

```
\providecommand*\multfootsep{\textsuperscript{\normalfont,}}
```

and can be changed if necessary.

6.6 Printing the footnote mark without printing the footnote text

`\footnoteXmark` `\footnoteXtext` In certain cases, you can't directly use `\footnoteX`; for example, when using `\uline` command of the `ulem` package. You need to print the footnote mark first, then call the footnote text to be inserted.

For all $\langle X \rangle$ command, `reledmac` provides a `\footnote\langle X \rangle mark` command and a `\footnote\langle X \rangle text` command, equivalent to standard \LaTeX 's command `\footnotemark` and `\footnotetext`. For example, to use with `\uline`, do:

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

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

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

6.7 Changing series

6.7.1 Create a new series

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

6.7.2 Delete series

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

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

6.7.3 Series order

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

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

6.8 Position of critical and familiar footnotes

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

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

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

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

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

`<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 `ℒTEX`.

7 Apparatus customization

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

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

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

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

There is also name convention:

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

7.1 Notes arrangement in a series

`\Xarrangement`
`\arrangementX`

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

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

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

- `paragraph` formats all of the footnotes of a series as a single paragraph; if you use this arrangement, you are strongly encouraged to read 19.1.6 p. 75.
- `twocol` formats them as separate paragraphs, but in two columns;
- `threecol`, in three columns.
- `normal`, restore normal arrangement.

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

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

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

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

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

7.2 Control line number printing

7.2.1 Print line number only at first time

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., once for line 1, once for line 2, etc.), you can use `\Xnumberonlyfirstinline[⟨s⟩]`.

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.

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.

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

7.2.2 Print page number only at first time

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

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

2. The ending page number of the lemma is the same as the starting page number of the lemma.

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

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

In any case, you can use:

- | | |
|---------------------------------------|--|
| <code>\Xendsympagenum</code> | • <code>\Xendsympagenum[⟨series⟩]{⟨c⟩}</code> to print $\langle c \rangle$ when the page number is not printed. |
| <code>\Xendinplaceofpagenumber</code> | • <code>\Xendinplaceofpagenumber[⟨series⟩]{⟨l⟩}</code> to print a $\langle l \rangle$ length horizontal space in case no symbol is printed instead of the page number. |

7.2.3 Arbitrary text before line number

- | | |
|------------------------------------|---|
| <code>\Xbeforenumber</code> | <code>\Xbeforenumber[⟨s⟩]{⟨txt⟩}</code> allows the insertion of $\langle txt \rangle$ before the line number only when the line number is printed, so taking into account <code>\Xnumberonlyfirstinline</code> and the like. |
| <code>\Xendbeforepagenumber</code> | For endnotes <code>\Xendbeforepagenumber[⟨s⟩]{⟨text⟩}</code> defines the text before the page number in endnotes. Default value is p. (“p” followed by a dot). |
| <code>\Xendafterpagenumber</code> | <code>\Xendafterpagenumber[⟨s⟩]{⟨text⟩}</code> defines the text after the page number in endnotes. Default value is) (open parenthesis followed by a single space). |
| <code>\Xendlineprefixsingle</code> | <code>\Xendlineprefixsingle[⟨s⟩]</code> defines the text before the line number in endnotes when there is only one line. Default value is empty. |
| <code>\Xendlineprefixmore</code> | <code>\Xendlineprefixmore[⟨s⟩]{⟨text⟩}</code> defines the text before the line number in endnotes when there is more than one line. Default value is empty. If you don’t define it, it will use the value defined by <code>\Xendlineprefixsingle</code> . |

7.2.4 Separator for line range

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

7.2.5 Abbreviate line range

- | | |
|---------------------------------|--|
| <code>\Xtwolines</code> | If a lemma is printed on two subsequent lines, <code>reledmac</code> 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)”. |
| <code>\Xmorethantwolines</code> | |

To achieve this, use `\Xtwolines[⟨s⟩]{⟨text⟩}` and `\Xmorethantwolines[⟨s⟩]{⟨text⟩}`. The $\langle text \rangle$ argument of `\Xtwolines` will be printed if the lemma is on two lines, and

the $\langle text \rangle$ 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 $\langle text \rangle$ 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[$\langle series \rangle$]`.

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

However, you can force print the final page number with `\Xtwolinesonlyinsamepage[$\langle series \rangle$]`.

You can disable `\Xtwolines` and related for a specific note by using the ‘[fulllines]’ argument in the note macro cf. 6.2.2 p. 25.

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

7.2.6 Disable line number

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

7.2.7 Printing pstart number

`\Xpstart` You can use `\Xpstart[$\langle s \rangle$]` 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[$\langle s \rangle$]`. In this case, the pstart number will be printed every time in footnote.

`\Xonlypstart` In combination with `\Xpstart`, you can use `\Xonlypstart[$\langle s \rangle$]` 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 changes the separator in side numbering.

`\Xendsublinesep`

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

However, it does not change anything for the separator in side numbering. 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, neither is the space. The default value is 0.5 em.

`\Xendbeforenumber`

`\Xendafternumber`

`\Xendbeforenumber` and `\Xendafternumber` are the equivalents of `\Xbeforenumber` and `\Xafternumber` 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

`\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 value set by `\Xbeforenumber`.

`\Xaftersymnumber`

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

`\Xendbeforenumber`

`\Xendaftersymnumber`

`\Xendbeforenumber` and `\Xendaftersymnumber` are the equivalents of

`\Xbeforesymmlinenumber` and `\Xaftersymmlinenumber` for the endnotes.

7.2.13 Space in place of number

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

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

7.2.14 Boxing line number and line symbol

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

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

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

`\Xendboxsymmlinenumber` `\Xendboxsymmlinenumber[\langle s \rangle]{\langle l \rangle}` is the same as `\Xboxsymmlinenumber` but for endnotes.

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

L to align left (default value);

R to align right;

C to center.

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

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

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

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

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

7.3 Arbitrary code around line number

`\Xendbhooklinenum` `\Xendbhooklinenum[⟨s⟩]{⟨code⟩}` is used to execute code before line numbers in endnotes. The code is executed before the `\Xendbeforelinenum` space and before the `\Xendnotenumfont` font setting.
`\Xendahooklinenum` `\Xendahooklinenum[⟨s⟩]{⟨code⟩}` is used to execute code after line number in endnotes. The code is executed after the `\Xendafternumber` space.
`\Xendbhookinplaceofnumber` `\Xendbhookinplaceofnumber[⟨s⟩]{⟨code⟩}` is used to execute code before space or symbol which replace line number in endnotes. The code is executed before the `\Xendbeforeysmlinenum` space and before the `\Xendnotenumfont` font setting.
`\Xendahookinplaceofnumber` `\Xendahookinplaceofnumber[⟨s⟩]{⟨code⟩}` is used to execute code after space or symbol which replace line number in endnotes. The code is executed after the `\Xendaftersym` space.

7.4 Separator between the lemma and the note

7.4.1 For footnotes

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

7.4.2 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`.

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

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

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

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

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

7.5 Font style

7.5.1 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`.

7.5.2 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.

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

`\Xlemmafont` 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.

7.5.3 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`.

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

7.6 Wrapping notes

7.6.1 Wrapping lemmas

<code>\Xwraplemma</code>	<code>\Xwraplemma[⟨s⟩]{⟨cmd⟩}</code> is used to wrap, in the footnote, the lemma in a \TeX command. For example, with the <code>bid</code> package, to ensure having a lemma written right to left, use <code>\Xwraplemma{\RL}</code> .
<code>\Xwrapendlemma</code>	<code>\Xendwraplemma[⟨s⟩]{⟨cmd⟩}</code> is the same for endnotes.

7.6.2 Wrapping contents

<code>\Xwrapcontent</code>	<code>\Xwrapcontent[⟨s⟩]{⟨cmd⟩}</code> 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 <code>\Xwrapcontent{\foreignlanguage{⟨language⟩}}</code> (with <code>babel</code>) or <code>\Xwrapcontent{\text{⟨language⟩}}</code> (for <code>babel</code>).
<code>\Xendwrapcontent</code>	<code>\Xendwrapcontent[⟨s⟩]{⟨cmd⟩}</code> is the same for endnotes.
<code>\wrapcontentX</code>	<code>\wrapcontentX[⟨s⟩]{⟨cmd⟩}</code> is the same for familiar footnotes.

7.7 Indent of notes content

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

7.8 Arbitrary code at the beginning of notes

The three next commands add arbitrary code at the beginning of notes. As the name's space is local to the notes, you can use it to redefine some style inside the notes. For

example, if you don't want the pstart number to be in bold, use :

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

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

7.9 Arbitrary code before inserting note

`\Xbeforeinserting` `\beforeinsertingX[⟨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 (7.6.1 p. 46)).

Note that the changes are local to the footnote.

7.10 Options for footnotes in columns

7.10.1 Alignment

`\Xcolalign` `\colalignX[⟨s⟩]{⟨code⟩}` 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`).

7.10.2 Size of the columns

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

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

`\Xhsizethreecol` `\Xhsizethreecol[⟨s⟩]{⟨l⟩}` is used to change width of a column when critical

notes are displaying in three columns. Default value is `.3 \hspace`.

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

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

7.11 Options for paragraphed footnotes and notes grouped by line

7.11.1 Mark separation of notes

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

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

`\parafootsepX` 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.

7.11.2 Ragged text

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

`\raggedX` Text in paragraphed footnotes is justified, but you can use `\raggedX[\langle s \rangle]{L}` if you want it to be ragged left, or `\raggedX[\langle s \rangle]{R}` if you want it to be ragged right.

7.12 Options for block of notes

7.12.1 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[\langle series \rangle]`.

In many cases, you might like to use it in combination with `\Xnumberonlyfirstinline` (7.2.1 p. 39).

`\Xgroupbylineseparetwolines` Note that the `\Xafternote` and `\Xparafootsep` settings are used to determine space and content between footnotes (7.11 p. 48). 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[\langle s \rangle]`.

In many cases, you might like to use it in combination with `\Xnumberonlyfirstintwolines` (7.2.1 p. 39).

7.12.2 Text before notes

`\Xtxtbeforenotes` You can add text before critical footnotes with `\Xtxtbeforenotes[⟨s⟩]{⟨text⟩}`.
`\txtbeforenotesX` You can add text before familiar footnotes with `\txtbeforenotesX[⟨s⟩]{⟨text⟩}`.
`\Xendtxtbeforenotes` You can add text before endnotes with `\Xendtxtbeforenotes[⟨s⟩]{⟨text⟩}`. The text will be typeset only if there are endnotes.
`\Xtxtbeforenotesonlyonce` By default, such texts are inserted at the beginning of the groups of notes on each page. You can add `\Xtxtbeforenotesonlyonce` (for critical footnotes) and `\txtbeforenotesonlyonceX` (for familiar footnotes) to insert them only the first time notes are typeset.

7.12.3 Code before notes

`\Xbhookgroup` While `\Xtxtbeforenotes` is for typesetting text before notes, `\Xbhookgroup[⟨s⟩]{⟨code⟩}`
`\bhookgroupX` and `\bhookgroupX[⟨s⟩]{⟨code⟩}` (for critical and familiar respectively) are for executing code before a group of notes, between the rules and the printing of the notes.

7.12.4 Spacing

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

7.12.5 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.

7.12.6 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.

7.12.7 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 7.10.2 p. 47.

7.13 Footnotes and the reledpar columns

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

`\noteswidthliketwocolumnsX`

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

7.14 Line number annotation

The way line number annotations are typeset can be changed using hooks as described in 9.10.2 p. 57 and 9.10.3 p. 57.

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

## 9 Verse

### 9.1 Basic

- `\stanza` Use `\stanza` at the start of a stanza. Each line in a stanza is ended by an ampersand (`&`),
- `\&`

and the stanza itself is ended by putting `\&` at the end of the last line.

If you need to add brackets directly after `\stanza`, `&` or `\&`, add `\norelax`. Otherwise, the brackets will be interpreted as delimitation of an optional argument (cf. 9.8 p. 54)

## 9.2 Define stanza indents

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

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

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

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

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

If you want the hanging verse to be flush right, you can use `\sethanginsymbol`: see p. 9.6 p. 54.

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:

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

is like

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

**Be careful: the feature is changed in eledmac 1.5.1. See Appendix A.3 p. 384.**

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{stanzaindentsrepetition}{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` 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 `\setstanzaindents` for this verse is skipped, and `{⟨value⟩}` is used instead.

`\stanzaindent*` 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 the 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. 73 for further details.

## 9.8 Content before/after verses

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

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

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

### `\AtEveryStanza`

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

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

### `\AtStartEveryStanza`

- Use `\AtStartEveryStanza` to automatically add content at the beginning of stanza (in the same paragraph).

- `&` 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.

A `\noindent` is automatically inserted before the contents of these optional arguments.

Use a third and fourth optional argument to not add these `\noindents` (to add content respectively after the current verse / 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.

- `\AtEveryStopStanza`
  - Use `\AtEveryStopStanza` to automatically add content after the end of stanzas (not in the same paragraph).  
Note that a `\noindent` will be inserted before the argument, and, consequently, a `\parskip`. You can use the starred version of `\AtEveryStopStanza` to avoid this `\noindent`.
- `\AtStartEveryStanza`
  - Use `\AtStartEveryStanza` to automatically add content at the end of stanza (in the same paragraph).

## 9.9 Numbering stanza

- `\numberstanzatrue` If you want to automatically number stanzas, use `\numberstanzatrue`. In this case, the line number will restart at each `\stanza`.
- `\numberstanzafalse` If you want to disable this feature again, use `\numberstanzafalse`.  
You can use this feature in combination with `\Xstanza` (7.2.8 p. 42).
- `\thestanza` You can redefine `\thestanza` to change the aspect of stanza number. Default value is:  

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

- `\stanzanumwrapper` You can change the value of the stanza counter with the usual commands of  $\text{\LaTeX}$ .  
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 Adding annotations to line numbers

You may want to have two or more numbers associated with a line of verse. Consider, for example, the following cases:

- You want to split a line of verse into two parts depending upon some stylistic / rhythmic / linguistic convention.
- You want to add the line number used by a previous edition of the work.

In such instances, you must add the second number manually, as `reledmac` can't determine a general pattern for such numbering, which depends too heavily upon the edited text.<sup>19</sup>

- `\linenumannotation` To resolve this issue, `reledmac` allows you to annotate line numbers using the fol-

<sup>19</sup>However, you may create your own commands dealing with such patterns which will in turn call `reledmac` commands.

lowing command:

```
\linenumannotation{<annotation>}
```

The annotation can contain any textual value (whether number, text, or other) such as the line number from the older edition. Here are some characteristics of line number annotation:

- An annotation is associated with a line of verse. It is reset with each new line of verse.
- It is printed alongside the line number in the margin.
- It is printed following the line number in critical footnotes and endnotes.
- And of particular interest: the annotation is printed in the critical note *only* when the `\edtext` is issued after the `\linenumannotation`.
- If two or more annotations are called before a `\edtext`, the last one is used in associated notes.
- If an annotation is called within an `\edtext`, it is printed as an annotation for the end of the lemma.

Here is an example, supposing we are on line 1:

```
\edtext{first lemma}
\linenumannotation{A}
\edtext{second lemma}{...}
\edtext{third%
 \linenumannotation{B}
 lemma}{...}
\edtext{fourth lemma}&
```

In the critical notes, the line number will be followed

- by nothing in the first lemma, as there is no annotation for this line;
- by “A” for the second lemma;
- by “A-B” for the third lemma, as it starts while annotation A is still in effect and it finishes after annotation B has already taken effect;
- by “B” for the fourth lemma.



### 9.10.1 Modifying annotation associated with a specific note

`\lineannot` The `\linenum` (6.2.5 p. 27) macro changes the line number passed to a note. The `\lineannot` macro does the same for the annotations. It takes one argument, containing the beginning and ending annotations separated by a pipe (`|`). As with `\linenum`, if one wishes to change some specific part of the annotation, one can use

```
\lineannot{b}% to change the ending annotation
\lineannot{a}% to change the beginning annotation
\lineannot{a|b}% to change both
\lineannot{empty}% to delete the beginning annotation
```

### 9.10.2 Changing the position of the annotation

By default, annotations are placed after line numbers in both margin and footnote.

To change this behaviour, one can use one of the following commands:

- `\Xlinenumannotationposition[⟨s⟩]{⟨v⟩}` changes the position in critical footnotes with `⟨s⟩` for the series of footnotes to which it applies, where `⟨s⟩` may be a comma-separated list of series. When `⟨s⟩` is empty, the change applies also to annotations at the sides of the page, alongside the line number.
  - `\Xlinenumannotationpositionside{⟨v⟩}` changes the position at the sides of the page, alongside the line numbers.
- Note that `\Xlinenumannotationposition` without any optional argument will override this setting.
- `\Xendlinenumannotationposition[⟨s⟩]{⟨v⟩}` changes the position in critical endnotes, `⟨s⟩` for the series of endnotes to which it applies, where `⟨s⟩` may once again be a comma-separated list of series.

In each of these three commands, `⟨v⟩` can be after or before to indicate position with regards to the line number.

### 9.10.3 Changing the macro in which annotations are wrapped

By default, annotations are wrapped in the `\textsuperscript` macro. You can modify this using one of the following:

- `\Xwraplinenumannotation[⟨s⟩]{⟨cmd⟩}` for critical footnotes; `⟨s⟩` indicates the series of notes to which it applies and may be a comma-separated list of series. When `⟨s⟩` is empty, the change applies to the annotations in the margins also—around the line number—as well as to the annotations referenced using the `\annotationref` command of the crossref mechanism (11 p. 59).
  - `\Xwraplinenumannotationside{⟨cmd⟩}` for annotations in the margins only.
- Note that `\Xwraplinenumannotation` without any optional argument will override this setting.

- `\Xwraplinenumannotationref{⟨cmd⟩}` for annotations referenced using the `\annotationref` command of the crossref mechanism (11 p. 59).

Note that `\Xwraplinenumannotation` without any optional argument will override this setting.

- `\Xendwraplinenumannotation[⟨s⟩]{⟨cmd⟩}` for critical endnotes, where `⟨s⟩` indicates the series of notes to which it applies, which may be a comma-separated list.

`⟨cmd⟩` is a command which can take one argument; the backslash is required though.

For example, if you do not wish to have annotations in the margins, but do want to have them in bold in the critical footnotes and endnotes, you say:<sup>20</sup>

```
\makeatletter
\Xwraplinenumannotation{\textbf}
\Xendwraplinenumannotation{\textbf}
\Xwraplinenumannotationside{\@gobble}
\makeatother
```

## 9.11 Various tools

- `\ampersand` If you need to print an `&` symbol in a stanza, use the `\ampersand` macro, not `\&` as this 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`.

## 9.12 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. 54).

# 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`

<sup>20</sup>The `\@gobble` command takes one argument, and returns nothing.

so it appears as normal text; it just might be that footnotes appear in the middle of a page, with text above and below.

`ledgroupsize`

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

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

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

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

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

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

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

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

## 11 Cross referencing

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

### 11.1 Basic use

`\edlabel`

First you place a label in the text using the command `\edlabel{\langle lab \rangle}`. `\langle lab \rangle` can be almost anything you like, including letters, numbers, punctuation, or a combination—anything but spaces; you might type `\edlabel{toves-3}`, for example.<sup>21</sup>

`\edpageref`

`\edlineref`

`\sublineref`

`\pstartref`

`\annotref`

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

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

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

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

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

<sup>21</sup>More precisely, you should stick to characters in the `TeX` categories of “letter” and “other”.

## 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. 62).

## 11.3 Cross-referencing which return a number in any case

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

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. 27).

For this situation, `reledmac` supplies variants of the reference commands, with the `x` prefix: `\xpageref`, `\xlineref`, `\xsublineref`, `\xpstartref` and `\xannotationref`. They have the following limitations:

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

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

`\xxref`

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

It automatically calls `\linenum` (6.2.5 p. 27) and `\lineannot` (?? p. ??) and sets the beginning page, line, subline numbers and line annotations to those of the place where `\edlabel{mouse}` was placed, and the ending ones to those where `\edlabel{elephant}` occurs.

For example, one might use the following:

```
\beginnumbering
```

```

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

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

\endnumbering

```

## 11.4 Not automatic cross-referencing

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

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

## 11.5 Normal $\LaTeX$ cross-referencing

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

## 11.6 References to start and end lines

### 11.6.1 Reference to main text lines

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

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

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

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

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

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

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

### 11.6.2 References to lines that are commented on in the apparatus

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

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

If you use `\linenum` (6.2.5 p. 27) to refer to these labels, `reledmac` will use your line settings to refer to the passage.

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

### 11.6.3 Settings

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

For example, you may use:

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

Note that if you do not use `\setapprefprefixmore`, the argument of `\setapprefprefixsingle` will be used in any case.

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

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

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

**Also note that `\setSerefonlypageprefixsingle` is only a shortcut for `\Xendbeforepagenumber` (see 11.6.3 p. 62). So if you use `\Xendbeforepagenumber` without any optional argument, it will override this setting.**

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

These commands are the following:

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

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

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`
- `\Xendmorethantwolines`
- `\Xendsublinesep`
- `\Xendtwolines`
- `\Xendtwolinesbutnotmore`
- `\Xendtwolinesonlyinsamepage`
- `\Xendlinenumannotationposition`
- `\Xendwraplinenumannotation`

**For one specific command** When calling `\appref` and `\Seref`, you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical footnotes (6.2.2 p. 25).

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

## 11.7 Compatibility with xr package

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

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

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

## 12 Sidenotes

### 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  $\TeX$  run. If the note positions change in your `.tex` file, you need two runs to get the correction position in the output file. You are strongly encouraged to use tools like *latexmk*, to be sure to get the correct number of runs.**

### 12.2 Setting

#### 12.2.1 Width

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



### 12.2.2 Vertical position

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

### 12.2.3 Distance to the main text

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

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

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

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

### 13.2 Use with `imakeidx` or `indextools`

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

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

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

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

### 13.3 Referring to critical notes

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

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

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

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

Where `#1` stands for the reference.

### 13.4 Separator between page and line numbers

`\pagelinesep`

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

- is the default separator used by the MAKEINDEX program.

You can reconfigure it, this example defines a colon as separator:

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

However, you also have to configure your `.ist` index style file. For example, if you use `:` as separator<sup>22</sup>.

```
page_compositor ":"
```

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

---

<sup>22</sup>For further detail, you can read <http://tex.stackexchange.com/a/32783/7712>.

## 13.5 Using xindy

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

`pagenumber-linenummer`

An example of such a file is provided in the “examples” folder. Read the `xindy` handbook to learn how to use it.<sup>23</sup>

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 choose 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) $\text{\TeX}$  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.<sup>24</sup>

## 13.6 Advanced setting

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

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

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

## 14 Glossary

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

<sup>23</sup>Or, for people who read French, read <http://geekographie.maieul.net/174>.

<sup>24</sup>These are the recommended lines to provide the best possible compatibility between `hyperref` and `xindy`, even without using `reledmac`.

## 14.1 Preamble setting

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

- Or set `.glossaries`:  
`\glsSetCompositor{-}`
- Or set `reledmac`:  
`\renewcommand{\pagelinesep}{-}`  
 In this case, the above will have consequences for your use of `\edindex` and you should set your `.ist` file (13.4 p. 66).

## 14.2 Commands

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

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

## 15 Tabular material

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

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

```

edarrayl \begin{edtabularc}
edarrayc 1 & 2 & 3 \\
edarrayr a & bb & ccc \\
edtabularl AAA & BB & C
edtabularc \end{edtabularc}
edtabularr

```

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

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

```

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

```

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

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

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

```

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

```

|   |      |    |    |                   |
|---|------|----|----|-------------------|
| 1 | 2    | 3  | 4  | 5                 |
| Q |      | fd | h  | qwertziohg        |
| v | wptz | x  | y  | vb                |
| g | nnn  |    |    |                   |
| k |      |    | pq | dgh               |
| 1 | 2    | 3  | co | ghweropjklmnbcxys |

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 | └─ |   | d |
| A | B  | C | D |

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

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

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

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

where  $\langle entry \rangle$  is an entry in the rightmost column, typesets  $\langle text \rangle$  right justified after the  $\langle entry \rangle$ .

For example:

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

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

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

**\edvertdots**

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

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

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

## 16 Sectioning commands

### 16.1 Sectioning commands without line numbers or critical notes

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

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

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

## 16.2 Sectioning commands with line numbering and critical notes

You have to use the following commands:

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

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

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

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

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

## 16.3 Optimization

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



## 17 Quotation environments

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

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

## 18 Page breaks

### 18.1 Control page breaking

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

`\ledpb`  
`\lednopb`

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

**These commands have effect only at the second run.**

`\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 p.  $n$ , and l. 444 at 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

`\lednopbinversettrue`

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

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

## 19 Miscellaneous

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

19.1 Known and suspected limitations

19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.12.6 p. 50 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. 72 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. 38, and described in more detail on XIII.6.3 p. 191, 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` (7.12.6 p. 50), 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 this²⁵ you will find \LaTeX whinging ‘Missing { inserted’, and then things start to fall apart. The trick in this case is to specify either:

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

or

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

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

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

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

```
\makeatletter
```

²⁵Reported by Dirk-Jan Dekker in the CTT thread ‘Incompatibility of “color” package’ on 2003/08/28.

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

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

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

19.2 Parallel typesetting

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

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

I Implementation overview

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

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

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

Next are commands for marking sections of the text for footnotes (Section VI), followed by the macros that take each paragraph apart, attach the line numbers and insertions, and send the result to the vertical list (Section VII). The footnote commands (Section XIII) and output routine (Section XXIII) finish the main part of the processing; cross-referencing (Section XXV) and endnotes (Section XX) 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}[2018/04/07 v2.26.7 typesetting critical editions]
4 %

```

II.3 Package options

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

```

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

```

II.3.1 Options of reledpar

Here, we define some booleans specific to `reledpar` options, but which have consequences on `reledmac` internal commands.

```

\newif\ifparledgroup
\newif\ifsameparallelpagenumber%
\newif\ifprevpgnotnumbered%%
\newif\ifwidthliketwocolumns%
\ifxindy@
\ifxindyhyperref@
\ifeledmaccompat@

```

II.3.2 Options of reledmac

```

11 \DeclareOptionX{series}[A,B,C,D,E]{\xdef\default@series{#1}}
12 \ExecuteOptionsX{series}%
13
14 \newif\if@noeled@sec%
15 \DeclareOptionX{noeledsec}{\@noeled@sectrue}
16
17 \newif\ifnocritical@%
18 \DeclareOptionX{nocritical}{\nocritical@true}%
19
20 \newif\ifnofamiliar@%
21 \DeclareOptionX{nofamiliar}{\nofamiliar@true}%
22
23 \newif\ifnoledgroup@%
24 \DeclareOptionX{noledgroup}{\noledgroup@true}%
25
26 \newif\ifnoend@%
27 \DeclareOptionX{noend}{%
28   \let\l@dend@open\@gobble%
29   \let\l@dend@close\relax%
30   \global\let\l@dend@stuff=\relax%
31   \noend@true%
32 }%

```

```

33 \newif\ifnoquotation@
34 \DeclareOptionX{noquotation}{\noquotation@true}
35
36 \newif\ifledfinal
37 \DeclareOptionX{final}{\ledfinaltrue}
38 \DeclareOptionX{draft}{\ledfinalfalse}
39 \ExecuteOptionsX{final}
40
41 \newif\ifparapparatus@
42 \DeclareOptionX{parapparatus}{\parapparatus@true}
43
44 \newif\iflednopbinverse
45 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
46
47 \newif\ifwidthliketwocolumns%
48 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
49
50 \newif\ifcontinuousnumberingwithcolumns
51 \DeclareOptionX{continuousnumberingwithcolumns}{\
52 continuousnumberingwithcolumnstrue}%
53
54 \newif\ifxindy@
55 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
56 \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
57 \newwrite\eledmac@xindy@out%
58 \xindy@true%
59 \gdef\eledmacmarkuplocdepth{:depth 1}%
60 \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
61 }%
62
63 \newif\ifxindyhyperref@
64 \DeclareOptionX{xindy+hyperref}{%
65 \xindyhyperref@true%
66 }%
67
68 \newif\ifeledmaccompat@%
69 \DeclareOptionX{eledmac-compat}{%
70 \eledmaccompat@true%
71 }%
72 \DeclareOptionX{nopenalties}{%
73 \AtBeginDocument{\let\add@penalties\relax}%
74 }
75 \def\l@auxdir{}%
76 \DeclareOptionX{auxdir}{%
77 \xdef\l@auxdir{#1/}%
78 }%
79
80 \newif\ifsw@caseinsensitive%
81 \DeclareOptionX{swcaseinsensitive}{%

```



```

82 \sw@caseinsensitivetrue%
83 }%
84 %

```

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.

```

85 \ProcessOptionsX*\relax
86
87 %

```

II.4 Loading packages

Loading package `xargs` to declare commands with optional arguments. Loading package `xparse` to declare fully expandable commands with optional argument. Ideally, we should use only `xparse` and not `xargs`. For historical reasons, we use both. `Etoolbox` is also used to make code clearer - for example, in dynamic command names (which can replace `\csname` etc.). Use suffix to declare commands with a starred version, `xstring` to work with strings, `ifluatex` and `ifxetex` to test if `LuaTeX` or `XYTeX` is running, and `ragged2e` to manage ragged justification for paragraphed notes.

```

88 \RequirePackage{xargs}
89 \RequirePackage{xparse}%
90 \RequirePackage{etoolbox}
91 \@ifl@t@r\fmtversion{2015/10/01}
92 {\ifboolexpr{not test{\@ifl@t@r\fmtversion{2016/03/31}}} or (test{\
  ifdefstring{\fmtversion}{2016/03/31}} and test {\ifnumless{\patch@level
  }{3}})}%
93 {\PackageWarning{reledmac}{You are using a LaTeX version older than
  2016/03/31 patch 3.%
94 \MessageBreak You are strongly encouraged to use a newer version.}}%
95 {}%
96 }%
97 {\RequirePackage{etex}%
98 \csname reserveinserts\endcsname{32}%
99 }%
100 \RequirePackage{suffix}
101 \RequirePackage{xstring}
102 \RequirePackage{ifluatex}
103 \RequirePackage{ragged2e}
104 \RequirePackage{ifxetex}%
105 %

```

II.5 Compatibility with LuaTeX

Here, we enable some primitives for `LuaTeX`.

```

106 \ifx\directlua\undefined\else%
107   \directlua{tex.enableprimitives("",{"texdir","pdir","bodydir"})}
108 \fi
109 %

```

II.6 Boolean flags

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

```

110 \newif\ifl@dmemoir
111 \ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
112
113 %

```

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

```

114 \newif\if@ledgroup%
115 %

```

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

```

116 \newif\ifl@imakeidx
117 \ifpackageloaded{imakeidx}{\l@imakeidxtrue}{\l@imakeidxfalse}%False is the default value
118 %

```

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

```

119 \newif\ifl@indextools%
120 \ifpackageloaded{indextools}{\l@indextoolstrue%
121   \l@imakeidxtrue%
122   \let\imki@wrindexentry\indtl@wrindexentry%
123 }{\l@imakeidxfalse}%
124 %
125 %

```

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

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

```

126 \newif\ifl@footmisc
127 \ifpackageloaded{footmisc}{\l@footmisctrue}{\l@footmiscfalse}%False is the default value
128 %

```

\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.

```

129 \ifdef{\if@RTL}{\newif\if@RTL}
130 %

```

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

```
131 \newif\if@firstlineofpage%
132 \newif\if@firstlineofpageR%
133 %
```

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.

```
134 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
135 %
```

`\reledmac@error` Write an error message.

```
136 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
137 %
```

```
\led@err@NumberingStarted 38 \newcommand*\led@err@NumberingStarted}{%
\led@err@NumberingNotStarted 39 \reledmac@error{Numbering has already been started}{\@ehc}}
NumberingShouldHaveStarted 40 \newcommand*\led@err@NumberingNotStarted}{%
141 \reledmac@error{Numbering was not started}{\@ehc}}
142 \newcommand*\led@err@NumberingShouldHaveStarted}{%
143 \reledmac@error{Numbering should already have been started}{\@ehc}}
144 %
```

```
\led@err@edtextoutsidestart 45 \newcommand*\led@err@edtextoutsidestart}{%
146 \reledmac@error{\string\edtext\space outside numbered paragraph (\pstart\
space ...\space\pend)}{\@ehc}}%
147 %
```

```
\led@mess@NotesChanged 58 \newcommand*\led@mess@NotesChanged}{%
149 \typeout{reledmac reminder: }%
150 \typeout{ The number of the footnotes in this section
151 has changed since the last run.}%
152 \typeout{ You will need to run LaTeX two more times
153 before the footnote placement}%
154 \typeout{ and line numbering in this section are
155 correct.}}
156 %
```

```

\led@mess@SectionContinued57 \newcommand*{\led@mess@SectionContinued}[1]{%
158 \message{Section #1 (continuing the previous section)}}
159 %

\led@err@LineationInNumbered60 \newcommand*{\led@err@LineationInNumbered}{%
161 \reledmac@error{You can't use \string\lineation\space within
162 a numbered section}{\@ehc}}
163 %

\led@warn@BadLineation64 \newcommand*{\led@warn@BadLineation}{%
\led@warn@BadLinenummargin65 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp66 \newcommand*{\led@warn@BadLinenummargin}{%
\led@warn@BadSublockdisp67 \reledmac@warning{Bad \string\linenummargin\space argument}}
168 \newcommand*{\led@warn@BadLockdisp}{%
169 \reledmac@warning{Bad \string\lockdisp\space argument}}
170 \newcommand*{\led@warn@BadSublockdisp}{%
171 \reledmac@warning{Bad \string\sublockdisp\space argument}}
172 %

\led@warn@NoFile73 \newcommand*{\led@warn@NoFile}[1]{%
174 \reledmac@warning{File `#1' not found}}
175 %

\led@warn@LineFileObsolete76 \newcommand*{\led@warn@Obsolete}[1]{%
177 \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
178 %

\led@warn@BadAdvancelineSubline79 \newcommand*{\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine80 \reledmac@warning{\string\advanceline\space produced a sub-line
181 number less than zero.}}
182 \newcommand*{\led@warn@BadAdvancelineLine}{%
183 \reledmac@warning{\string\advanceline\space produced a line
184 number less than zero.}}
185 %

\led@warn@BadSetline86 \newcommand*{\led@warn@BadSetline}{%
\led@warn@BadSetlinenum87 \reledmac@warning{Bad \string\setline\space argument}}
188 \newcommand*{\led@warn@BadSetlinenum}{%
189 \reledmac@warning{Bad \string\setlinenum\space argument}}
190 %

```

```

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

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

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

```

\led@warn@NoMarginpars31 \newcommand*\led@warn@NoMarginpars}{%
232 \reledmac@warning{You can't use \string\marginpar\space in numbered text
}}
233 %

```

```

\led@warn@BadSidenotemargin34 \newcommand*\led@warn@BadSidenotemargin}{%
235 \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
236 %

```

```

\led@warn@NoIndexFile37 \newcommand*\led@warn@NoIndexFile}[1]{%
238 \reledmac@warning{Undefined index file #1}}
239 %

```

```

\led@warn@SeriesStillExist40 \newcommand\led@warn@SeriesStillExist}[1]{%
241 \reledmac@warning{Series #1 is still existing !}%
242 }%
243 %

```

```

\led@err@BadAction44 \newcommand*\led@err@StanzaIndentNotDefined}{%
245 \reledmac@error{You have not defined the indentation for the line \number
\stanza@count}{\@ehc}}%
246 %

```

```

\led@err@ManySidenotes47 \newcommand\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes48 \ifledRcol{%
\led@err@ManyRightnotes49 \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
space p. \the\page@numR}%
250 \else%
251 \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\
space p. \the\page@num}%
252 \fi%
253 }%
254 \newcommand\led@err@ManyLeftnotes}{%
255 \ifledRcol{%
256 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\
space p. \the\page@numR}%
257 \else%
258 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\
space p. \the\page@num}%
259 \fi%
260 }%
261 \newcommand\led@err@ManyRightnotes}{%
262 \ifledRcol{%
263 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
space p. \the\page@numR}%

```

```

264 \else%
265 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\
space p. \the\page@num}%
266 \fi%
267 }%
268 %

```

```

\led@err@TooManyColumns 69 \newcommand*\led@err@TooManyColumns}{%
\led@err@UnequalColumns 70 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn 71 \newcommand*\led@err@UnequalColumns}{%
\led@err@HighEndColumn 72 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns 73 in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
274 \newcommand*\led@err@LowStartColumn}{%
275 \reledmac@error{Start column is too low}{\@ehc}}
276 \newcommand*\led@err@HighEndColumn}{%
277 \reledmac@error{End column is too high}{\@ehc}}
278 \newcommand*\led@err@ReverseColumns}{%
279 \reledmac@error{Start column is greater than end column}{\@ehc}}
280 %

```

```

\endnotes@outsidenumbering 81 \newcommand\led@err@toendnotes@outsidenumbering}{%
282 \reledmac@error{\string\toendnotes\space and related commands must be
called inside a numbered texte (\string\beginnumbering\space ... \space \string
\endnumbering)}{\@ehc}%
283 }%
284 %

```

```

\err@EdtextWithoutFootnote 85 \newcommand\led@err@EdtextWithoutFootnote}{%
286 \reledmac@error{edtext without Xfootnote. Check syntaxis}{\@ehc}%
287 }%
288 %

```

```

\footnoteNotInSecondArgEdtext 89 \newcommand\led@err@FootnoteNotInSecondArgEdtext}[1]{%
290 \reledmac@error{#1footnote outside of the second argument of an edtext.
Check syntax}{\@ehc}%
291 }%
292 %

```

```

\error@PackageAfterEledmac 93 \newcommand\led@error@PackageAfterEledmac}[1]{%
294 \reledmac@error{#1 must be loaded before reledmac}{\@ehc}%
295 }%
296 %

```

```

\led@error@fail@patch@@makecol297 \newcommand{\led@error@fail@patch@@makecol}{%
298 \reledmac@error{Fail to patch \string\@makecol\space command}{\@ehc}%
299 }%
300 %

```

```

\led@error@fail@patch@@reinserts301 \newcommand{\led@error@fail@patch@@reinserts}{%
302 \reledmac@error{Fail to patch \string\@reinserts\space command}{\@ehc}%
303 }%
304 %

```

```

\led@error@fail@patch@@doclearpage305 \newcommand{\led@error@fail@patch@@doclearpage}{%
306 \reledmac@error{Fail to patch \string\@doclearpage\space command}{\@ehc}%
307 }%
308 %

```

```

\led@error@fail@patch@@iiiminipage309 \newcommand{\led@error@fail@patch@@iiiminipage}{%
310 \reledmac@error{Fail to patch \string\@iiiminipage\space command}{\@ehc}%
311 }%
312 %

```

```

\led@error@fail@patch@endminipage313 \newcommand{\led@error@fail@patch@endminipage}{%
314 \reledmac@error{Failed to patch the \string\endminipage\space command}{\@ehc}%
315 }%
316 %

```

```

\led@error@fail@patch@endminipage317 \newcommand{\led@error@fail@patch@makeindex}{%
318 \reledmac@error{Failed to patch the \string\makeindex\space command}{\@ehc}%
319 }%
320 %

```

```

\led@warn@edinde@outsidenumbering321 \newcommand{\led@warn@edinde@outsidenumbering}{%
322 \reledmac@warning{\string\edindex\space called outside of \string\beginnumbering\space ... \space \string\endnumbering. \MessageBreak Automatically switched to \string\index.}%
323 }%
324 %

```

```

\led@warning@hsizeX@deprecated325 \newcommand{\led@warning@hsizeX@deprecated}{%
326 \reledmac@warning{\string\hsizeX\space command deprecated, use \string\widthX\space instead.}%
327 }%
328 %

```



```

warning@Xhsize@deprecated 29 \newcommand{\led@warning@Xhsize@deprecated}{%
330 \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
331 }%
332 %

```

```

warning@msdatawithoutstop 33 \newcommand{\led@warning@msdatawithoutstop}{%
334 \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
335 }%
336 %

```

```

ning@preXnotes@deprecated 37 \newcommand{\led@warning@preXnotes@deprecated}{%
338 \reledmac@warning@preXnotes@deprecated%
339 }%
340 %

```

II.8 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree 41 \providecommand*\@gobblethree}[3]{}
\@gobblefour 42 \providecommand*\@gobblefour}[4]{}
\@gobbleseven 43 \providecommand*\@gobbleseven}[7]{}
344 %

```

II.9 Wrapping content

`\l@wrap@ifnotemptybox` The `\l@wrap@ifnotemptybox` macro wraps its second argument in the command passed as the first argument, but only if the second argument does not produce an empty box. That is useful for example when we need to wrap something on a `\textsuperscript`.

Indeed `ab` does not produce the same typographical result as `ab`.²⁶ The `\l@wrapcs@ifnotemptybox` does the same, but take as argument a control sequence name, so without backslash.

```

345 \newcommand{\l@wrap@ifnotemptybox}[2]{%
346 \setbox0=\hbox{#2}%
347 \ifdim\wd0=\z@ \else%
348 #1{#2}%
349 \fi%
350 }%
351 \newcommand{\l@wrapcs@ifnotemptybox}[2]{%
352 \l@wrap@ifnotemptybox{\csname #1\endcsname}{#2}%
353 }%
354 %

```

²⁶Compare the distance between letters in the first case (`ab`) and in the second one (`ab`).

II.10 Miscellaneous commands

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

```

355 \ifledfinal
356   \newcommand*{\showlemma}[1]{#1}
357 \else
358   \newcommand*{\showlemma}[1]{\underline{#1}}
359 \fi
360
361 %

```

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

```

362 \let\linenumberlist=\empty
363
364 %

```

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

```

365 \newcount\@l@tempcnta \newcount\@l@tempcntb
366 %

```

II.11 Prepare reledpar

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

```

367 \newif\ifl@dpairing
368 \newif\ifl@dpaging%
369 \newif\ifl@dprintingpages%
370 \newif\ifl@dprintingcolumns%
371 \newif\ifpst@rtedL
372 \newcount\l@dnumpstartsL
373 %

```

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

```

374 \newif\ifledRcol
375 \newif\ifledRcol@
376 %

```

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

```
377 \newif\ifnumberingR
378 %
```

`\ifafterendnumberingR` The `\ifafterendnumberingR` boolean is set to TRUE at `\endnumberingR` and to FALSE at next `\beginnumberingR` or real `\beginnumbering` (not `\resumenummering`). It is mainly used for the `continuousnumberingwithcolumns`, to avoid incrementing right column line counters after a `\endnumberingR`.

```
379 \newif\ifafterendnumberingR%
380 %
```

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

```
381 \newif\ifXnote@%
382 %
```

II.12 Booleans provided by other optional packages which are required in any case

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

```
383 \providebool{indtl@innote}%
384 \providebool{indtl@notenumber}%
385 %
```

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 `\resumenummering` 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.

```

386 \newcount\section@num
387 \section@num=0
388 \let\extensionchars=\empty
389 %

```

\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.

```

390 \newif\ifnumbering
391 %

```

\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.

```

392 \newcommand*{\beginnumbering}{%
393   \ifnumbering
394     \led@err@NumberingStarted
395   \endnumbering
396 \fi
397 \global\numberingtrue
398 \global\afterendnumberingRfalse%
399 \global\advance\section@num \@ne
400 \initnumbering@reg
401 \message{Section \the\section@num }%
402 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
403 \ifcontinuousnumberingwithcolumns%
404   \unless\ifafterendnumberingR%
405     \unless\ifl@dpairing%
406       \ledRcoltrue%
407       \global\advance\section@numR \@ne%
408       \message{Section \the\section@numR R (continuousnumbering)}%
409       \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
410       \ledRcolfalse%
411     \fi%

```

```

412 \fi%
413 \fi%
414 \l@dend@stuff
415 \setcounter{pstart}{1}
416 \ifl@dpairing
417 \global\l@dnumpstartsL \z@
418 \global\pst@rtedLfalse
419 %

```

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.

```

420 \else
421 \begingroup
422 \global\@afterindenttrue%In order to reestablish normal feature if the \
begingroup was not here
423 \initnumbering@quote
424 \ifwidthliketwocolumns%
425 \setwidthliketwocolumns%
426 \csuse{setpositionliketwocolumns@\columns@position}%
427 \fi%
428 \fi
429 \gdef\eled@sections@{}%
430 \if@noeled@sec\else%
431 \makeatletter%
432 \InputIfFileExists%
433 {\l@auxdir\jobname.eledsec\the\section@num}%
434 {}%
435 {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@num}}%
436 \makeatother%
437 \immediate\openout\eled@sectioning@out=\l@auxdir\jobname.eledsec\the\
section@num\relax%
438 \fi%
439 }
440 \newcommand*{\initnumbering@reg}{%
441 \global\pst@rtedLfalse
442 \global\l@dnumpstartsL \z@
443 \global\absline@num \z@
444 \gdef\normal@page@break{}
445 \gdef\l@prev@pb{}
446 \gdef\l@prev@nopb{}
447 \global\line@num \z@
448 \global\subline@num \z@
449 \global\@lock \z@
450 \global\sub@lock \z@
451 \global\sublines@false

```

```

452 \global\let\next@page@num=\relax
453 \global\let\this@section@next@page@num=\relax%
454 \global\let\sub@change=\relax
455 \global\last@page@num=-10000%
456 \ifdefined\line@numR%
457   \line@numR=\z@%
458   \last@page@numR=\z@%
459 \fi%
460 \resetprevline@
461 \resetprevpage@num
462 \global\stopmsdata@inserted@true%
463 \global\let\@msdata@list\relax%
464 \global\csundef{\@msdata@\add@msd@c @data}%
465 }
466
467 %

```

`\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. We define it using `\def` and not `\newcommand` because \TeX does not allow defining command whose name starts by “end” except if we are currently creating an environment, which is not the case here.

```

468 \def\endnumbering{%
469   \ifnumbering
470     \global\numberingfalse
471     \normal@pars
472     \ifnum\l@dnum@startsL=0%
473       \led@err@NumberingWithoutPstart%
474     \fi%
475 %

```

For the `continuousnumberingwithcolumn` options, we need to store some value globally, for the next `\endnumbering`. We didn’t modify this counter globally, because that creates problem for nested `\edtext`.

```

476 \global\page@num=\this@section@page@num%
477 \global\last@page@num=\this@section@last@page@num%
478 \global\let\next@page@num\this@section@next@page@num%
479 \ifl@dpairing
480   \global\pst@rtedLfalse
481 \else
482   \ifx\insertlines@list\empty\else
483     \global\noteschanged@true
484   \fi
485   \ifx\line@list\empty\else
486     \global\noteschanged@true
487   \fi
488 \fi
489 \ifnoteschanged@

```

```

490     \led@mess@NotesChanged
491   \fi
492 \else
493   \led@err@NumberingNotStarted
494 \fi
495 \autoparfalse
496 \if@noeled@sec\else%
497   \immediate\closeout\eled@sectioning@out%
498 \fi%
499 \ifl@dpairing\else
500   \global\l@dnumstartsL=\z@%
501 \endgroup
502 \fi
503 }
504 %

```

`\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.²⁷ The `\ifresumenumbering@start` `\pausenumbering@page@num` counter stores the `\this@section@next@page@num` when the `\pause@numbering` is called.

```

505 \newcount\pausenumbering@page@num%
506 \newcommand{\pausenumbering}{%
507   \ifx\this@section@next@page@num\relax%
508     \global\pausenumbering@page@num=0%
509   \else%
510     \global\pausenumbering@page@num=\this@section@next@page@num%
511   \fi%
512   \ifautopar\global\autopar@pausettrue\fi%
513   \endnumbering\global\numberingtrue}
514 %

```

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. The boolean `\ifresumenumbering@start` is set to true. That allows us to avoid resetting the line number at the first line of `\resumenumbering` if the lineation is by page. This boolean is set to false after the first action.

```

515 \newif\ifresumenumbering@start%
516 \newcommand*{\resumenumbering}{%
517   \ifnumbering
518     \ifautopar@pause\autopar\fi
519     \global\pst@rtedLtrue
520     \global\advance\section@num \@ne
521     \global\resumenumbering@starttrue%
522     \led@mess@SectionContinued{\the\section@num}%
523     \set@continuousnumberingforL%

```

²⁷Peter Wilson's thanks to Wayne Sullivan, who suggested the idea behind these macros.

```

524 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
525 \ifcontinuousnumberingwithcolumns%
526   \unless\ifafterendnumberingR%
527     \unless\ifl@dpairing%
528       \ledRcoltrue%
529       \global\advance\section@numR \@ne%
530       \message{Section \the\section@numR R (continuousnumbering)}%
531       \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
532       \ledRcolfalse%
533     \fi%
534   \fi%
535 \fi%
536 \l@dend@stuff
537 \ifl@dpairing\else%
538   \begingroup%
539   \initnumbering@quote%
540   \ifwidthliketwocolumns%
541     \setwidthliketwocolumns%
542     \csuse{setpositionliketwocolumns@\columns@position}%
543   \fi%
544 \fi%
545 \else
546   \led@err@NumberingShouldHaveStarted
547   \endnumbering
548   \beginnumbering
549 \fi}
550
551
552 %

```

`\set@continuousnumberingforL` `\set@continuousnumberingforl` sets pstart counters at a `\beginnumbering` or a `\resumenumbers` in order to have continuous numbering with single column text.

```

553 \newcommand{\set@continuousnumberingforL}{%
554   \ifcontinuousnumberingwithcolumns%
555     \ifl@dpairing%
556       \unless\ifl@dpageing%
557         \global\c@pstartL=\c@pstart%
558       \fi%
559     \fi%
560   \fi%
561 }%
562 %

```

IV List macros

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

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

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

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

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

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

```
563 \newcommand*\list@create}[1]{%
564   \global\let#1=\empty%
565 }%
566 %
```

`\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 .

```
567 \newcommand*\list@clear}[1]{%
568   \global\let#1=\empty%
569 }
570 %
```

`\xright@appenditem` `\xright@appenditem` expands an item and appends it to the right end of a list macro.
`\led@toksa` We want the expansion because we will often be using this to store the current value
`\led@toksb` of a counter. `\xright@appenditem` creates global control sequences, like `\xdef`, and uses two temporary token-list registers, `\@toksa` and `\@toksb`.

```
571 \newtoks\led@toksa \newtoks\led@toksb
572 \global\led@toksa={\}
573 \long\def\xright@appenditem#1\to#2{%
574   \global\led@toksb=\expandafter{#2}%
575   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
576   \global\led@toksb={}}
577 %
```

`\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`.

```
578 \long\def\xleft@appenditem#1\to#2{%
579   \global\led@toksb=\expandafter{#2}%
580   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
581   \global\led@toksb={}}
582 %
```

`\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.

```
583 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
584 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
585
586 %
```

V Line counting

V.1 Choosing the system of lineation

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

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

`\bypstart@true`

`\bypstart@false`

`\ifbypage@`

`\bypage@true`

`\bypage@false`

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

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

```
587 \newif\ifbypage@
588 \newif\ifbypstart@
589 %
```

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@R90 \newif\ifbypage@R
\ifbypstart@R91 \newif\ifbypstart@R
592 %
```

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

```
593 \newcommand*{\lineation}[1]{\{
594 %
```

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

```
595 \ifnumbering
596 \led@err@LineationInNumbered
597 \else
598 %
```

If the argument is page.

```

599 \def\@tempa{#1}\def\@tempb{page}%
600 \ifx\@tempa\@tempb
601 \global\bypage@true
602 \global\bystart@false
603 \unless\ifnocritical@%
604 \Xpstart[] [false]%
605 \fi%
606 %

```

If the argument is pstart.

```

607 \else
608 \def\@tempb{pstart}%
609 \ifx\@tempa\@tempb
610 \global\bypage@false
611 \global\bystart@true
612 \unless\ifnocritical@%
613 \Xpstart%
614 \fi%
615 %

```

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

```

616 \else
617 \def\@tempb{section}
618 \ifx\@tempa\@tempb
619 \global\bypage@false
620 \global\bystart@false
621 \unless\ifnocritical@%
622 \Xpstart[] [false]%
623 \fi%
624 %

```

In other case, it is an error.

```

625 \else
626 \led@warn@BadLineation
627 \fi
628 \fi
629 \fi
630 \fi}}
631 %

```

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`

```

632 \newcount\line@margin%
633 \newcount\line@margin@columns%Only for parallel typesetting
634 \line@margin@columns=\m@ne%
635
636 \newcommand*\linenummargin}[1]{%
637   \l@getline@margin{#1}%
638   \ifnum\@l@tempcntb>\m@ne
639     \ifledRcol
640       \global\line@marginR=\@l@tempcntb
641       \led@warn@setting@in@rightside{\linenummargin}%
642     \else
643       \global\line@margin=\@l@tempcntb
644     \fi
645   \fi}}
646
647 \newcommand*\l@getline@margin#[1]{%
648   \def\@tempa{#1}\def\@tempb{left}%
649   \ifx\@tempa\@tempb
650     \@l@tempcntb \z@
651   \else
652     \def\@tempb{right}%
653     \ifx\@tempa\@tempb
654       \@l@tempcntb \@ne
655     \else
656       \def\@tempb{outer}%
657       \ifx\@tempa\@tempb
658         \@l@tempcntb \tw@
659       \else
660         \def\@tempb{inner}%
661         \ifx\@tempa\@tempb
662           \@l@tempcntb \thr@@
663         \else
664           \led@warn@BadLinenummargin
665           \@l@tempcntb \m@ne
666         \fi
667       \fi
668     \fi
669   \fi}
670
671 %

```

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.

```

672 \newcounter{firstlinenum}
673   \setcounter{firstlinenum}{5}
674 \newcounter{linenumincrement}
675   \setcounter{linenumincrement}{5}
676 %

```

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

```

677 \newcounter{firstsublinenum}
678   \setcounter{firstsublinenum}{5}
679 \newcounter{sublinenumincrement}
680   \setcounter{sublinenumincrement}{5}
681
682 %

```

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

```

\linenumincrement
\firstsublinenum
\sublinenumincrement
683
684 \newcommand*{\firstlinenum}[1]{%
685   \ifledRcol%
686     \setcounter{firstlinenumR}{#1}%
687     \led@warn@setting@in@rightside{\firstlinenum}%
688   \else%
689     \setcounter{firstlinenum}{#1}%
690   \fi%
691 }
692 \newcommand*{\linenumincrement}[1]{%
693   \ifledRcol%
694     \setcounter{linenumincrementR}{#1}%
695     \led@warn@setting@in@rightside{\linenumincrement}%
696   \else%
697     \setcounter{linenumincrement}{#1}%
698   \fi%
699 }
700 \newcommand*{\firstsublinenum}[1]{%
701   \ifledRcol%
702     \setcounter{firstsublinenumR}{#1}%
703     \led@warn@setting@in@rightside{\firstsublinenum}%
704   \else%
705     \setcounter{firstsublinenum}{#1}%
706   \fi%
707 }
708 \newcommand*{\sublinenumincrement}[1]{%
709   \ifledRcol%
710     \setcounter{sublinenumincrementR}{#1}%
711     \led@warn@setting@in@rightside{\sublinenumincrement}%
712   \else%
713     \setcounter{sublinenumincrement}{#1}%
714   \fi%

```

```

715 }
716
717 %

```

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.

```

718 \newcount\lock@disp
719 \newcommand{\lockdisp}[1]{%
720   \l@getlock@disp{#1}%
721   \ifnum\l@dttempcntb>\m@ne
722     \global\lock@disp=\l@dttempcntb
723   \else
724     \led@warn@BadLockdisp
725   \fi}}
726 \newcommand*{\l@getlock@disp}[1]{
727   \def\@tempa{#1}\def\@tempb{first}%
728   \ifx\@tempa\@tempb
729     \l@dttempcntb \z@
730   \else
731     \def\@tempb{last}%
732     \ifx\@tempa\@tempb
733       \l@dttempcntb \@ne
734     \else
735       \def\@tempb{all}%
736       \ifx\@tempa\@tempb
737         \l@dttempcntb \tw@
738       \else
739         \l@dttempcntb \m@ne
740       \fi
741     \fi
742   \fi}
743
744 %

```

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

```

745 \newcount\sublock@disp
746 \newcommand{\sublockdisp}[1]{%
747   \l@getlock@disp{#1}%
748   \ifnum\l@dttempcntb>\m@ne
749     \global\sublock@disp=\l@dttempcntb
750   \else

```

```

751 \led@warn@BadSublockdisp
752 \fi}}
753
754 %

```

V.5 Line number style

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

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

`\linenumr@p` and `\sublinenumr@p`.

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

`\sublinenumrep`

`\sublinenumr@p`

```

755 \newcommand*{\linenumberstyle}[1]{%
756   \def\linenumrep##1{\@nameuse{@#1}{##1}}
757 \newcommand*{\sublinenumberstyle}[1]{%
758   \def\sublinenumrep##1{\@nameuse{@#1}{##1}}
759 %

```

Initialise the number styles to arabic.

```

760 \linenumberstyle{arabic}
761 \let\linenumr@p\linenumrep
762 \sublinenumberstyle{arabic}
763 \let\sublinenumr@p\sublinenumrep
764
765 %

```

V.6 Line number printing

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

`\rightlinenum`

`\linenumsep`

`\numlabfont`

`\ledlinenum`

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

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

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

```

766 \newlength{\linenumsep}
767 \setlength{\linenumsep}{1pc}
768 \newcommand*{\numlabfont}{\normalfont\scriptsize}
769 \newcommand*{\ledlinenum}{%
770   \bgroup%
771   \ifluatex%
772     \texdir TLT%
773   \fi%
774   \numlabfont%
775   \ifdefstring{\Xlinenumannotationposition@side}{before}{%
776     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
777       \csuse{annot@the\absline@num @the\section@num}%
778     }%
779   }{}%
780   \linenumrep{\line@num}%
781   \ifsublines@
782     \ifnum\subline@num>0\relax
783       \unskip%
784       \Xsublinesep@side%
785       \sublinenumrep{\subline@num}%
786     \fi
787   \fi%
788   \ifdefstring{\Xlinenumannotationposition@side}{after}{%
789     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
790       \csuse{annot@the\absline@num @the\section@num}%
791     }%
792   }{}%
793   \egroup%
794 }%
795
796 \newcommand*{\leftlinenum}{%
797   \ledlinenum
798   \kern\linenumsep}
799 \newcommand*{\rightlinenum}{%
800   \kern\linenumsep
801   \ledlinenum}
802
803 %

```

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`.

```
804 \newcount\line@num
805 %
```

`\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.

```
806 \newcount\subline@num
807 %
```

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

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

`\sublines@false`

```
808 \newif\ifsublines@
809 %
```

`\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.

```
810 \newcount\absline@num
811 %
```

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.

```
812 \newcount\@lock
813 \newcount\sub@lock
814 %
```

V.9 Line number associated to lemma

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

`\insertlines@list`
`\actionlines@list`
`\actions@list`
`\annot@list`

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

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

```
23|35|0|24|3|0|OT1/cmr/m/n.
```

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

- `\insertlines@list`: the line numbers of lines that have footnotes or other insertions. These are the absolute numbers where the corresponding lemmas begin. This list contains one entry for every footnote in the section; one lemma may contribute no footnotes or many footnotes. This list is used by `\add@inserts` within `\do@line`, to tell it where to insert notes.
- `\actionlines@list`: a list of absolute line numbers at which we are to perform special actions; these actions are specified by the `\actions@list` list defined below.
- `\actions@list`: action codes corresponding to the line numbers in `\actionlines@list`. These codes tell `reledmac` what action it is supposed to take at each of these lines. One action, the page-start action, is generated behind the scenes by `reledmac`

itself; the others, for specifying sub-lineation, line-number locking, and line-number alteration, are generated only by explicit commands in your input file. The page-start and line-number-alteration actions require arguments, to specify the new values for the page or line numbers; instead of storing those arguments in another list, we have chosen the action-code values so that they can encode both the action and the argument in these cases. Action codes greater than -1000 are page-start actions, and the code value is the page number; action codes less than -5000 specify line numbers, and the code value is a transformed version of the line number; action codes between these two values specify other actions which require no argument.

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

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

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

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

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

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

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

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

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

An action code of -5000 or less sets the current visible line number (either the line number or the sub-line number, whichever is currently being advanced) to a specific positive value. The value of the code is $-(5000 + n)$, where n is the value (always ≥ 0) assigned to the current line number. Action codes of this type are added to the list by the `\set@line@action` macro, as called to implement

the `\advanceline` and `\setline` macros: this action only occurs when the user has specified some change to the line numbers using those macros. Normally `reledmac` computes the visible line numbers from the absolute line numbers with reference to the other action codes and the settings they invoke; it does not require an entry in the action-code list for every line.

- `\annot@list` stores line number annotations for the start and the end lines of each `\edtext`, as `\line@list` does for line numbers. We don't store that in the same list as line numbers as `\printlines` already have height arguments, and a `TEX` macro can take at most nine arguments.

Here are the commands to create these lists:

```
815 \list@create{\line@list}
816 \list@create{\insertlines@list}
817 \list@create{\actionlines@list}
818 \list@create{\actions@list}
819 \list@create{\annot@list}%
820 %
```

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

```
\endpage@num
\endline@num
\endsubline@num
821 \newcount\page@num
822 \newcount\endpage@num
823 \newcount\endline@num
824 \newcount\endsubline@num
825 %
```

`\this@section@page@num` The `\this@section@page@num` stores the page number of the on which a numbering section finish.

```
826 \newcount\this@section@page@num%
827 %
```

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

```
828 \newif\ifnoteschanged@
829 %
```

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

```
\resetprevline@ \newcommand*{\resetprevline@}{%
831   \def\do##1{\global\csundef{prevline##1}}%
832   \dolistloop{\@series}%
833 }
834 %
```

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

```
\resetprevpage@ \newcommand*{\resetprevpage@num}{%
835   \def\do##1{%
836     \ifcsdef{prevpage##1@num}{%
837       \global\csname prevpage##1@num\endcsname=\z@%
838       \global\csname prevpage##1@numR\endcsname=\z@%
839     }%
840     {}%
841   \ifcsdef{##1prevpage@num}{%
842     \global\csname ##1prevpage@num\endcsname=\z@%
843     \global\csname ##1prevpage@numR\endcsname=\z@%
844   }%
845   {}%
846   }%
847   \dolistloop{\@series}%
848 }
849 %
850 %
```

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.

```
851 \newread\@inputcheck
852 \newcommand*{\read@linelist}[1]{%
853   \ifledRcol%
854   \list@clearing@regR%
855   \else%
856   \list@clearing@reg%
857   \fi%
858 }
```

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).

```
859 \list@clear{\maxlinesinpar@list}
860 %
```

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

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

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

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

```
861 \get@linelistfile{#1}%
862 \@stopmsd%Security if last \endms{} is forgotten
863 \unless\ifledRcol%Get the last line of the last page
864 \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
865 \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
866 \else%
867 \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
868 \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
869 \fi%
870 \endgroup
871 %
```

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.

```
872 \ifledRcol
873 \global\page@numR=\m@ne
874 \ifx\actionlines@listR\empty
875 \gdef\next@actionlineR{1000000}%
876 \else
877 \glp\actionlines@listR\to\next@actionlineR
878 \glp\actions@listR\to\next@actionR
879 \fi
880 \else
881 \global\page@num=\m@ne
882 \ifx\actionlines@list\empty
```

```

883 \gdef\next@actionline{1000000}%
884 \else
885 \glp\actionlines@list\to\next@actionline
886 \glp\actions@list\to\next@action
887 \fi
888 \fi
889 }
890 %

```

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

```

891 \newcommand*\list@clearing@reg{%
892 \list@clear{\line@list}%
893 \list@clear{\insertlines@list}%
894 \list@clear{\actionlines@list}%
895 \list@clear{\actions@list}%
896 \list@clear{\linesinpar@listL}%
897 \list@clear{\linesonpage@listL}%
898 }%
899 %

```

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

```

900 \newcommand*\get@linelistfile[1]{%
901 \InputIfFileExists{\l@auxdir#1}{%
902 \global\noteschanged@false
903 \begingroup
904 \catcode`\[=1 \catcode`\]=2
905 \makeatletter \catcode`\^M=9}{%
906 \led@warn@NoFile{\l@auxdir#1}%
907 \global\noteschanged@true
908 \begingroup}%
909 }
910 %
911 %

```

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. 20 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.

```

912 \newcommand{\line@list@version}[1]{%
913   \IfStrEq{#1}{\this@line@list@version}%
914   {}%
915   {\ifledRcol%
916     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
917     \else%
918     \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
919     \fi%
920     \endinput%
921   }%
922 }%
923 %

```

`\@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 `\@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`.


```

925 \newcommand*{\@nl}[2]{%
926   \@page{#1}%
927   \ifledRcol%
928     \@nl@regR%
929   \else%
930     \@nl@reg%
931   \fi%
932 }
933 \newcommand*{\@nl@reg}{%
934   \ifx\l@dchset@num\relax \else
935     \reset@current@annot%
936     \advance\absline@num \@ne
937     \csgdef{\l@dchset@num@the\absline@num}{}%To remember this line have
been marked by a \setlinenum
938     \set@line@action
939     \let\l@dchset@num=\relax
940     \advance\absline@num \m@ne
941     \advance\line@num \m@ne
942   \fi
943   %

```

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

```

944   \advance\absline@num \@ne
945     \ifx\next@page@num\relax \else
946       \page@action
947       \let\next@page@num=\relax
948     \fi
949   \ifx\sub@change\relax \else
950     \ifnum\sub@change>\z@
951       \sublines@true
952     \else
953       \sublines@false
954     \fi
955     \sub@action
956     \let\sub@change=\relax
957   \fi
958   %

```

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

```

959   \ifcase\@lock
960     \or
961       \@lock \tw@
962     \or \or
963       \@lock \z@
964     \fi
965   \ifcase\sub@lock
966     \or

```

```

967         \sub@lock \tw@
968         \or \or
969         \sub@lock \z@
970     \fi
971 %

```

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

```

972     \ifsublines@
973         \ifnum\sub@lock<\tw@
974             \advance\subline@num \@ne
975         \fi
976     \else
977         \ifnum\@lock<\tw@
978             \advance\line@num \@ne \subline@num \z@
979         \fi
980     \fi}
981 %
982 %

```

`\last@page@num` `\@page` determines whether or not a new page has been started, based on the page values held by `\@n1`.

```

\@page
983 \newcount\last@page@num
984 \last@page@num=-10000
985
986 \newcount\this@section@last@page@num%
987 \this@section@last@page@num=-10000%
988
989 \newcommand*{\@page}[1]{%
990     \ifledRcol
991         \ifnum #1=\last@page@numR
992     \else
993         \csnumgdef\@lastabsline@forpageR@{\the\page@numR}{\the\absline@numR}%
994         \csnumgdef\@lastline@forpageR@{\the\page@numR}{\the\line@numR}%
995         \ifbypage@R
996             \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
997                 \ifboolexpr{%
998                     bool{resumenumberingR@start}%
999                     and test {\ifnumequal{\last@page@numR}{-10000}}%
1000                 }%
1001                 {}%
1002                 {%
1003                     \line@numR \z@%
1004                     \subline@numR \z@%
1005                     \global\csdef{reset@lineR\the\numexpr\absline@numR+1\relax @\
the\section@numR}{}%
1006                 }%
1007             \fi%
1008         \fi
1009         \page@numR=#1%

```

```

1010 \this@section@page@numR=#1%
1011 \last@page@numR=#1%
1012 \this@section@last@page@numR=#1%
1013 \def\next@page@numR{#1}%
1014 \gdef\this@section@next@page@numR{#1}%
1015 \fi
1016 \else
1017 \ifnum #1=\last@page@num
1018 \else
1019 \csnumgdef\@lastabsline@forpage@{\the\page@num}{\the\absline@num}%
1020 \csnumgdef\@lastline@forpage@{\the\page@num}{\the\line@num}%
1021 \ifbypage@
1022 \ifx\l@dchset@num\relax%Not resetting if preceded by a \setlinenum
1023 \line@num \z@%
1024 \subline@num \z@%
1025 \global\csdef{reset@line\the\numexpr\absline@num+1\relax @\the\
section@num}{}%
1026 \fi%
1027 \fi
1028 \page@num=#1%
1029 \global\this@section@page@num=#1%
1030 \last@page@num=#1%
1031 \global\this@section@last@page@num=#1%
1032 \def\next@page@num{#1}%
1033 %

```

And we set a flag that tells \@n1 that a new page number is to be set, because other associated actions shouldn't occur until the next line-start occurs.

```

1034 \gdef\this@section@next@page@num{#1}%
1035 \listxadd{\normal@page@break}{\the\absline@num}
1036 \fi
1037 \fi}
1038 %

```

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

```

1039 \newcommand*\@pend}[1]{}
1040 \newcommand*\@pendR}[1]{}
1041 \newcommand*\@lopL}[1]{}
1042 \newcommand*\@lopR}[1]{}
1043
1044 %

```

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

```

1045 \newcommand*\sub@on{\ifsublines@
1046   \let\sub@change=\relax
1047   \else
1048     \def\sub@change{1}%
1049   \fi}
1050 \newcommand*\sub@off{\ifsublines@
1051   \def\sub@change{-1}%
1052   \else
1053     \let\sub@change=\relax
1054   \fi}
1055
1056 %

```

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

```

1057
1058 \newcommand*\@adv}[1]{%
1059   \ifsublines@
1060     \ifledRcol
1061       \advance\subline@numR by #1\relax
1062       \ifnum\subline@numR<\z@
1063         \led@warn@BadAdvancelineSubline
1064         \subline@numR \z@
1065       \fi
1066     \else
1067       \advance\subline@num by #1\relax
1068       \ifnum\subline@num<\z@
1069         \led@warn@BadAdvancelineSubline
1070         \subline@num \z@
1071       \fi
1072     \fi
1073   \else
1074     \ifledRcol
1075       \advance\line@numR by #1\relax
1076       \ifnum\line@numR<\z@
1077         \led@warn@BadAdvancelineLine
1078         \line@numR \z@
1079       \fi
1080     \else
1081       \advance\line@num by #1\relax
1082       \ifnum\line@num<\z@
1083         \led@warn@BadAdvancelineLine
1084         \line@num \z@
1085       \fi
1086     \fi
1087   \fi
1088   \set@line@action}
1089
1090 %

```

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

```

1091
1092 \newcommand*\@set}[1]{%
1093   \ifledRcol
1094     \ifsublines@
1095       \subline@numR=#1\relax
1096     \else
1097       \line@numR=#1\relax
1098     \fi
1099     \set@line@action
1100   \else
1101     \ifsublines@
1102       \subline@num=#1\relax
1103     \else
1104       \line@num=#1\relax
1105     \fi
1106     \set@line@action
1107   \fi}
1108
1109 %

```

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

```

1110
1111 \newcommand*\l@d@set}[1]{%
1112   \ifledRcol
1113     \line@numR=#1\relax
1114     \advance\line@numR \@ne
1115     \def\l@dchset@num{#1}
1116   \else
1117     \line@num=#1\relax
1118     \advance\line@num \@ne
1119     \def\l@dchset@num{#1}
1120   \fi}
1121 \let\l@dchset@num\relax
1122
1123 %

```

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

```

1124
1125 \newcommand*\page@action{%
1126   \ifledRcol
1127     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1128     \xright@appenditem{\next@page@numR}\to\actions@listR

```

```

1129 \else
1130 \xright@appenditem{\the\absline@num}\to\actionlines@list
1131 \xright@appenditem{\next@page@num}\to\actions@list
1132 \fi}
1133 %

```

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

```

1134
1135 \newcommand*\set@line@action{%
1136 \ifledRcol
1137 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1138 \ifsublines@
1139 \@l@tempcnta=-\subline@numR
1140 \else
1141 \@l@tempcnta=-\line@numR
1142 \fi
1143 \advance\@l@tempcnta by -5000\relax
1144 \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1145 \else
1146 \xright@appenditem{\the\absline@num}\to\actionlines@list
1147 \ifsublines@
1148 \@l@tempcnta=-\subline@num
1149 \else
1150 \@l@tempcnta=-\line@num
1151 \fi
1152 \advance\@l@tempcnta by -5000\relax
1153 \xright@appenditem{\the\@l@tempcnta}\to\actions@list
1154 \fi}
1155 %

```

\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.

```

1156
1157 \newcommand*\sub@action{%
1158 \ifledRcol
1159 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1160 \ifsublines@
1161 \xright@appenditem{-1001}\to\actions@listR
1162 \else
1163 \xright@appenditem{-1002}\to\actions@listR
1164 \fi
1165 \else
1166 \xright@appenditem{\the\absline@num}\to\actionlines@list
1167 \ifsublines@
1168 \xright@appenditem{-1001}\to\actions@list
1169 \else
1170 \xright@appenditem{-1002}\to\actions@list

```

```

1171 \fi
1172 \fi}
1173 %

```

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

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

```

1174 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1175
1176 \newcommand*{\do@lockon}{%
1177   \ifx\next\lock@off
1178     \global\let\lock@off=\skip@lockoff
1179   \else
1180     \ifledRcol
1181       \do@lockonR
1182     \else
1183       \do@lockonL
1184     \fi
1185   \fi}
1186
1187
1188 \newcommand*{\do@lockonL}{%
1189   \xright@appenditem{\the\absline@num}\to\actionlines@list
1190   \ifsublines@
1191     \xright@appenditem{-1005}\to\actions@list
1192     \ifnum\sub@lock=\z@
1193       \sub@lock \@ne
1194     \else
1195       \ifnum\sub@lock=\thr@@
1196         \sub@lock \@ne
1197       \fi
1198     \fi
1199   \else
1200     \xright@appenditem{-1003}\to\actions@list
1201     \ifnum\@lock=\z@
1202       \@lock \@ne
1203     \else
1204       \ifnum\@lock=\thr@@
1205         \@lock \@ne
1206       \fi
1207     \fi
1208   \fi}
1209 %
1210 %

```

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

```

1211 \do@lockoff
1212 \do@lockoffL
1213 \skip@lockoff
1214 \newcommand*{\do@lockoffL}{%
1215   \xright@appenditem{\the\absline@num}\to\actionlines@list
1216   \ifsublines@
1217     \xright@appenditem{-1006}\to\actions@list
1218     \ifnum\sub@lock=\tw@
1219       \sub@lock \thr@@
1220     \else
1221       \sub@lock \z@
1222     \fi
1223   \else
1224     \xright@appenditem{-1004}\to\actions@list
1225     \ifnum\@lock=\tw@
1226       \@lock \thr@@
1227     \else
1228       \@lock \z@
1229     \fi
1230   \fi}
1231 \newcommand*{\do@lockoff}{%
1232   \reset@current@annot%
1233   \ifledRcol
1234     \do@lockoffR
1235   \else
1236     \do@lockoffL
1237   \fi}
1238 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1239 \global\let\lock@off=\do@lockoff
1240 %
1241 %

```

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

```

1240 \newcommand*{\n@num}{%
1241   \ifledRcol%
1242     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1243     \xright@appenditem{-1007}\to\actions@listR
1244   \else%
1245     \xright@appenditem{\the\absline@num}\to\actionlines@list%
1246     \xright@appenditem{-1007}\to\actions@list%
1247   \fi%
1248 }%
1249 %
1250 %

```

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


```

1251 \newcommand*{\n@num@stanza}{%
1252   \ifledRcol%
1253     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1254     \xright@appenditem{-1008}\to\actions@listR%
1255   \else%
1256     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1257     \xright@appenditem{-1008}\to\actions@list%
1258   \fi%
1259 }
1260 %

```

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

```

\hidenumberingonleftpage
\hidenumberingonrightpage
1261 \newif\ifl@dhidenumber
1262 \newcommand*{\hidenumbering}{%
1263   \ifledRcol%
1264     \write\linenum@outR{\string\hide@num}%
1265   \else%
1266     \write\linenum@out{\string\hide@num}%
1267   \fi%
1268 }%
1269 \newcommand*{\hide@num}{%
1270   \ifledRcol%
1271     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1272     \xright@appenditem{-1009}\to\actions@listR%
1273   \else%
1274     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1275     \xright@appenditem{-1009}\to\actions@list%
1276   \fi%
1277 }
1278 \newcommand*{\hidenumberingonleftpage}{%
1279   \ifledRcol%
1280     \write\linenum@outR{\string\hide@num@left}%
1281   \else%
1282     \write\linenum@out{\string\hide@num@left}%
1283   \fi%
1284 }%
1285
1286 \newcommand*{\hide@num@left}{%
1287   \ifledRcol%
1288     \ifodd\page@numR\else%
1289       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1290       \xright@appenditem{-1009}\to\actions@listR%
1291     \fi%
1292   \else%
1293     \ifodd\page@num\else%
1294       \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1295       \xright@appenditem{-1009}\to\actions@list%

```

```

1296 \fi%
1297 \fi%
1298 }%
1299
1300 \newcommand*\hidenumberingonrightpage}{%
1301 \ifledRcol%
1302 \write\linenum@outR{\string\hide@num@right}%
1303 \else%
1304 \write\linenum@out{\string\hide@num@right}%
1305 \fi%
1306 }%
1307
1308 \newcommand*\hide@num@right}{%
1309 \ifledRcol%
1310 \ifodd\page@numR%
1311 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1312 \xright@appenditem{-1009}\to\actions@listR%
1313 \fi%
1314 \else%
1315 \ifodd\page@num%
1316 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1317 \xright@appenditem{-1009}\to\actions@list%
1318 \fi%
1319 \fi%
1320 }%
1321
1322 %

```

\@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.

```

1323 \newcount\insert@count
1324 %

```

- #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.

```

1325 \newcommand*\dummy@ref}[2]{#2}
1326 %

```

\@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.

```

1327 \newcommand*{\@ref}[2]{%
1328   \ifledRcol%
1329     \@ref@regR{#1}{#2}%
1330   \else%
1331     \@ref@reg{#1}{#2}%
1332   \fi%
1333 }%
1334 \newcommand*{\@ref@reg}[2]{%
1335   \global\insert@count=#1\relax
1336   \global\advance\@edtext@level by 1%
1337   \loop\ifnum\insert@count>\z@
1338     \xright@appenditem{\the\absline@num}\to\insertlines@list
1339     \global\advance\insert@count \m@ne
1340   \repeat
1341   %

```

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.

```

1342 \begingroup
1343   \let\@ref=\dummy@ref
1344   \let\@lopL\@gobble
1345   \let\page@action=\relax
1346   \let\sub@action=\relax
1347   \let\set@line@action=\relax
1348   \let\@lab=\relax
1349   \let\@lemma=\relax%
1350   \let\@sw\@gobblethree%
1351   \let\store@annot@to@absline\@gobble%
1352   #2
1353   \global\endpage@num=\page@num
1354   \global\endline@num=\line@num
1355   \global\endsubline@num=\subline@num
1356   \global\let\endcurrent@annot=\current@annot%
1357 \endgroup
1358 %

```

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

```

1359 \xright@appenditem%
1360   {\the\page@num|\the\line@num}%
1361   \ifsublines@ \the\subline@num \else 0\fi%
1362   \the\endpage@num|\the\endline@num}%
1363   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1364 \xright@appenditem%
1365   {\current@annot|\endcurrent@annot}\to\annot@list%
1366 %

```

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

```

1367 \@ref@reg@parse{#2}%
1368 %
Decrease edtext level counter.
1369 \global\advance\@edtext@level by -1%
1370 }
1371 %

```

\@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.

```

1372 \newcommand{\@ref@reg@parse}[1]{%
1373 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
\@edtext@level\endcsname}%
1374 \providebool{lemmacommand@\the\@edtext@level}%
1375 \boolfalse{lemmacommand@\the\@edtext@level}%
1376 %

```

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

```

1377 #1%
1378 %

```

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.

```

1379 \ifnum\@edtext@level>0%
1380 \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
\csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1381 \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level}%
1382 \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
1383 \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}%
1384 \xright@appenditem{\expandonce\@tmpp}{to\@tmp}%
1385 \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
1386 \fi%
1387 %
1388 }
1389 %
1390 %

```

\@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.

```

1391 \newcommand{\@ref@later}[1]{%
1392 \global\advance\@edtext@level by \@ne%
1393 \ifledRcol%

```

```

1394 \@ref@reg@parseR{#1}%
1395 \else%
1396 \@ref@reg@parse{#1}%
1397 \fi%
1398 \global\advance\@edtext@level by -\@ne%
1399 }%
1400 %

```

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`.

```

1401 \newwrite\linenum@out
1402 %

```

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

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

```

1403 \newif\iffirst@linenum@out
1404 \first@linenum@out@true
1405 %

```

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

```

1406 \newcommand{\this@line@list@version}{7}%
1407 %

```

`\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.

`\next@line@list@stuff`

```

1408 \let\next@line@list@stuff\relax%
1409 \newcommand*{\line@list@stuff}[1]{%
1410 %

```

First, define a toggle set to true when we are not in the first run.

```

1411 \global\newtoggle{notfirststrun@#1}%
1412 \IfFileExists{\l@auxdir#1}%
1413   {\global\toggletrue{notfirststrun@#1}}%
1414   {\global\togglefalse{notfirststrun@#1}}%
1415 %

```

A internal hook (not used yet).

```

1416 \next@line@list@stuff%
1417 \global\let\next@line@list@stuff\relax%
1418 %

```

Use the commands of the previous section to interpret the line-list file from the last run.

```

1419 \read@linelist{#1}%
1420 %

```

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. `reledmac` and `reledpar` can fill the `\next@line@list@stuff` hook between a `\endnumbering` (associated with numbered file n) and a `\beginnumbering` (associated with numbered file $n + 1$). It allows adding content to the numbered file $n + 1$ and not n .

```

1421 \iffirst@linenum@out@
1422   \global\first@linenum@out@false%
1423   \immediate\openout\linenum@out=\l@auxdir#1\relax%
1424   \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1425   \ifl@dpaging%
1426     \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
1427   \fi%
1428   \else
1429 %

```

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.

```

1430   \if@minipage%
1431     \leavevmode%
1432   \fi%
1433   \closeout\linenum@out%
1434   \openout\linenum@out=\l@auxdir#1\relax%
1435   \write\linenum@out{\string\line@list@version{\this@line@list@version}}%
1436 %
1436   \ifl@dpaging%

```

```

1437 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
1438 %
1438 \fi%
1439 \fi%
1440 }%
1441 %
1442 %

```

`\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. It writes the two forms of the page number:

- Raw form (`\the\c@page`, which can be used to make numeric test).
- Formatted form (for example, in Roman).

Actually, only the first form is effectively used by `\@nl`. If we use the `\sameparallelpagenumber` option of `reledpar`, we must write not the real page number (i.e. page counter, defined in standard \LaTeX) but the printed page number (i.e. `par@page` counter, defined only on `reledmac`).

- For the raw form, we use `\the\c@page@page` macro, because the `{par@page}` counter is increased at each page.
- For the formatted version, `\thepage` is patched through `\par@patch@thepage`. So we have nothing to change.

```

1443 \newcommand*{\new@line}{%
1444 \IfStrEq{\led@pb@setting}{after}%
1445 {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1446 {\xifinlist{\the\absline@num}{\normal@page@break}%
1447 {\numgdef{\@next@page}{\c@par@page+\@ne}%
1448 \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%
1449 }%
1450 {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}%
1451 }%
1452 {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}}%
1453 }%
1454 \IfStrEq{\led@pb@setting}{before}%
1455 {\numgdef{\next@absline}{\the\absline@num+\@ne}%
1456 \xifinlist{\next@absline}{\l@prev@nopb}%
1457 {\xifinlist{\the\absline@num}{\normal@page@break}%
1458 {\numgdef{\nc@page}{\c@par@page+\@ne}%
1459 \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1460 }%
1461 {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}%
1462 }%
1463 {\write\linenum@out{\string\@nl[\the\c@par@page][\thepage]}}}%
1464 }%
1465 {}%
1466 \IfStrEqCase{\led@pb@setting}%

```

```

1467     {%
1468     {before}{\relax}%
1469     {after}{\relax}%
1470   }[%
1471     \write\linenum@out{\string\@nl[\the\c@par@page][\thepage]]}%
1472   ]%
1473 }
1474
1475 \newcommand{\l@the\c@page}{%
1476   \ifboolexpr{%
1477     bool{sameparallelpagenumber}%
1478     or bool{prevpgnotnumbered}%
1479   }%
1480   {\the\c@par@page}%
1481   {\the\c@page}%
1482 }%
1483 %

```

\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

```

1484 \newif\if@noneed@Footnote%
1485
1486 \newcommand*{\flag@start}{%
1487   \ifledRcol%
1488     \edef\next{\write\linenum@outR{%
1489       \string\@ref[\the\insert@countR][\thepage]}%
1490     }%
1491     \ifnum\insert@countR<1%
1492       \if@noneed@Footnote\else%
1493         \led@err@EdtextWithoutFootnote%
1494       \fi%
1495     \fi%
1496   \else%
1497     \edef\next{\write\linenum@out{%
1498       \string\@ref[\the\insert@count][\thepage]}%
1499     }%
1500     \ifnum\insert@count<1%
1501       \if@noneed@Footnote\else%
1502         \led@err@EdtextWithoutFootnote%
1503       \fi%
1504     \fi%
1505   \fi}%
1506
1507 \newcommand*{\flag@end}{%
1508   \ifledRcol%

```



```

1509 \write\linenum@outR{}}%
1510 \else%
1511 \write\linenum@out{}}%
1512 \fi}%
1513
1514
1515 %

```

`\flag@start@RTL` With \LaTeX , there is a problem when using RTL: the writing of a command in the numbered auxiliary files (.1, .2 etc) is reversed when the first argument of `\edtext` is typset in one line, but it is **not** reversed when this first argument is typset in two lines or more.²⁸

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

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

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

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

```

1516 \newcommand{\flag@start@RTL}{%
1517 \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1518 \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1519 {000}}%
1520 {}%
1521 {%
1522 \ifnumequal%
1523 {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1524 {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1525 {\flag@end}%
1526 {\flag@start}%
1527 }%
1528 }%
1529
1530 \newcommand{\flag@end@RTL}{%
1531 \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}%

```

²⁸This problem is caused by the way \LaTeX manages right-to-left typesetting. David Carlisle explains it on <http://tex.stackexchange.com/a/333373/7712> and provides a potential solution, using `\vadjust`. However in some cases this adds spurious vertical spaces in `reledmac`. That is why we are using the solution explained below.

```

1532 \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level
    }}}%
1533     {000}%
1534     {}%
1535     {%
1536     \ifnumequal%
1537         {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1538         {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1539         {\flag@start}%
1540         {\flag@end}%
1541     }%
1542 }%
1543 %

```

`\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

```

1544 \newcommand*{\flag@start@later}{%
1545     \ifledRcol%
1546         \write\linenum@outR{\string\@ref@later []}%
1547     \else%
1548         \write\linenum@out{\string\@ref@later []}%
1549     \fi%
1550 }%
1551 \newcommand{\flag@end@later}{%
1552     \ifledRcol%
1553         \write\linenum@outR{[]}%
1554     \else%
1555         \write\linenum@out{[]}%
1556     \fi%
1557 }
1558 %

```

`\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.

```

1559
1560
1561 \newcommand*{\startsub}{\dimen0\lastskip

```

```

1562 \ifdim\dimen0>0pt \unskip \fi
1563 \ifledRcol \write\linenum@outR{\string\sub@on}%
1564 \else      \write\linenum@out{\string\sub@on}%
1565 \fi
1566 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1567 \def\endsub{\dimen0\lastskip
1568 \ifdim\dimen0>0pt \unskip \fi
1569 \ifledRcol \write\linenum@outR{\string\sub@off}%
1570 \else      \write\linenum@out{\string\sub@off}%
1571 \fi
1572 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1573
1574 %

```

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

```

1575 \newcommand*{\advanceline}[1]{\leavevmode%
1576 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1577 \else      \write\linenum@out{\string\@adv[#1]}%
1578 \fi%
1579 }
1580 %

```

\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.

```

1581
1582 \newcommand*{\setline}[1]{%
1583 \leavevmode%
1584 \ifnum#1<\z@
1585 \led@warn@BadSetline%
1586 \else%
1587 \ifledRcol%
1588 \write\linenum@outR{\string\@set[#1]}%
1589 \else%
1590 \write\linenum@out{\string\@set[#1]}%
1591 \fi%
1592 \fi%
1593 }%
1594
1595 %

```

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

```

1596
1597 \newcommand*{\setlinenum}[1]{%
1598 \ifnum#1<\z@
1599 \led@warn@BadSetlinenum

```

```

1600 \else
1601 \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1602 \else \write\linenum@out{\string\l@d@set[#1]} \fi
1603 \fi}
1604
1605 %

```

`\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.

```

1606
1607 \newcommand*{\startlock}{%
1608 \ifledRcol \write\linenum@outR{\string\lock@on}%
1609 \else \write\linenum@out{\string\lock@on}%
1610 \fi}
1611 \def\endlock{%
1612 \ifledRcol \write\linenum@outR{\string\lock@off}%
1613 \else \write\linenum@out{\string\lock@off}%
1614 \fi}
1615 %

```

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

`\ifl@dskipversenumber`

`\l@dskipnumbertrue`

`\l@dskipnumberfalse`

`\skipnumbering`

```

1616 \newif\ifl@dskipnumber
1617 \newif\ifl@dskipversenumber%
1618 \newcommand*{\skipnumbering}{%
1619 \leavevmode%
1620 \ifledRcol%
1621 \ifinstanza%
1622 \write\linenum@outR{\string\n@num@stanza}%
1623 \else%
1624 \write\linenum@outR{\string\n@num}%
1625 \fi%
1626 \advanceline{-1}%
1627 \else%
1628 \ifinstanza%
1629 \write\linenum@out{\string\n@num@stanza}%
1630 \else%
1631 \write\linenum@out{\string\n@num}%
1632 \fi%
1633 \advanceline{-1}%
1634 \fi%
1635 }%
1636
1637 %

```

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

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

VI.1 `\edtext` itself

The various note-generating macros might want to request that commands be executed not at once, but in close connection with the start or end of the lemma. For example,

footnote numbers in the text should be connected to the end of the lemma; or, instead of a single macro to create a note listing variants, you might want to use several macros in series to create individual variants, which would each add information to a private macro or token register, which in turn would be formatted and output when all of #2 for the lemma has been read.

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

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

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

```
1638 \list@create{\end@lemmas}
1639 %
```

`\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.

```
1640 \newcommand{\dummy@edtext}[2]{#1}
1641 %
```

`\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`.

```
1642 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1643 %
```

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

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

The first class is font-changing macros. We suppress expansion for them by letting them become equal to zero.²⁹ This is done because we want to pass into our notes the

²⁹Since ‘control sequences equivalent to characters are not expandable’—*The TeXbook*, answer to Exercise 20.14.

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.

The `\newseries` 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 `XY \TeX` .)

```

1644 \newcommand*{\no@expands}{%
1645   \let\select@lemmafnt=0%
1646   \let\startsub=\relax \let\endsub=\relax
1647   \let\startlock=\relax \let\endlock=\relax
1648   \let\edlabel=\@gobble
1649   \let\setline=\@gobble \let\advanceline=\@gobble
1650   \let\sameword\sameword@inedtext%
1651   \let\edtext=\dummy@edtext
1652   \let\edindex\dummy@edindex%
1653   \l@dtabnoexpands
1654   \let\linenumannotation=\@gobble%
1655   \morenoexpands}

```

```

1656 \let\morenoexpands=\relax
1657
1658 %

```

\@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.

```

1659 \newcommand{\@tag}{}
1660 %

```

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

```

1661 \newcount\@edtext@level%
1662 \@edtext@level=0%
1663 %

```

\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.

```

1664 \newif\if@edtext@secondarg%
1665 %

```

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

```

1666 \newcounter{edtext}
1667 \renewcommand{\theedtext}{edtxt@arabic{edtext}}%
1668 %

```

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

```

1669 \newcommand{\edtext}[2]{\leavevmode%
1670 %

```

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

```

1671 \ifnumberedpar%
1672 %

```

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

```

1673 \global\advance\@edtext@level by 1%
1674 %

```

We also increase the edtext L^AT_EX counter to insert a hypertarget if the hyperref package is loaded, and also works with \edtext on right-to-left typesetting with Xe_L^AT_EX.

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


```

1675 \stepcounter{edtext}%
1676 \csxdef{thisedtext@the\@edtext@level}{\theedtext}%
1677 %

```

By default, we do not use `\lemma`

```

1678 \global\@lemmacommand@false%
1679 %

```

```

1680 \begingroup%
1681 %

```

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`.

```

1682 \ifledRcol%
1683 \ifcvoid{sw@list@edtextR@the\@edtext@level}%
1684 {\global\let\sw@inthisedtext\empty}%
1685 {\expandafter\gl@p\csname sw@list@edtextR@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1686 \else%
1687 \ifcvoid{sw@list@edtext@the\@edtext@level}%
1688 {\global\let\sw@inthisedtext\empty}%
1689 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1690 \fi%
1691 %

```

\@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.

```

1692 \global\renewcommand{\@tag}{%
1693 \no@expands #1%
1694 }%
1695 %

```

\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`.

```

1696 \set@line%
1697 %

```

`\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`.

```

1698     \ifledRcol \global\insert@countR \z@%
1699     \else     \global\insert@count \z@ \fi%
1700 %

```

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.

```

1701     \@edtext@secondarg@true%
1702     \ignorespaces #2\relax%
1703     \@edtext@secondarg@false%
1704 %

```

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

```

1705     \if@RTL%
1706     \flag@start@RTL%
1707     \else%
1708     \flag@start%
1709     \fi%
1710 %

```

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

```

1711     \if@lemmacommand%
1712     \ifledRcol%
1713     \write\linenum@outR{\string\@lemma}%
1714     \else%
1715     \write\linenum@out{\string\@lemma}%
1716     \fi%
1717     \fi%
1718 %

```

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.

```

1719     \endgroup%
1720     \ifdef{\hypertarget}%
1721     {%
1722     \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\@edtext@level}:start}{}}%
1723     \showlemma{#1}%
1724     \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:end}{}}%

```

```

1725     }%
1726     {%
1727     \showlemma{#1}%
1728     }%
1729 %

```

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.

```

1730     \ifx\end@lemmas\empty \else%
1731     \glp\end@lemmas\to\x@lemma%
1732     \x@lemma%
1733     \global\let\x@lemma=\relax%
1734     \fi%
1735     \if@RTL%
1736     \flag@end@RTL%
1737     \else%
1738     \flag@end%
1739     \fi%
1740 %

```

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`.

```

1741     \global\@noneed@Footnotefalse%
1742     \global\advance\@edtext@level by -1%
1743     \global\@lemmacommand@false%
1744 %

```

We also reset `\@beforeinsertofthisedtext`

```

1745     \global\let\@beforeinsertofthisedtext\relax%
1746 %

```

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

```

1747     \else%
1748     \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidestart%
1749     \fi%
1750 }%
1751
1752
1753 %

```

`\@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`.

```
1754 \let\@beforeinsertofthistext\relax
1755 %
```

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

```
1756 \newif\ifnumberline
1757 \numberlinetrue
1758 %
```

`\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.

```
1759 \newcommand*\set@line{%
1760   \ifledRcol
1761     \ifx\line@listR\empty
1762       \global\noteschanged@true
1763       \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1764     \else
1765       \glp\line@listR\to\@tempb
1766       \xdef\l@d@nums{\@tempb|\edfont@info}%
1767       \global\let\@tempb=\undefined
1768     \fi
1769     \ifx\annot@listR\empty%
1770       \xdef\l@current@annotR{||}%
1771     \else%
1772       \glp\annot@listR\to\@tempb%
1773       \xdef\l@current@annotR{\@tempb}%
1774       \global\let\@tempb=\undefined%
1775     \fi%
1776   \else
1777     \ifx\line@list\empty
1778       \global\noteschanged@true
1779       \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
1780     \else
1781       \glp\line@list\to\@tempb
1782       \xdef\l@d@nums{\@tempb|\edfont@info}%
1783       \global\let\@tempb=\undefined
1784     \fi
```

```

1785 \ifx\annot@list\empty%
1786 \xdef\l@current@annot{||}%
1787 \else%
1788 \gl@p\annot@list\to\@tempb%
1789 \xdef\l@current@annot{\@tempb}%
1790 \global\let\@tempb=\undefined%
1791 \fi%
1792 \fi}
1793
1794 %

```

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

```

1795 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1796
1797 %

```

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. 137).

```

1798 \newcommand*{\lemma}[1]{%
1799 \global\@lemmacommand@true%
1800 \global\renewcommand{\@tag}{%
1801 \no@expands #1%
1802 }%
1803 \ignorespaces%
1804 }%
1805 %

```

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

```

1806 \newcommand{\@lemma}{%
1807 \booltrue{lemmacommand@the\@edtext@level}%
1808 }%
1809 %

```

\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.

```

1810 \newif\if@lemmacommand@%
1811 %

```

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. 106): 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.

```
1812 \newcommand*{\linenum}[1]{%
1813   \xdef\@tempa{#1|||||\\noexpand\\l@d@nums}%
1814   \global\let\l@d@nums=\empty
1815   \expandafter\line@set\@tempa|\\ignorespaces}
1816 %
```

\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.

```
1817 \def\line@set#1|#2\\#3|#4\\{%
1818   \gdef\@tempb{#1}%
1819   \ifx\@tempb\empty
1820     \l@d@add{#3}%
1821   \else
1822     \l@d@add{#1}%
1823   \fi
1824   \gdef\@tempb{#4}%
1825   \ifx\@tempb\empty\else
1826     \l@d@add{|}\line@set#2\\#4\\%
1827   \fi}
1828 %
```

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

```
1829 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1830
1831 %
```

\lineannot Last but not least, `\lineannot` allows us to substitute line number annotation. It is different from `\linenum` for backward compatibility with older versions of `reledmac`. It calls `\lineannot@set` to determine whether we must change only one annotation or two, or none.

```

1832 \newcommand*{\lineannot}[1]{%
1833   \lineannot@set#1|}%
1834 }%
1835 \def\lineannot@set#1|#2|{%
1836   \expandafter\parse@annot#1|#2|}%
1837   \IfStrEq{#1}{}%
1838     {\let\@tempa\annot@start}%
1839     {\def\@tempa{#1}}}%
1840   \IfStrEq{#2}{}%
1841     {\let\@tempb\annot@start}%
1842     {\def\@tempb{#2}}}%
1843   \xdef\l@current@annot{\@tempa|\@tempb}%
1844 }%
1845 %

```

VI.4 Lemma disambiguation

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

The general mechanism is the following:

- **At the first run**, each `\sameword` command increments an `etoolbox` counter the name of which contains the argument of the `\sameword` commands.
- Then this counter, associated with the argument of `\sameword` is stored with the `\@sw` command in the auxiliary file of the current `reledmac` section (the `.1`, `.2...` file).
- **When this auxiliary file is read at the second run**, different operations are achieved:
 1. Get the rank of each `\sameword` in a line (relative rank) from the rank of each `\sameword` in all the numbered section (absolute rank):
 - For each paired `\sameword` argument and absolute line number, a counter is defined. Its value corresponds to the number of times `\sameword{<argument>}` is called from the beginning of the lineation to the end of the current line. We also store the same data for the preceding absolute line number, if it does not have `\sameword{<argument>}`.
 - For each `\sameword` having the same argument, we subtract from its absolute rank the number stored for the paired `\sameword` argument and previous absolute line number. Consequently, we obtain the relative rank.
 - See the following example which explains how, for same `\sameword`, absolute ranks are transformed to relative ranks.

At line 1:

absolute rank 1 becomes relative rank $1-0 = 1$

```

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

```

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

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

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

In this example, all lemma information for \edtext is framed. The text in parenthesis is the content of critical notes associated to the preceding frame. As you can see, we have two levels 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`.³⁰

Because there is a bug with `\detokenize` and \XeTeX when using non BMP characters³¹, we detokenize only for non- \XeTeX engines. In any case, in \XeTeX a `\csname` construction can contain UTF-8 characters without a problem, as UTF-8 characters are not managed with category codes, but instead read directly as UTF-8 characters.

```

1846 \newcommand{\get@sw@txt}[1]{%
1847   \begingroup%
1848   \swnoexpands%
1849   %.
```

Using case sensibility option.

```

1850   \ifsw@caseinsensitive%
1851   \def\@tmpa##1{\lowercase{##1}}%
1852   \else%
1853   \def\@tmpa##1{##1}%
1854   \fi%
1855   %
```

And now, define `\sw@txt`.

```

1856   \ifxetex%
1857   \@tmpa{\xdef\sw@txt{##1}}%
```

³⁰See <http://tex.stackexchange.com/q/244538/7712>.

³¹<http://sourceforge.net/p/xetex/bugs/108/>

```

1858 \else%
1859 \tmpa{\expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize
1860 {#1}}}%
1861 \fi%
1862 \endgroup%
1863 }%
1864 %

```

Allow some macros inside `\sameword`. We use `\RenewExpandableDocumentCommand` to get expandable command with optional argument. Cf. <https://tex.stackexchange.com/a/384783/7712>.

```

\sw@noexpand \newcommand{\swnoexpands}{%
1865 \let\sameword\@firstofone% Allow to have nested \sameword
1866 \let\emph\@firstofone%
1867 \let\textit\@firstofone%
1868 \let\textbf\@firstofone%
1869 \let\textsc\@firstofone%
1870 \let\framebox\@firstofone%
1871 \RenewExpandableDocumentCommand{\edindex}{om}{}%
1872 \ifdefined\index%
1873 \RenewExpandableDocumentCommand{\index}{om}{}%
1874 \fi%
1875 \let\selectlanguage\@gobble%
1876 \let\foreignlanguage\@secondoftwo%
1877 \ifdefined\xpg@loaded%
1878 \renewcommand\do[1]{%
1879 \expandafter\RenewExpandableDocumentCommand\expandafter{\csname
text##1\endcsname}{om}{###2}}%
1880 }%
1881 \expandafter\docsvlist\expandafter{\xpg@loaded}%
1882 \fi%
1883 }%
1884 %

```

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

```

1885 \newcommandx{\sameword}[2][1,usedefault]{%
1886 \leavevmode%
1887 \get@sw@txt{#2}%
1888 %

```

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

```

1889 \unless\ifledRcol%
1890 \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1891 %

```

Then, write its value to the numbered file.

```

1892 \protected@write\linenum@out{}\string\@sw{\sw@txt}{\csuse{sw@\sw@txt
}}{#1}}%
1893 %

```

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

```

1894 \else%
1895 \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1896 \protected@write\linenum@outR{}\string\@sw{\sw@txt}{\csuse{sw@\sw@txt
}}{#1}}%
1897 \fi%
1898 %

```

And print the word.

```

1899 #2%
1900 }%
1901 %

```

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

```

\if@addsw%2 \newif\if@addsw%
1903 %

```

\@sw The command printed in the auxiliary files.

```

1904 \newcommand{\@sw}[3]{%
1905 \get@sw@txt{#1}%
1906 \unless\ifledRcol%
1907 %

```

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

```

1908 \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
1909 %

```

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

```

1910 \numdef{\prev@line}{\the\absline@num-1}%
1911 \ifcsundef{sw@\sw@txt @\prev@line @\the\section@num}{%
1912 \csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
1913 }{}%
1914 %

```

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

```

1915 \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1916 %

```

And do the same thing for the right side.

```

1917 \else%
1918 \csxdef{sw@sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1919 \numdef{\prev@line}{\the\absline@numR-1}%
1920 \ifcsundef{sw@sw@txt @\prev@line @\the\section@numR @R}{%
1921 \csnumgdef{sw@sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1922 }{}%
1923 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R
1924 }}%
1925 \fi%
1926 %

```

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

```

1926 \@tempcnta=\@edtext@level
1927 \@whilenum{\@tempcnta>0}\do{%
1928 \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1929 {%
1930 \@addswfalse%
1931 \notbool{lemmacommand@\the\@tempcnta}%
1932 {\@addswtrue}%
1933 {\IfStrEq{#3}{inlemma}%
1934 {\@addswtrue}%
1935 {%
1936 \def\do##1{%
1937 \ifnumequal{##1}{\the\@tempcnta}%
1938 {\@addswtrue\listbreak}%
1939 }%
1940 }%
1941 \docsvlist{#3}%
1942 }%
1943 }%
1944 \if@addsw%
1945 \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1946 \ifledRcol%
1947 \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
1948 \else%
1949 \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
1950 \fi%
1951 \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
1952 \fi%
1953 }%
1954 }%
1955 \advance\@tempcnta by -1%
1956 }%
1957 }%
1958 %

```

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

```

1959 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1960 \get@sw@txt{#2}%

```

```

1961 \unless\ifledRcol@%
1962 %

```

Just a precaution.

```

1963 \ifx\sw@list@inedtext\empty%
1964 \def\the@sw{999}%
1965 \def\this@absline{-99}%
1966 \else%
1967 %

```

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

```

1968 \gl@p\sw@list@inedtext\to\@tmp%
1969 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1970 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1971 \fi%
1972 %

```

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

```

1973 \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
1974 \numdef{\prev@line}{\this@absline-1}%
1975 \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1976 }%
1977 {\numdef{\sw@atthisline}{0}}%
1978 %

```

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

```

1979 \ifnumgreater{\sw@atthisline}{1}%
1980 {\showwordrank{#2}{\the@sw}}%
1981 {#2}%
1982 %

```

And the same for right side.

```

1983 \else%
1984 \ifx\sw@list@inedtext\empty%
1985 \def\the@sw{999}%
1986 \def\this@absline{-99}%
1987 \else%
1988 \gl@p\sw@list@inedtext\to\@tmp%
1989 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1990 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1991 \fi%
1992 \ifcsdef{sw@\sw@txt @\this@absline @\the\section@numR @R}{%
1993 \numdef{\prev@line}{\this@absline-1}%
1994 \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
1995 }%

```

```

1996     {\numdef{\sw@atthisline}{0}}%
1997     \ifnumgreater{\sw@atthisline}{1}%
1998         {\showwordrank{#2}{\the@sw}}%
1999         {#2}%
2000     \fi%
2001 }%
2002 %

```

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

```

2003 \newcommand{\showwordrank}[2]{%
2004     #1\textsuperscript{#2}%
2005 }%
2006 %

```

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.

```

2007 \newbox\raw@text
2008 \newif\ifnumberedpar@
2009 \newcount\num@lines
2010 \newbox\one@line
2011 \newcount\par@line
2012 %

```

`\pstarts@typeset@L` The `\pstarts@typeset@L` counts the number of LEFT `\pstart` typeset. A left `\pstart` can be a `\pstart` typeset in normal mode, or `\pstart` typeset in left column or page in parallel mode (`withreledpar`). In the first case, its value is increased at every `\pstart` command. In the second case, its value is increased in `\Columns` or `\Pages` when we prepare to typeset a left `\pstart`. The `\pstarts@readL` counts the number

of LEFT \pstart read. A left \pstart can be a \pstart typeset in normal mode, or \pstart typset in left column or page in parallel mode (withreledpar). In the first case, its value is increased at every \pstart command. In the second case, its value is increased at every \pstarL.

```
2013 \newcount\pstarts@typeset@L%
2014 \newcount\pstarts@read@L%
2015 %
```

\pstart \pstart starts the paragraph by clearing the \inserts@list list and other relevant variables, and then arranges for the subsequent text to go into the \raw@text box.
\AtEveryPstart \pstart needs to appear at the start of every paragraph that is to be numbered; the
\AtStartEveryPstart \autopar command below may be used to insert these commands automatically.
\numberpstarttrue Beware: everything that occurs between \pstart and \pend is happening within
\numberpstartfalse a group; definitions must be global if you want them to survive past the end of the
\labelpstarttrue paragraph.
\labelpstartfalse

\thepstart

```
2016 \if@every@pstart@star@%
2017 \newcommand{\AtStartEveryPstart}[1]{%
2018   \ifstrempy{#1}%
2019   {\gdef\@at@start@every@pstart{}}%
2020   {\gdef\@at@start@every@pstart{#1}}%
2021 }%
2022 \def\@at@start@every@pstart{}%
2023
2024 \newif\if@every@pstart@star@%
2025 \newcommand{\AtEveryPstart}[1]{%
2026   \ifstrempy{#1}%
2027   {\gdef\@at@every@pstart{}}%
2028   {\gdef\@at@every@pstart{\noindent#1}}%
2029   \global\@at@every@pstart@star@false%
2030 }%
2031 \WithSuffix\newcommand\AtEveryPstart*[1]{%
2032   \ifstrempy{#1}%
2033   {\gdef\@at@every@pstart{}}%
2034   {\gdef\@at@every@pstart{#1}}%
2035   \global\@at@every@pstart@star@true%
2036 }%
2037 \def\@at@every@pstart{}%
2038
2039 \newcounter{pstart}
2040 \renewcommand{\thepstart}{\bfseries\@arabic@c@pstart}. }
2041 \newif\ifnumberpstart
2042 \numberpstartfalse
2043 \newif\iflabelpstart
2044 \labelpstartfalse
2045 \newcommandx*\pstart[2][1,2,usedefault]{%
2046   \normal@pars%
2047   \ifboolexpr{%
2048     test {\ifstrempy{#1}}%
```

```

2049   and test {\ifstrempy{#2}}%
2050 }%
2051   {\at@every@pstart}%
2052   {%
2053     \ifstrempy{#1}{\noindent#1}%
2054     \ifstrempy{#2}{\#2}%
2055   }%
2056   \ifautopar%
2057     \autopar%
2058   \fi%
2059   \ifluatex%
2060     \edef\l@luatextextdir@L{\the\textdir}%
2061   \fi%
2062   \@nbreaktrue%
2063   \ifnumbering \else%
2064     \led@err@PstartNotNumbered%
2065     \beginnumbering%
2066   \fi%
2067   \ifnumberedpar@%
2068     \led@err@PstartInPstart%
2069     \pend%
2070   \fi%
2071   \list@clear{\inserts@list}%
2072   \global\let\next@insert=\empty%
2073   \begingroup\normal@pars%
2074   \global\advance \l@dnumpstartsL\@ne
2075   \global\advance \pstarts@typeset@L\@ne%
2076   \global\advance \pstarts@read@L\@ne%
2077   \global\setbox\raw@text=\vbox\bgroup%
2078     \if@nbreak%
2079       \if@afterindent\else%
2080         \noindent%
2081         \global\@afterindenttrue%
2082       \fi%
2083     \fi%
2084     \ifboolexpr{%
2085       bool{autopar}%
2086       and bool{by@autopar}%
2087     }%
2088     {}%
2089     {%
2090       \ifnumberpstart%
2091         \ifinstanza\else%
2092           \ifsidepstartnum\else%
2093             \thepstart%
2094           \fi%
2095         \fi%
2096       \fi%
2097     }%
2098   \numberedpar@true%

```



```

2099 \iflabelpstart%
2100 \protected@edef\@currentlabel{\p@pstart\thepstart}%
2101 \fi%
2102 \l@dzzeropenalties%
2103 \@at@start@every@pstart%
2104 \global\by@autoparfalse%
2105 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
2106 }
2107 %

```

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

```

2108 \newcommand*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
2109 \led@err@PendNotNumbered%
2110 \fi%
2111 \global\l@dskipversenumberfalse%
2112 \ifnumberedpar@ \else%
2113 \led@err@PendNoPstart%
2114 \fi%
2115 %

```

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.

```

2116 \l@dzzeropenalties%
2117 \@at@end@every@pend%
2118 \endgraf\global\num@lines=\prevgraf\egroup%
2119 \global\par@line=0%
2120 %

```

We check if lineation is by pstart: in this case, we reset the 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.

```

2121 \csnumdef{pstartline}{0}%
2122 \loop\ifvbox\raw@text%
2123 \csnumdef{pstartline}{\pstartline+\@ne}%
2124 \do@line%
2125 \ifbypstart@%
2126 \ifnumequal{pstartline}{1}{%
2127 \bgroup%
2128 \let\leavevmode\relax%
2129 \setline{1}%
2130 \egroup%
2131 \resetprevline@}{}%

```

```

2132 \fi%
2133 \repeat%
2134 %

```

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

```

2135 \flush@notes%
2136 \endgroup%
2137 \ignorespaces%
2138 %

```

Increase `pstart` counter.

```

2139 \ifnumberpstart%
2140 \global\pstartnumtrue%
2141 \fi%
2142 \addtocounter{pstart}{1}%
2143 \ifcontinuousnumberingwithcolumns%
2144 \addtocounter{pstartL}{1}%
2145 \addtocounter{pstartR}{1}%
2146 \fi%
2147 %

```

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

```

2148 \normal@pars%
2149 \ifboolexpr{%
2150   test {\ifstrempy{#1}}%
2151   and test {\ifstrempy{#2}}%
2152 }%
2153 {\at@every@pend}%
2154 {%
2155   \ifstrempy{#1}{\noindent#1}%
2156   \ifstrempy{#2}{#2}%
2157 }%
2158 %

```

Restore standard “nobreak” and “autopar” settings. Normally, `\if@nobreak` is true only immediately after a sectioning command (see `latex.ltx` file). As a `\pstart ... \pend` structure can’t contain any sectioning command, we set `\if@nobreak` to false.

```

2159 \@nobreakfalse%
2160 \ifautopar%
2161 \autopar%
2162 \fi%
2163 }
2164 %

```

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.

```

\AtEveryPend2165
\at@every@pend2166
\ifat@every@pend@star@2167
\newif\ifat@every@pend@star@%
\newcommand{\AtEveryPend}[1]{%
  \ifstrempy{#1}%
    {\gdef\at@every@pend{}}%
    {\gdef\at@every@pend{\noindent#1}}%
  \global\at@every@pend@star@false%
}%
\WithSuffix\newcommand\AtEveryPend*[1]{%
  \ifstrempy{#1}%
    {\gdef\at@every@pend{}}%
    {\gdef\at@every@pend{#1}}%
  \global\at@every@pend@star@true%
}%
\xdef\at@every@pend{}%
%

```

\AtEndEveryPend Here a macro to insert automatically any content at the end of \pend, in numbered lines.

```

2183 \newcommand{\AtEndEveryPend}[1]{%
2184   \ifstrempy{#1}%
2185     {\xdef\@at@end@every@pend{}}%
2186     {\gdef\@at@end@every@pend{#1}}%
2187   }%
2188   \def\@at@end@every@pend{}%
2189   %

```

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

```

2190 \newcommand*{\l@dzeropenalties}{%
2191   \brokenpenalty \z@ \clubpenalty \z@
2192   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
2193   \postdisplaypenalty \z@ \widowpenalty \z@}
2194
2195 %

```

\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.

The boolean `\ifautopar` is set to TRUE while `\autopar` is enabled, and the `\ifby@autopar` is set to TRUE at each `\pstart` automatically called by `\autopar` feature. A manually `\pstart` will have a `\ifby@autopar` set to false.

```

2196 \newif\ifautopar
2197 \newif\ifby@autopar%
2198 \newcommand*{\autopar}{
2199   \ifledRcol
2200     \ifnumberingR \else
2201       \led@err@AutoparNotNumbered
2202       \beginnumberingR
2203       \fi
2204     \else
2205       \ifnumbering \else
2206         \led@err@AutoparNotNumbered
2207         \beginnumbering
2208         \fi
2209       \fi
2210   \autopartrue
2211   \everypar{\setbox0=\lastbox
2212             \endgraf \vskip-\parskip
2213             \global\by@autopartrue%
2214             \pstart \noindent \kern\wd0%
2215             \ifnumberpstart%
2216               \ifinstanza\else%
2217                 \thepstart%
2218               \fi%
2219             \fi%
2220             \let\par=\pend}%
2221   \ignorespaces}
2222 %

```

`\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.

```

2223 \newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}
2224
2225 %

```

\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.

```

2226 \newif\ifautopar@pause
2227 %

```

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. The `\l@dunhbox@line` macro only `\unhbox` one lines, but some package as `\microtype` can override it for custom needs.

```

2228 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
2229 \newcommand*{\do@line}{%
2230   {\vbadness=10000
2231     \splittopskip=\z@
2232     \do@linehook
2233   \l@demptyd@ta
2234     \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2235   \unvbox\one@line \global\setbox\one@line=\lastbox
2236   \getline@num
2237   \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{%}
2238   \ifnum\@lock>\@ne
2239     \inserthangingsymboltrue
2240   \else
2241     \inserthangingsymbolfalse
2242   \fi
2243   \check@pb@in@verse
2244   \ifl@dhiddenumber%
2245     \global\l@dhiddenumberfalse%
2246     \f@x@l@cks%
2247   \else%
2248     \affixline@num%
2249   \fi%
2250   %

```

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

```

2251 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@}%
2252   {\print@eledsection}%
2253   {\print@line}%
2254   \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{%}

```

```
2255 }%
2256 %
```

VII.2.2 Process for “normal” line

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

```
2257 \def\print@line{
2258 %
```

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

```
2259 \affixpstart@num%
2260 %
```

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

```
2261 \hb@xt@ \linewidth{%
2262 %
```

User hook.

```
2263 \do@insidelinehook%
2264 %
```

Left line number

```
2265 \l@dld@ta%
2266 %
```

Prepare text to be inserted before notes.

```
2267 \if@firstlineofpage%
2268 \set@Xtxtbeforenotes%
2269 \set@txtbeforenotesX%
2270 \global\@firstlineofpagefalse%
2271 \fi%
2272 %
```

Insert footnotes made of manuscripts data and critical footnotes.

```
2273 \ifdefstring{\ms@data@position}{msdata-regular}{%
2274 \insert@msdata%
2275 \add@inserts%
2276 \add@Xgroupbyline%
2277 }{%
2278 \add@inserts%
2279 \add@Xgroupbyline%
2280 \insert@msdata%
2281 }%
2282 %
```

Insert marginal notes.

```
2283 \affixside@note%
2284 %
```

Print left notes.

```
2285 \l@dlsn@te
2286 %
```

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

```
2287 {\ledllfill\hb@xt@ \wd\one@line{%
2288 \new@line%
2289 %
```

If we use the continuousnumberingwithcolumns, we increase right line number.

```
2290 \continuousnumberingwithcolumns@sync@linenumber@singletext%
2291 %
```

If we use Lua^LTeX then restore the direction.

```
2292 \ifluatex%
2293 \textdir\l@luatextextdir@L%
2294 \fi%
2295 %
```

Insert, if needed, the hanging symbol.

```
2296 \inserthangingsymbol%
2297 %
```

And so, print the line.

```
2298 \l@dunhbox@line{\one@line}}%
2299 %
```

Right line number

```
2300 \ledrlfill\l@drd@ta%
2301 %
```

Print right notes.

```
2302 \l@drsn@te%
2303 }}%
2304 %
```

And reinsert penalties (for page breaking)...

```
2305 \add@penalties%
2306 }
2307 %
```

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.

```

2308 \def\print@eledsection{%
2309   \if@firstlineofpage%
2310     \set@Xtxtbeforenotes%
2311     \set@txtbeforenotesX%
2312     \global\@firstlineofpagefalse%
2313   \fi%
2314   \ifdefstring{\ms@data@position}{msdata-regular}{}%
2315     \insert@msdata%
2316     \add@inserts%
2317     \add@Xgroupbyline%
2318   }{%
2319     \add@inserts%
2320     \add@Xgroupbyline%
2321     \insert@msdata%
2322   }%
2323   \affixside@note%
2324   \numdef{\temp@}{\pstarts@typeset@L-1}%
2325   \xifinlist{\temp@}{\eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
2326   \@eled@sectioningtrue%
2327   \csuse{eled@sectioning@the\pstarts@typeset@L}%
2328   \@eled@sectioningfalse%
2329   \global\csundef{eled@sectioning@the\pstarts@typeset@L}%
2330   \if@RTL%
2331     \hspace{-3\paperwidth}%
2332     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2333   \else%
2334     \hspace{3\paperwidth}%
2335     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2336   \fi%
2337   \vskip-\baselineskip%
2338   \continuousnumberingwithcolumns@sync@linenumber@singletext%
2339 }
2340 %

```

VII.2.4 Hooks

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

```

2341 \newcommand*{\do@linehook}{}
2342 \newcommand*{\do@insidelinehook}{}
2343 %

```

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

```

2344 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
2345 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
2346

```


2347 %

VII.2.5 Sidenotes and marginal line number initialization

`\l@emptyd@ta` Nulls the `\. . . d@ta`, which may later hold line numbers. Similarly for `\l@dcsnotetext`,
`\l@dld@ta` `\l@dcsnotetext@l`, `\l@dcsnotetext@r` for the texts of the sidenotes, left and right
`\l@drd@ta` notes.
`\l@dcsnotetext`
2348 `\l@dcsnotetext@l` `\newcommand*{\l@emptyd@ta}{%`
2349 `\l@dcsnotetext@r` `\gdef\l@dld@ta{}`
2350 `\gdef\l@drd@ta{}`
2351 `\gdef\l@dcsnotetext@l{}`
2352 `\gdef\l@dcsnotetext@r{}`
2353 `\gdef\l@dcsnotetext{}`
2354
2355 %

`\l@dlsn@te` Zero width boxes of the left and right sidenotes, together with their kerns, and, eventu-
`\l@drsn@te` ally, with additional space if we are in parallel columns typesetting.

2356 `\newcommand{\l@dlsn@te}{%`
2357 `\ifboolexpr{%`
2358 `bool {l@dprintingcolumns}%`
2359 `and bool {ledRcol}%`
2360 `}{%` If we are on a right column
2361 `\@tempdima=\@morespace@leftnote@rightcolumn%`
2362 `}{%`
2363 `\@tempdima=\z@%`
2364 `}%`
2365 `\hb@xt@ \z@{\hss\box\l@dldp@rbox\kern\ledlsnotesep\hskip\@tempdima}%`
2366 `}%`
2367 `\newcommand{\l@drsn@te}{%`
2368 `\ifboolexpr{%`
2369 `bool {l@dprintingcolumns}%`
2370 `and not bool {ledRcol}%`
2371 `}{%` If we are on a left column
2372 `\@tempdima=\@morespace@rightnote@leftcolumn%`
2373 `}{%`
2374 `\@tempdima=\z@%`
2375 `}%`
2376 `\hb@xt@ \z@{\hskip\@tempdima\kern\ledrsnotesep\box\l@drp@rbox\hss}%`
2377 `}%`
2378
2379 %

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

2380 `\newcommand*{\ledllfill}{\hfil}`
2381 `\newcommand*{\ledrlfill}{}`

```
2382
2383 %
```

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.

```
2384 \newcommand*{\getline@num}{%
2385   \global\advance\absline@num \@ne%
2386   \do@actions
2387   \do@ballast
2388   \ifnumberline
2389     \ifsublines@
2390       \ifnum\sub@lock<\tw@
2391         \global\advance\subline@num \@ne
2392       \fi
2393     \else
2394       \ifnum\@lock<\tw@
2395         \global\advance\line@num \@ne
2396         \global\subline@num \z@
2397       \fi
2398     \fi
2399   \fi
2400 }
2401 %
```

\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 XII.2 p. 176).

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

\c@ballast

```
2402 \newcount\ballast@count
2403 \newcounter{ballast}
2404 \setcounter{ballast}{0}
2405 %
```

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.

```
2406 \newcommand*{\do@ballast}{\global\ballast@count \z@
2407   \begingroup
2408     \advance\absline@num \@ne
```

```

2409 \ifnum\next@actionline=\absline@num
2410 \ifnum\next@action>-1001\relax
2411 \global\advance\ballast@count by -\c@ballast
2412 \fi
2413 \fi
2414 \endgroup}
2415 %

```

\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`.

```

2416 \newcommand*{\do@actions}{%
2417 \global\let\do@actions@next=\relax
2418 \ifnum\absline@num<\next@actionline\else
2419 %

```

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.

```

2420 \ifnum\next@action>-1001
2421 \global\page@num=\next@action
2422 \ifresumenummering@start%
2423 \setbox0=\hbox{}}%Required to get the correct page number, when the
\resumenummering is just after a \newpage
2424 \ifnum\pausenumbering@page@num<\page@num%
2425 \global\resumenummering@startfalse%
2426 \fi%
2427 \fi%
2428 \ifboolexpr{%
2429 bool{resumenummering@start}%
2430 and test {\ifdimgreater{\pagedepth}{\z@}}}%
2431 }%
2432 {}%
2433 {\global\@firstlineofpagetrue}%
2434 \ifcsdef{reset@line\the\absline@num @\the\section@num}%
2435 {%
2436 \global\line@num=\z@%
2437 \global\subline@num=\z@%
2438 \resetprevline@%
2439 }%
2440 {}%
2441 \global\resumenummering@startfalse%
2442 \add@msdata@firstlineofpage%
2443 %

```

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`.)

```

2444 \else
2445   \ifnum\next@action<-4999
2446     \@l@dttempcnta=-\next@action
2447     \advance\@l@dttempcnta by -5001
2448     \ifsublines@
2449       \global\subline@num=\@l@dttempcnta
2450     \else
2451       \global\line@num=\@l@dttempcnta
2452     \fi
2453 %

```

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

```

2454 \else
2455   \@l@dttempcnta=-\next@action
2456   \advance\@l@dttempcnta by -1000
2457   \do@actions@fixedcode
2458 \fi
2459 \fi
2460 %

```

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.

```

2461 \ifx\actionlines@list\empty
2462   \gdef\next@actionline{1000000}%
2463 \else
2464   \gl@p\actionlines@list\to\next@actionline
2465   \gl@p\actions@list\to\next@action
2466   \global\let\do@actions@next=\do@actions
2467 \fi
2468 \fi
2469 %

```

Make the recursive call, if necessary.

```

2470 \do@actions@next}
2471
2472 %

```

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

```

2473 \newcommand*{\do@actions@fixedcode}{%
2474   \ifcase\@l@dttempcnta
2475   \or%                               % 1001 = starting sublineation
2476     \global\sublines@true

```

```

2477 \or% % 1002 = ending sublineation
2478 \global\sublines@false
2479 \or% % 1003 = starting locking number
2480 \global\@lock=\@ne
2481 \or% % 1004 = ending locking number
2482 \ifnum\@lock=\tw@
2483 \global\@lock=\thr@@
2484 \else
2485 \global\@lock=\z@
2486 \fi
2487 \or% % 1005 = starting locking subnumber
2488 \global\sub@lock=\@ne
2489 \or% % 1006 = ending locking subnumber
2490 \ifnum\sub@lock=\tw@
2491 \global\sub@lock=\thr@@
2492 \else
2493 \global\sub@lock=\z@
2494 \fi
2495 \or% % 1007 = skipping numbering
2496 \l@dskipnumbertrue
2497 \or% % 1008 = skipping numbering in stanza
2498 \l@dskipversenumbertrue%
2499 \or% % 1009 = hiding number
2500 \l@dhidenumbertrue
2501 \or% % 1010 = inserting msdata
2502 \add@msdata%
2503 \else
2504 \led@warn@BadAction
2505 \fi}
2506
2507
2508 %

```

VIII.1 Continuous line numbering between parallel typesetting and normal typesetting

`\sync@linenumber@singletext` The `\continuousnumberingwithcolumns` option allows to alternate between single text and two parallel texts, keeping same line numbers when switching from one layout to the other one. The lines counter for the text in a single column and for the text in the text in the left column in parallel typesetting is the same. But the lines counter for the text in the same column is not the same.

When typesetting single column text, if the option is enabled, we need to “simulate” typesetting right line, in order to keep the two counters synchronized. That is the aim of the

`\continuousnumberingwithcolumns@sync@linenumber@singletext` macro.

```

2509 \newcommand{\continuousnumberingwithcolumns@sync@linenumber@singletext}{%

```

```

2510 \ifcontinuousnumberingwithcolumns%
2511 \unless\ifafterendnumberingR%
2512 \new@lineR%
2513 \xappto\next@line@list@stuffR{%
2514 \unexpanded{\global\line@numR=}\the\line@num%
2515 }%
2516 \fi%
2517 \fi%
2518 }%
2519 %

```

IX Line number annotation

The `\linenumannotation` allows users to add manual annotations to line numbers, for example to refer to line numbers from an older edition.

That implies that annotations be added

- in marginal line numbers. This feature is implemented by associating annotations with the absolute line numbers when reading the auxiliary numbered files (.1, .2 etc.).
- in critical footnotes. This feature is implemented by associating start / end line number annotations with each `\edtext` when reading the auxiliary numbered files.
- to crossref commands which refer to line numbers (to be completed when I will have resolved this issue).

`\linenumannotation` First, the user-level command, which only writes commands to the numbered files, storing the annotation.

```

2520 \newcommand{\linenumannotation}[1]{%
2521 \leavevmode%In case it begins with a \pstart, ensure the \@annot is
written after \@nl
2522 \ifledRcol%
2523 \write\linenum@outR{\string\@annot[#1]}%
2524 \else%
2525 \write\linenum@out{\string\@annot[#1]}%
2526 \fi%
2527 }%
2528 %

```

`\Xlinenumannotationposition@side` The `\Xlinenumannotationposition@side` macro, to determine the position of line number annotations in lines printed in the side, relative to the line number position. The default value is “after”, but it can be changed to “before”, using the `\Xlinenumannotationposition` macro.

```

2529 \def\Xlinenumannotationposition@side{after}%
2530 %

```

`\xwraplinenumannotation@ref` The `\xwraplinenumannotation@ref` macro is applied to crossrefs made to line annotations. The `\xwraplinenumannotation@side` macro is applied to line annotation in sides. They must be modified by users using `\xwraplinenumannotation`. By default, they call `\textsuperscript`.

```
2531 \def\xwraplinenumannotation@ref{\textsuperscript}%
2532 \def\xwraplinenumannotation@side{\textsuperscript}%
2533 %
```

`\@annot` Then, the numbered files command `\@annot`, which

`\store@annot@to@absline`

- associates the annotation to the absolute lines number.
- stores the current annotation in a macro to be used by the `\@ref@reg` macro, which manages all things related to `\edtext` in numbered files. As we don't want to store it multiple times in the event of nested `\edtext`, we use an auxiliary macro, `\store@annot@to@absline`, which is set to `\@gobble` when we parse nested `\edtext` in numbered auxiliary files.

```
2534 \newcommand{\@annot}[1]{%
2535   \store@annot@to@absline{#1}%
2536   \def\current@annot{#1}%
2537 }%
2538 \newcommand{\store@annot@to@absline}[1]{%
2539   \ifledRcol%
2540     \csgappto{annotR@the\absline@numR @the\section@numR}{#1}%
2541   \else%
2542     \csgappto{annot@the\absline@num @the\section@num}{#1}%
2543   \fi%
2544 }%
2545 %
```

`\current@annot` By default, there is no annotation to a line number, so we store an empty `\current@annot`

```
2546 \let\current@annot=\empty%
2547 %
```

`\reset@current@annot` `\reset@current@annot` is called at each `\@nl` in numbered auxiliary files. It resets the annotation of line numbers at each line.

```
2548 \newcommand{\reset@current@annot}{%
2549   \let\current@annot\empty%
2550 }%
2551 %
```

`\parse@annot` In the `\annot@list` list, each individual `\edtext` is associated with a starting and an ending line number annotation, which is stored this way: `start annotation|end annotation`. The `\parse@annot` reads the format and defines two global macros.

```

2552 \def\parse@annot#1|#2|{%
2553   \gdef\annot@start{#1}%
2554   \gdef\annot@end{#2}%
2555 }%
2556 %

```

X Line number printing

`\affixline@num` `\affixline@num` just puts a left line number into `\l@ldl@ta` or a right line number into `\l@drd@ta` as required.

To determine whether we need to affix a line number to this line, we compute the following:

$$n = \text{int}((\text{linenum} - \text{firstlinenum}) / \text{linenumincrement})$$

$$m = \text{firstlinenum} + (n \times \text{linenumincrement})$$

(where *int* truncates a real number to an integer). *m* will be equal to *linenum* only if we are to paste a number on here. However, the formula breaks down for the first line to number (and any before that), so we check that case separately: if `\line@num ≤ \firstlinenum`, we compare the two directly instead of making these calculations.

We compute, in the scratch counter `\@l@tempcnta`, the number of the next line that should be printed with a number (*m* in the above discussion), and move the current line number into the counter `\@l@tempcntb` for comparison.

First, the case when we are within a sub-line range.

```

2557 \newcommand*{\affixline@num}{%
2558 %

```

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).

```

2559 \ifl@dskipnumber\else
2560   \ifnumberline
2561     \ifl@dskipnumber
2562       \global\l@dskipnumberfalse
2563     \else
2564       \ifsublines@
2565         \@l@tempcntb=\subline@num
2566         \ifnum\subline@num>\c@firstsublinenum
2567           \@l@tempcnta=\subline@num
2568           \advance\@l@tempcnta by-\c@firstsublinenum
2569           \divide\@l@tempcnta by\c@sublinenumincrement
2570           \multiply\@l@tempcnta by\c@sublinenumincrement
2571           \advance\@l@tempcnta by\c@firstsublinenum
2572         \else
2573           \@l@tempcnta=\c@firstsublinenum
2574         \fi
2575       %

```


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.

```
2576 \ch@cksub@l@ck
2577 %
```

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

```
2578 \else
2579 \@l@tempcntb=\line@num
2580 %
```

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

```
2581 \ifx\linenumberlist\empty
2582 \ifnum\line@num>\c@firstlinenum
2583 \@l@tempcnta=\line@num
2584 \advance\@l@tempcnta by-\c@firstlinenum
2585 \divide\@l@tempcnta by\c@linenumincrement
2586 \multiply\@l@tempcnta by\c@linenumincrement
2587 \advance\@l@tempcnta by\c@firstlinenum
2588 \else
2589 \@l@tempcnta=\c@firstlinenum
2590 \fi
2591 \else
2592 %
```

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

```
2593 \@l@tempcnta=\line@num
2594 \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2595 \edef\sc@n@list{\def\noexpand\sc@n@list
2596 ###1,\number\@l@tempcnta,###2|\def\noexpand\rem@inder
2597 {####2}}}%
2598 \sc@n@list\expandafter\sc@n@list\rem@inder|
2599 \ifx\rem@inder\empty%
2600 \advance\@l@tempcnta\@ne
2601 \fi
2602 %
```

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

```
2603 \ch@ck@l@ck
2604 \fi
2605 %
```

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

```
2606 \ifnum\@l@tempcnta=\@l@tempcntb
2607 \ifl@dskipversenumber\else
2608 %
```

If we got here, we are going to print a line number; so now we need to calculate a number that will tell us which side of the page will get the line number. We start from `\line@margin`, which asks for one side always if it is less than 2; and then if the side does depend on the page number, we simply add the page number to this side code—because the values of `\line@margin` have been devised so that this produces a number that is even for left-margin numbers and odd for right-margin numbers.

For \LaTeX we have to consider two column documents as well. In this case Peter Wilson thought we need to put the numbers at the outside of the column — the left of the first column and the right of the second. Do the `twocolumn` stuff before going on with the original code.

`\l@dld@ta` A left line number is stored in `\l@dld@ta` and a right one in `\l@drd@ta`.

```
\l@drd@ta
2609      \if@twocolumn
2610          \if@firstcolumn
2611              \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2612          \else
2613              \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2614          \fi
2615      \else
2616          \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}}%
2617              {\@l@tempcntb=\line@margin@columns}%
2618              {\@l@tempcntb=\line@margin}%
2619              \ifnum\@l@tempcntb>\@ne
2620                  \advance\@l@tempcntb \page@num
2621              \fi
2622              \ifboolexpr{%
2623                  bool {l@dprintingcolumns}%
2624                  and (%
2625                      (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{left
2626                          and test {\ifnumodd{\page@num}}}%
2627                      )%
2628                      or%
2629                      (test {\ifdefstring{\linenum@OnlyPages@ForColumns}{right
2630                          and not test {\ifnumodd{\page@num}}}%
2631                      )%
2632                  )%
2633              }%
2634              {%
2635              {%
2636                  \ifodd\@l@tempcntb%
2637                      \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2638                  \else%
2639                      \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2640                  \fi%
2641              }%
2642          \fi
```

```

2643     \fi
2644 \fi
2645 %

```

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

```

2646     \fix@l@cks
2647 \fi
2648 \fi
2649 \fi
2650 }
2651
2652 %

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck` checks subline locking. If it fails, then we disable the line-number display by setting the counters to arbitrary but unequal values.

```

2653 \newcommand*{\ch@cksub@l@ck}{%
2654 \ifcase\sub@lock
2655 \or
2656 \ifnum\sublock@disp=\@ne
2657 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2658 \fi
2659 \or
2660 \ifnum\sublock@disp=\tw@ \else
2661 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2662 \fi
2663 \or
2664 \ifnum\sublock@disp=\z@
2665 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2666 \fi
2667 \fi}
2668 %

```

Similarly for line numbers.

```

2669 \newcommand*{\ch@ck@l@ck}{%
2670 \ifcase\@lock
2671 \or
2672 \ifnum\lock@disp=\@ne
2673 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2674 \fi
2675 \or
2676 \ifnum\lock@disp=\tw@ \else
2677 \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2678 \fi
2679 \or
2680 \ifnum\lock@disp=\z@
2681 \@l@tempcntb=\z@ \@l@tempcnta=\@ne

```

```

2682         \fi
2683     \fi}
2684 %

```

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

```

2685 \newcommand*{\fix@l@cks}{%
2686     \ifcase\@lock
2687     \or
2688         \global\@lock=\tw@
2689     \or \or
2690         \global\@lock=\z@
2691     \fi
2692     \ifcase\sub@lock
2693     \or
2694         \global\sub@lock=\tw@
2695     \or \or
2696         \global\sub@lock=\z@
2697     \fi}
2698 %
2699 %

```

XI Pstart number printing inside

Inside, the printing of the pstart number runs like the printing of the line number. There are only a few differences:

`\affixpstart@num`
`\pstartnum`

- The pstarts counter is upgraded 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.

```

\leftpstartnum00
\rightpstartnum01 \newif\ifsidepstartnum
\ifsidepstartnum02 \newcommand*{\affixpstart@num}{%
2703     \ifsidepstartnum
2704         \if@twocolumn
2705             \if@firstcolumn
2706                 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2707             \else
2708                 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2709             \fi
2710         \else
2711             \l@dttempcntb=\line@margin%

```

```

2712         \ifnum\@l@dttempcntb>\@ne
2713             \advance\@l@dttempcntb \page@num
2714         \fi
2715         \ifodd\@l@dttempcntb
2716             \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2717         \else
2718             \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2719         \fi
2720     \fi
2721 \fi
2722
2723 }
2724 %
2725
2726 \newif\ifpstartnum
2727 \pstartnumtrue
2728 \newcommand*{\leftpstartnum}{
2729     \ifpstartnum\thepstart
2730     \kern\linenumsep\fi
2731     \global\pstartnumfalse
2732 }
2733 \newcommand*{\rightpstartnum}{
2734     \ifpstartnum
2735     \kern\linenumsep
2736     \thepstart
2737     \fi
2738     \global\pstartnumfalse
2739 }
2740 %

```

XII 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.

XII.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.

```

2741 \list@create{\inserts@list}
2742 %

```

`\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`.

```
2743 \newcommand*{\add@inserts}{%
2744   \global\let\add@inserts@next=\relax
2745 }
```

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

```
2746 \ifx\inserts@list\empty \else
2747 %
```

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.

```
2748 \ifx\next@insert\empty
2749   \ifx\insertlines@list\empty
2750     \global\noteschanged@true
2751     \gdef\next@insert{100000}%
2752   \else
2753     \glp\insertlines@list\to\next@insert
2754   \fi
2755 \fi
2756 %
```

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.

```
2757 \ifnum\next@insert=\absline@num
2758   \glp\inserts@list\to\@insert
2759   \@insert
2760   \global\let\@insert=\undefined
2761   \global\let\next@insert=\empty
2762   \global\let\add@inserts@next=\add@inserts
2763 \fi
2764 \fi
2765 %
```

Make the recursive call, if necessary.

```
2766 \add@inserts@next}
2767
2768 %
```

`\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`.

```

2769 \newcommand{\add@Xgroupbyline}{%
2770   \unless\ifnocritical@%
2771     \def\do##1{%Looping on the series
2772       \def\do####1{%Looping on the ##1@forinserting command
2773         \ifcsdef{##1@forinserting@####1}{%
2774           \X@beforeinsertion{##1}%
2775           \if@ledgroup%
2776             \global\setbox\@nameuse{mp##1footins}=\vbox%
2777           \else%
2778             \insert\csname ##1footins\endcsname%
2779           \fi%
2780           {%
2781             \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
2782               {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
2783             {%
2784               \if@ledgroup%
2785                 \unvbox\@nameuse{mp##1footins}%
2786               \fi%
2787               \X@atbegininsertion{##1}%
2788               \ifcsstring{series@display##1}%
2789                 {%
2790                   \Xledsetnormalparstuff{##1}%
2791                   \rule\z@\splittopskip%
2792                 }%
2793               {%
2794                 \csuse{##1@forinserting@####1}%
2795                 \strut\par%
2796               }%
2797               \global\csundef{##1@forinserting@####1}%
2798             }%
2799           }%
2800         }%
2801         \ifcsdef{##1@forinserting}{%
2802           \dolistcsloop{##1@forinserting}%
2803         }{%
2804           \global\csundef{##1@forinserting}%
2805         }%
2806         \dolistloop{\@series}%
2807       \fi%
2808     }%
2809
2810
2811 %

```

XII.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. 162). Finally, the penalty is checked to see that it does not go below -10000 .

```

2812 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
2813   \ifnum\num@lines>\@ne
2814     \global\advance\par@line \@ne
2815     \ifnum\par@line=\@ne
2816       \advance\@l@tempcnta \clubpenalty
2817     \fi
2818     \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
2819     \ifnum\@l@tempcntb=\num@lines
2820       \advance\@l@tempcnta \widowpenalty
2821     \fi
2822     \ifnum\par@line<\num@lines
2823       \advance\@l@tempcnta \interlinepenalty
2824     \fi
2825   \fi
2826   \ifnum\@l@tempcnta=\z@
2827     \relax
2828   \else
2829     \ifnum\@l@tempcnta>-10000
2830       \penalty\@l@tempcnta
2831     \else
2832       \penalty -10000
2833     \fi
2834   \fi}
2835
2836 %

```

XII.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. For the first run, we do not flush the notes, as that means all the notes will be added at the end of numbered section, and so, very far of the expected position

```

2837 \newcommand*{\flush@notes}{%
2838   \iftoggle{notfirstrun@jobname.\extensionchars\the\section@num}{%

```



```

2839 \xloop%
2840 \ifx\inserts@list\empty \else%
2841 \gl@p\inserts@list\to\@insert%
2842 \@insert%
2843 \global\let\@insert=\undefined%
2844 \repeat%
2845 }{}%
2846 }%
2847
2848
2849 %

```

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

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

```

2850 \def\xloop#1\repeat{%
2851 \def\body{#1\expandafter\body\fi}%
2852 \body}
2853
2854 %

```

XII.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.

```

2855 \newcommand{\set@Xtxtbeforenotes}{%
2856 \unless\ifnocritical%
2857 \def\do##1{%
2858 \nottoggle{Xtxtbeforenotesonlyonce@##1}{%
2859 \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2860 }{}%
2861 }%
2862 \dolistloop{\@series}%
2863 \fi%
2864 }%
2865 %

```

\set@txtbeforenotesX The `\set@txtbeforenotesX` does the same for the `\textbeforenotesX`.

```

2866 \newcommand{\set@txtbeforenotesX}{%
2867 \unless\ifnofamiliar%

```

```

2868 \def\do##1{%
2869   \nottoggle{txtbeforenotesonlyonceX@##1}{%
2870     \global\togglefalse{txtbeforenotesX@##1@typeset}%
2871   }{%
2872   }%
2873   \dolistloop{\@series}%
2874 \fi%
2875 }%
2876 %

```

`\insert@Xtxtbeforenotes` `\insert@Xtxtbeforenotes{<series>}`, called when inserting a familiar footnote, 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.

`\insert@txtbeforenotesX` is the same for familiar footnotes.

```

2877 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2878   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2879     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2880   \ifcsvoid{Xtxtbeforenotes@#1}{%
2881     \ifcsstring{series@display#1}{paragraph}%
2882     {\noindent\csuse{Xtxtbeforenotes@#1}}%
2883     {\expandafter\insert\csname#1footins\endcsname%
2884       \bgroup%
2885         \noindent%
2886         \ifcsdef{\csuse{series@display#1}@begin@insert}{%
2887           \csuse{\csuse{series@display#1}@begin@insert}{#1}%
2888         }{%
2889           \strut\csuse{Xnotefontsize@#1}%
2890           \csuse{Xtxtbeforenotes@#1}%
2891         }%
2892       \egroup%
2893     }%
2894   }%
2895   {}%
2896 }%
2897
2898
2899 \newcommand{\insert@txtbeforenotesX}[1]{%
2900   \nottoggle{txtbeforenotesX@#1@typeset}{%
2901     \global\toggletrue{txtbeforenotesX@#1@typeset}%
2902   \ifcsvoid{txtbeforenotesX@#1}{%
2903     \ifcsstring{series@displayX#1}{paragraph}%
2904     {\noindent\csuse{txtbeforenotesX@#1}}%
2905     {\expandafter\insert\csname#1\endcsname%
2906       \bgroup%
2907         \noindent%
2908         \ifcsdef{\csuse{series@displayX#1}@begin@insert}{%
2909           \csuse{\csuse{series@displayX#1}@begin@insert}{#1}%
2910         }{%

```

```

2911 \strut\csuse{notefontsizeX@#1}\csuse{txtbeforenotesX@#1}%
2912 \egroup%
2913 }%
2914 }%
2915 }%
2916 {}%
2917 }%
2918
2919
2920 %

```

XIII Critical footnotes

The footnote macros are adapted from those in PLAIN T_EX, but they differ in these respects: the outer-level commands must add other commands to a list macro rather than doing insertions immediately; there are many separate levels of the footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

XIII.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.

```

2921 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@lemmafont#7|}
2922 \def\select@@lemmafont#1/#2/#3/#4|%
2923 {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2924 \selectfont}
2925
2926 %

```

XIII.2 Individual note options

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

```

2927 \newcommand*{\footnoteoptions@}[3]{%
2928 \def\do##1{%
2929 \ifstrequal{#1}{L}{% On the left side
2930 \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}\to\inserts@list%

```

```

2931 \global\advance\insert@count \@ne% Increment the left insert
counter.
2932 }%
2933 {%
2934 \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}\to\inserts@listR%
2935 \global\advance\insert@countR \@ne% Increment the right insert
counter insert.
2936 }%
2937 }%
2938 \notblank{#2}{\docsvlist{#2}}}% Parsing all options
2939 }
2940 %

```

XIII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua[®]TeX is used.

```

2941 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
2942 \ifstrequal{#1}{L}{%
2943 \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@list%Know the dir of lemma
2944 \global\advance\insert@count \@ne%
2945 \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@list%Know the dir of lemma
2946 \global\advance\insert@count \@ne%
2947 }%
2948 {%
2949 \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@listR%Know the dir of lemma
2950 \global\advance\insert@countR \@ne%
2951 \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@listR%Know the dir of lemma
2952 \global\advance\insert@countR \@ne%
2953 }%
2954 }
2955 %

```

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

```

2956 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2957 \ifstrequal{#1}{L}{%
2958 \if@RTL%
2959 \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@list%Know the language used in the lemma
2960 \global\advance\insert@count \@ne%
2961 \else

```

```

2962 \xright@appenditem{{\csxdef{footnote@dir}{@RTLfalse}}}\to\
inserts@list%Know the language of lemma
2963 \global\advance\insert@count \@ne%
2964 \fi%
2965 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}}}\
to\inserts@list%Know the language of lemma
2966 \global\advance\insert@count \@ne%
2967 }%
2968 {%
2969 \if@RTL
2970 \xright@appenditem{{\csxdef{footnote@dir}{@RTLtrue}}}\to\
inserts@listR%Know the language of lemma
2971 \global\advance\insert@countR \@ne%
2972 \else
2973 \xright@appenditem{{\csxdef{footnote@dir}{@RTLfalse}}}\to\
inserts@listR%Know the language of lemma
2974 \global\advance\insert@countR \@ne%
2975 \fi
2976 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}}}\
to\inserts@listR%Know the language of lemma
2977 \global\advance\insert@countR \@ne%
2978 }%
2979 }
2980 %

```

XIII.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”.

XIII.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`.

```

2981 \newcommand*{\footsplitskips}{%
2982   \interlinepenalty=\interfootnotelinepenalty
2983   \unless\ifl@dprintingpages%
2984     \floatingpenalty=\@MM%
2985   \fi%
2986   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2987   \leftskip=\z@skip \rightskip=\z@skip}
2988
2989 %

```

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

```

2990 \let\normalfootnoterule=\footnoterule
2991 %

```

XIII.6 Footnotes arrangement

XIII.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.

```

2992 \newcommandx{\Xarrangement}[2][1,usedefault]{%
2993   \def\do##1{%
2994     \csname Xarrangement@#2\endcsname{##1}%
2995   }%
2996   \ifstrempy{#1}%
2997     {%
2998       \dolistloop{\@series}%
2999     }%
3000     {
3001       \docsvlist{#1}%
3002     }%
3003   }%
3004 %

```

XIII.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.

```
3005 \newcommand*{\Xarrangement@normal}[1]{%
3006   \csdef{series@display#1}{normal}
3007   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
3008   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
3009   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
3010   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
3011   \expandafter\let\csname #1footnoterule\endcsname=%
3012                                     \normalfootnoterule
3013   \count\csname #1footins\endcsname=1000
3014   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3015   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3016   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3017   %
```

The `reledpar` provides tools in order to confine notes to one side. The mechanism is explained in the `reledpar`’s handbook. For now, just retain we need to store default value of the counter associated to the notes \TeX ’s inserts.

```
3018   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
3019   side only
3019   %
```

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).

```
3020   \ifnoledgroup@else%
3021     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3022     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
3023     \count\csname mp#1footins\endcsname=1000
```

```

3024 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3025 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3026 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3027 \fi
3028 }
3029
3030 %

```

\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.

```

3031 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
3032 \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
3033 \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
3034 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
3035 \X@beforeinsertion{#1}%
3036 \insert\csname #1footins\endcsname{%
3037 \X@atbegininsertion{#1}%
3038 \normalvfootnote@inserted{#1}{#2}%
3039 }%
3040 }%
3041 }%
3042 %

```

\normalvfootnote@inserted The \normalvfootnote@inserted macro is expanded to the content to be add to a \insert for normal critical footnote.

```

3043 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote@inserted}[2]{%
3044 \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
3045 \csuse{Xnotefontsize@#1}%
3046 \footsplitskips
3047 \ifl@dpairing\ifl@dpaging\else%
3048 \setXnoteswidthliketwocolumns@{#1}%
3049 \fi\fi%
3050 \setXnotespositionliketwocolumns@{#1}%
3051 \spaceskip=\z@skip \xspaceskip=\z@skip%
3052 \csname #1footfmt\endcsname #2{#1}%
3053 }%
3054 %

```

```

\X@beforeinsertion\newcommand{\X@beforeinsertion}[1]{%
3056 \if@ledgroup\else%
3057 \insert@Xtxtbeforenotes{#1}%

```



```

3058 \fi%
3059 \csuse{Xbeforeinserting@#1}%
3060 }%
3061 %

```

```

\beforeinsertion@X62 \newcommand{\beforeinsertion@X}[1]{%
3063 \if@ledgroup\else%
3064 \insert@txtbeforenotesX{#1}%
3065 \fi%
3066 \csuse{beforeinsertingX@#1}%
3067 }%
3068 %

```

```

\X@atbegininsertion69 \newcommand{\X@atbegininsertion}[1]{%
3070 \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3071 }%
3072 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote73 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
3074 \iftoggle{Xgroupbyline@#1}{%
3075 \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
3076 }%
3077 {%
3078 \global\setbox\@nameuse{mp#1footins}%
3079 \vbox{%
3080 \unvbox\@nameuse{mp#1footins}%
3081 \mpnormalvfootnote@inserted{#1}{#2}%
3082 }%
3083 }%
3084 }%
3085 %
3086 %

```

```

\mpnormalvfootnote@inserted87 \newcommand{\mpnormalvfootnote@inserted}[2]{%
3088 \noindent\csuse{Xbhooknote@#1}%
3089 \csuse{Xnotefontsize@#1}%
3090 \hspace\columnwidth%
3091 \@parboxrestore%
3092 \color@begingroup%
3093 \csname #1footfmt\endcsname #2{#1}\color@endgroup%
3094 }%
3095 %

```

`\normalfootfmt` `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 106), 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; #4 is the note’s series. 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.

```

3096 \notbool{parapparatus@}{\newcommand*{\normalfootfmt}[4]{%
3097   \xstorelineinfo{#1}{#4}%
3098   \nottoggle{Xgroupbyline@#4}{\xledsetnormalparstuff{#4}}{}%
3099   \hangindent=\csuse{Xhangindent@#4}%
3100   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
3101   \nottoggle{Xgroupbyline@#4}{\rule{z@}{\splittopskip}}{}%
3102   {\printlinefootnote{#1}{#4}}%
3103   \print@lemma{#1}{#2}{#4}%
3104   \csuse{Xwrapcontent@#4}{#3}%
3105   \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
3106 }%
3107 %

```

`\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`.

```

3108 \newcommand*{\normalfootstart}[1]{%
3109 %

```

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 XIX p. 238. Here is part of this algorithm, when the block of notes are ready to be printed.

```

3110 \ifdimequal{0pt}{\Xprenotes@}{}%
3111   {%
3112     \iftoggle{Xprenotes@}{%
3113       \togglefalse{Xprenotes@}%

```

```

3114         \skip\csname #1footins\endcsname=%
3115         \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3116     }%
3117 {}%
3118 }%
3119 \vskip\skip\csname #1footins\endcsname%
3120 %

```

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 XVI p. 236 for the general description of the problem.

```

3121 \leftskip0pt \rightskip0pt
3122 \ifl@dpairing\else%
3123     \hsize=\old@hsize%
3124 \fi%
3125 \setXnoteswidthliketwocolumns@{#1}%
3126 \setXnotespositionliketwocolumns@{#1}%
3127 %

```

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

```

3128 \print@Xfootnoterule{#1}%
3129 }%
3130 %

```

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

```

3131 \newcommand*{\normalfootgroup}[1]{%
3132     \csuse{Xbhookgroup@#1}%
3133     \unvbox\csname #1footins\endcsname%
3134     \hsize=\old@hsize%
3135 }%
3136
3137 %

```

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

```

3138 \unless\ifnoledgroup@
3139 \newcommand*{\mpnormalfootgroup}[1]{%
3140     \vskip\skip\@nameuse{mp#1footins}
3141     \ifl@dpairing\ifparledgroup%
3142         \leavevmode\marks\parledgroup@{begin}%
3143         \marks\parledgroup@series{#1}%
3144         \marks\parledgroup@type{Xfootnote}%
3145     \fi\fi\normalcolor%
3146     \ifparledgroup%
3147         \ifl@dpairing%
3148         \else%

```

```

3149 \setXnoteswidthliketwocolumns@{#1}%
3150 \setXnotespositionliketwocolumns@{#1}%
3151 \print@Xfootnoterule{#1}%%
3152 \fi%
3153 \else%
3154 \setXnoteswidthliketwocolumns@{#1}%
3155 \setXnotespositionliketwocolumns@{#1}%
3156 \print@Xfootnoterule{#1}%%
3157 \fi%
3158 \setlength{\parindent}{0pt}
3159 \csuse{Xhookgroup@#1}%
3160 \unvbox\csname mp#1footins\endcsname}}
3161 \fi
3162 %

```

XIII.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.

```

3163 \newcommand*{\Xarrangement@paragraph}[1]{%
3164 \csgdef{series@display#1}{paragraph}
3165 \expandafter\let\csname #1footstart\endcsname=\parafootstart
3166 \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
3167 \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
3168 \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
3169 \count\csname #1footins\endcsname=1000
3170 \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
3171 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
3172 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3173 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3174 \para@footsetup{#1}
3175 %

```

And the extra setup for minipages.

```

3176 \ifnoledgroup@else
3177 \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote

```

```

3178 \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
3179 \count\csname mp#1footins\endcsname=1000
3180 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
3181 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3182 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3183 \fi
3184 }
3185 %

```

\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.

```

3186 \providecommand{\footfudgefiddle}{64}
3187 %

```

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

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

```

3188 \newcommand*{\para@footsetup}[1]{\{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}
3189 \setXnoteswidthliketwocolumns@#1}%
3190 \ifcempty{Xwidth@#1}%
3191 {}%
3192 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
3193 \dimen0=\baselineskip
3194 \multiply\dimen0 by 1024
3195 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
3196 \csxdef{#1footfudgefactor}{%
3197 \expandafter\strip@pt\dimen0 }}
3198 %
3199 %

```

`\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.

```

3200 \newcommand*{\parafootstart}[1]{%
3201   \rightskip=0pt \leftskip=0pt%
3202   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3203   \ifdimequal{0pt}{\Xprenotes@}{}%
3204   {%
3205     \iftoggle{Xprenotes@}{%
3206       \togglefalse{Xprenotes@}%
3207       \skip\csname #1footins\endcsname=%
3208       \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
3209     }%
3210   }%
3211 }%
3212 \vskip\skip\csname #1footins\endcsname%
3213 \setXnoteswidthliketwocolumns@{#1}%
3214 \setXnotespositionliketwocolumns@{#1}%
3215 \print@Xfootnoterule{#1}%
3216 \let\bidir@RTL@everypar\@empty%
3217 \noindent\leavevmode}
3218 %

```

\parafootnote `\parafootnote` 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.³²

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.³³ 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

³²Michael Downes, ‘Line Breaking in `\unhboxed` Text’, *TUGboat* **11** (1990), pp. 605–612.

³³See *The TeXbook*, p. 455 (editions after January 1990).

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.³⁴ 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.³⁵ 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. XIII.6.2 p. 186 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.

```

3219 \newcommand*{\paravfootnote}[2]{%
3220   \csuse{Xbeforeinserting@#1}%
3221   \insert\csname #1footins\endcsname
3222   \bgroup
3223     \csuse{Xnotefontsize@#1}
3224     \footsplitskips
3225     \setbox0=\vbox{\hsize=\maxdimen%
3226       \let\bid@RTL@everypar\@empty%
3227       \insert@Xtxtbeforenotes{#1}%
3228       \noindent\csuse{Xhooknote@#1}%
3229       \csname #1footfmt\endcsname #2{#1}}%
3230     \setbox0=\hbox{\Xunvxh{0}{#1}}%
3231     \dp0=0pt
3232     \ht0=\csname #1footfudgefactor\endcsname\wd0
3233   %

```

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

```

3234   \if@RTL\noindent \leavevmode\fi\box0%
3235   \penalty0
3236 \egroup}
3237 %
3238 %

```

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

³⁴Wayne 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.

³⁵‘Line Breaking’, p. 610.

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

`\mpparavfootnote` This version is for minipages.

```

3239 \newcommand*{\mpparavfootnote}[2]{%
3240   \global\setbox\@nameuse{mp#1footins}\vbox{%
3241     \unvbox\@nameuse{mp#1footins}%
3242     \csuse{Xnotefontsize@#1}
3243     \footplitskips
3244     \setbox0=\vbox{\hsize=\maxdimen%
3245       \let\bidir@RTL@everypar\@empty%
3246       \insert@Xtxtbeforenotes{#1}%
3247       \noindent\color@begingroup%
3248       \csuse{Xhooknote@#1}%
3249       \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
3250     \setbox0=\hbox{\Xunvxh{0}{#1}}%
3251     \dp0=\z@
3252     \ht0=\csname #1footfudgefactor\endcsname\wd0
3253     \box0
3254     \penalty0
3255   }}
3256
3257 %

```

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

```

3258 \newcommand*{\Xunvxh}[2]{%
3259   \setbox0=\vbox{\unvbox#1%
3260     \global\setbox1=\lastbox}%
3261   \unhbox1
3262   \unskip           % remove \rightskip,
3263   \unskip           % remove \parfillskip,
3264   \unpenalty        % remove \penalty of 10000,
3265   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
3266
3267 %

```


`\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).

```

3268 \newcommand*{\parafootfmt}[4]{%
3269   \Xstorelineinfo{#1}{#4}%
3270   \Xinsertparafootsep{#4}%
3271   \ledsetnormalparstuff@common%
3272   \printlinefootnote{#1}{#4}%
3273   \print@lemma{#1}{#2}{#4}%
3274   \csuse{Xwrapcontent@#4}{#3}%
3275   \penalty-10 }
3276 %

```

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.

```

3277 \newcommand*{\parafootgroup}[1]{%
3278   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3279   \unvbox\curname #1footins\endcurname
3280   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3281   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3282   \makehboxofhboxes
3283   \setbox0=\hbox{\unhbox0 \removehboxes}%
3284   \csuse{Xbhookgroup@#1}%
3285   \csuse{Xnotefontsize@#1}%
3286   \unhbox0\par%
3287   \global\hsize=\old@hsize%
3288   }%
3289 %
3290 %

```

`\mpparafootgroup` The minipage version.

```

3291 \newcommand*{\mpparafootgroup}[1]{%
3292   \setXnoteswidthliketwocolumns@{#1}%
3293   \vskip\skip\@nameuse{mp#1footins}
3294   \ifl@dpairing\ifparledgroup%
3295     \leavevmode\marks\parledgroup@{begin}%
3296     \marks\parledgroup@series{#1}%

```

```

3297 \marks\parledgroup@type{Xfootnote}%
3298 \fi\fi\normalcolor
3299 \ifparledgroup%
3300 \ifl@dpairing%
3301 \else%
3302 \setXnoteswidthliketwocolumns@{#1}%
3303 \setXnotespositionliketwocolumns@{#1}%
3304 \print@Xfootnoterule{#1}%
3305 \fi%
3306 \else%
3307 \setXnoteswidthliketwocolumns@{#1}%
3308 \setXnotespositionliketwocolumns@{#1}%
3309 \print@Xfootnoterule{#1}%
3310 \fi%
3311 \unvbox\csname mp#1footins\endcsname
3312 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3313 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3314 \makehboxofhboxes
3315 \setbox0=\hbox{\unhbox0 \removehboxes}%
3316 \csuse{Xbhookgroup@#1}%
3317 \csuse{Xnotefontsize@#1}%
3318 \nottoggle{Xparindent@#1}{\parindent=z@}{}%
3319 \unhbox0\par}}
3320
3321 %

```

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

```

\makehboxofhboxes22 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes23 \loop
3324 \unpenalty
3325 \setbox2=\lastbox
3326 \ifhbox2
3327 \setbox0=\hbox{\box2\unhbox0}%
3328 \repeat}
3329
3330 \newcommand*{\removehboxes}{\setbox0=\lastbox
3331 \ifhbox0{\removehboxes}\unhbox0 \fi}
3332
3333 %

```

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

```

\prevpage@num34 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep35 \ifledRcol@%
3336 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%

```

```

3337     {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3338         {\ifcsequal{prevline#1}{lineinfo@}%
3339             {\ifcseempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3340             {\csuse{Xparafootsep@#1}}}%
3341         }%
3342         {\csuse{Xparafootsep@#1}}%
3343     }%
3344     {}%
3345     \global\csname #1prevpage@numR\endcsname=\page@numR%
3346 \else%
3347     \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
3348         {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3349             {\ifcsequal{prevline#1}{lineinfo@}%
3350                 {\ifcseempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
3351                 {\csuse{Xparafootsep@#1}}}%
3352             }%
3353             {\csuse{Xparafootsep@#1}}%
3354         }%
3355         {}%
3356         \global\csname #1prevpage@num\endcsname=\page@num%
3357     \fi%
3358 }
3359 %

```

XIII.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` `\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`.

```

3360 \newcount\@k \newdimen\@h
3361 \newcommand*{\Xrigidbalance}[3]{%
3362     \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
3363     \rigidbalance{#1}{#2}{#3}%
3364 }%
3365
3366 \newcommand*{\rigidbalanceX}[3]{%
3367     \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
3368     \rigidbalance{#1}{#2}{#3}%

```

```

3369 }%
3370
3371 \newcommand*{\rigidbalance}[3]{%
3372   \setbox0=\box#1 \@k=#2 \@h=#3%
3373   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3374   \valign{##\vfil\cr\dosplits}}}}
3375
3376 \newcommand*{\dosplits}{\ifnum\@k>0 \noalign{\hfil}\splitoff
3377   \global\advance\@k-1\cr\dosplits\fi}
3378
3379 \newcommand*{\splitoff}{\dimen0=\ht0
3380   \divide\dimen0 by\@k \advance\dimen0 by\@h
3381   \setbox2 \vsplit0 to \dimen0
3382   \unvbox2 }
3383
3384 %

```

Three columns

```

\Xarrangement@threecol 3385 \newcommand*{\Xarrangement@threecol}[1]{%
3386   \csgdef{series@display#1}{threecol}
3387   \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3388   \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3389   \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
3390   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3391   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3392   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3393   \threecolfootsetup{#1}
3394   %

```

The additional setup for minipages.

```

3395   \ifnoledgroup@else
3396     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3397     \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3398     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3399     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3400     \mpthreecolfootsetup{#1}
3401   \fi
3402 }
3403
3404 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XIII.6.2 p. 186 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.

```

3405 \newcommand*{\threecolfootsetup}[1]{%
3406   \count\csname #1footins\endcsname 333
3407   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3408   \multiply\dimen\csname #1footins\endcsname \thr@@}
3409   %

```

`\mpthreecolfootsetup` The setup for minipages.

```

3410 \newcommand*{\mpthreecolfootsetup}[1]{%
3411   \count\csname mp#1footins\endcsname 333
3412   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3413   %
3414   %

```

`\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).

```

3415 \notbool{parapparatus@}\newcommand*{\newcommand}\threecolvfootnote}[2]{%
3416   \iftoggle{Xgroupbyline@#1}{%
3417     \prepare@Xgroupbyline{#1}{#2}\threecolvfootnote@inserted}%
3418   }%
3419   {%
3420     \X@beforeinsertion{#1}%
3421     \insert\csname #1footins\endcsname{%
3422       \threecolvfootnote@inserted{#1}{#2}%
3423     }%
3424   }%
3425   %
3426   %

```

```

\threecol footnote@inserted 27 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
threecol footnote@inserted}[2]{%
3428 \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3429 \noindent\csuse{Xhooknote@#1}%
3430 \csuse{Xnotefontsize@#1}%
3431 \footsplitskips%
3432 \csname #1footfmt\endcsname #2{#1}%
3433 }%
3434 %

```

\threecol footfmt \threecol footfmt 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.

```

3435 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecol footfmt}[4]{%
3436 \Xstorelineinfo{#1}{#4}%
3437 \threecol@begin@insert{#4}%
3438 \hspace{\parindent}%
3439 \printlinefootnote{#1}{#4}%
3440 \print@lemma{#1}{#2}{#4}%
3441 \csuse{Xwrapcontent@#4}{#3}%
3442 \nottoggle{Xgroupbyline@#4}%
3443 {\strut\par\allowbreak}%
3444 }%
3445 %
3446 %

```

\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 \threecol footfmt and by \insert@Xtxtbeforenotes.

```

3447 \newcommand{\threecol@begin@insert}[1]{%
3448 \normal@pars%
3449 \nottoggle{Xgroupbyline@#1}%
3450 {\hspace \csuse{Xhsizethreecol@#1}}%
3451 }%
3452 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3453 \tolerance=5000%
3454 \hangindent=\csuse{Xhangindent@#1}%
3455 \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3456 \@tempdima=\parindent%
3457 \csuse{Xcolalign@#1}%
3458 \parindent=\@tempdima%
3459 \strut%
3460 }%
3461 %

```

\threecol footgroup 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.

```

3462 \newcommand*{\threecolfootgroup}[1]{%
3463   \csuse{Xbhookgroup@#1}\par%
3464   \splittopskip=\ht\strutbox
3465   \expandafter
3466   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
3467   %

```

`\mpthreecolfootgroup` The setup for minipages.

```

3468 \newcommand*{\mpthreecolfootgroup}[1]{%
3469   \vskip\skip\@nameuse{mp#1footins}
3470   \ifl@dpairing\ifparledgroup%
3471     \leavevmode\marks\parledgroup@{begin}%
3472     \marks\parledgroup@series{#1}%
3473     \marks\parledgroup@type{Xfootnote}%
3474   \fi\fi\normalcolor
3475   \ifparledgroup%
3476     \ifl@dpairing%
3477     \else%
3478       \setXnoteswidthliketwocolumns@{#1}%
3479       \setXnotespositionliketwocolumns@{#1}%
3480       \print@Xfootnoterule{#1}%
3481     \fi%
3482   \else%
3483     \setXnoteswidthliketwocolumns@{#1}%
3484     \setXnotespositionliketwocolumns@{#1}%
3485     \print@Xfootnoterule{#1}%
3486   \fi%
3487   \csuse{Xbhookgroup@#1}\par%
3488   \splittopskip=\ht\strutbox
3489   \expandafter
3490   \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3491   %
3492   %

```

Two columns

```

\Xarrangement@twocol 93 \newcommand*{\Xarrangement@twocol}[1]{%
3494   \csgdef{series@display#1}{twocol}
3495   \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3496   \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt

```

```

3497 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3498 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3499 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3500 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3501 \twocolfootsetup{#1}
3502 %

```

The additional setup for minipages.

```

3503 \ifnoledgroup@else
3504   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3505   \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
3506   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3507   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3508   \mptwocolfootsetup{#1}
3509 \fi
3510 }
3511
3512 %

```

`\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 `\hspace`.

`\twocolvfootnote`
`\twocolvfootnote@inserted`
`\twocolfootfmt`
`\twocolfootgroup`

```

3513 \newcommand*{\twocolfootsetup}[1]{%
3514   \count\csname #1footins\endcsname 500
3515   \csxdef{default@#1footins}{500}%
3516   \multiply\dimen\csname #1footins\endcsname \tw@}
3517 %

3518 \notbool{parapparatus@}{\newcommand*}{\newcommand*}{\twocolvfootnote}[2]{%
3519   \iftoggle{Xgroupbyline@#1}{%
3520     \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3521   }{%
3522     \X@beforeinsertion{#1}%
3523     \insert\csname #1footins\endcsname{%
3524       \twocolvfootnote@inserted{#1}{#2}%
3525     }%
3526   }%
3527 }%

3528 \notbool{parapparatus@}{\newcommand*}{\newcommand*}{\twocolvfootnote@inserted}[2]{%
3529   \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3530   \noindent\csuse{Xbhooknote@#1}%
3531   \csuse{Xnotefontsize@#1}%
3532   \footplitskips%
3533   \csname #1footfmt\endcsname #2{#1}%
3534 }%
3535 %

```



```

3536 \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolfootfmt}[4]{% 4th
arg is optional, for backward compatibility
3537 \Xstorelineinfo{#1}{#4}%
3538 \twocol@begin@insert{#4}%
3539 \hspace{\parindent}%
3540 \printlinefootnote{#1}{#4}%
3541 \print@lemma{#1}{#2}{#4}%
3542 \csuse{Xwrapcontent@#4}{#3}%
3543 \nottoggle{Xgroupbyline@#4}%
3544   {\strut\par\allowbreak}%
3545   }%
3546 }%
3547 %

3548 \newcommand{\twocol@begin@insert}[1]{%
3549   \normal@pars%
3550   \hsize \csuse{Xhsizetwocol@#1}%
3551   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3552   \tolerance=5000%
3553   \hangindent=\csuse{Xhangindent@#1}%
3554   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3555   \@tempdima=\parindent%
3556   \csuse{Xcolalign@#1}%
3557   \parindent=\@tempdima%
3558   \strut%
3559 }%
3560
3561 \newcommand*{\twocolfootgroup}[1]{%
3562   \csuse{Xhookgroup@#1}\par%
3563   \splittopskip=\ht\strutbox
3564   \expandafter
3565   \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
3566
3567 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3568 \newcommand*{\mptwocolfootsetup}[1]{%
3569   \count\csname mp#1footins\endcsname 500
3570   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3571 %

3572 \newcommand*{\mptwocolfootgroup}[1]{%
3573   \vskip\skip\@nameuse{mp#1footins}
3574   \ifl@dpairing\ifparledgroup%
3575     \leavevmode\marks\parledgroup@{begin}%
3576     \marks\parledgroup@series{#1}%
3577     \marks\parledgroup@type{Xfootnote}%
3578   \fi\fi\normalcolor
3579   \ifparledgroup%

```

```

3580 \ifl@pairing%
3581 \else%
3582   \setXnoteswidthliketwocolumns@{#1}%
3583   \setXnotespositionliketwocolumns@{#1}%
3584   \print@Xfootnoterule{#1}%
3585 \fi%
3586 \else%
3587   \setXnoteswidthliketwocolumns@{#1}%
3588   \setXnotespositionliketwocolumns@{#1}%
3589   \print@Xfootnoterule{#1}%
3590 \fi%
3591 \csuse{Xbhookgroup@#1}\par%
3592 \splittopskip=\ht\strutbox
3593 \expandafter
3594 \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3595
3596 %

```

XIII.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.

XIII.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.

```

3597 \def\endashchar{\textnormal{--}}
3598
3599 \newcommand*{\fullstop}{\textnormal{.}}
3600 \def\Xsublinesep@side{\fullstop}
3601

```

```

3602 \newcommand*{\rbracket}{\textnormal{%
3603   \csuse{text\csuse{footnote@lang}}{%
3604     \ifluatex%
3605       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]{\thinspace
3606     }%
3607     \else%
3608       \thinspace}%
3609     \fi}%
3610 }
3611
3612 %

```

XIII.7.2 Pstart number in footnote

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

```

3613 \newcommand{\printpstart}[0]{%
3614   \ifbool{bool{1@dpairing} or bool{1@dprintingpages} or bool{
3615     1@dprintingcolumns}}{%
3616     \ifledRcol%
3617       \thepstartR%
3618     \else%
3619       \thepstartL%
3620     \fi%
3621   }{%
3622     \thepstart%
3623   }%
3624 }

```

XIII.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).

```

3625 %
3626 \newcommand{\print@lemma}[3]{%
3627   \bgroup%
3628   \nottoggle{Xlemmadisablefontselection@#3}%
3629   {\select@lemmafont#1|}%
3630   }%
3631   \bgroup%
3632   \csuse{Xlemmafont@#3}%Deprecated
3633   \csuse{Xwraplemma@#3}{#2}%
3634   \egroup%
3635   \egroup%
3636   \iftoggle{nosep@}{%
3637     \hskip\csuse{Xinplaceoflemmaseparator@#3}%

```

```

3638 \relax%
3639 }%
3640 {\ifcseempty{Xlemmaseparator@#3}%
3641 {%
3642 \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3643 \relax%
3644 }%
3645 {%
3646 \nobreak%
3647 \hskip\csuse{Xbeforelemmaseparator@#3}%
3648 \csuse{Xlemmaseparator@#3}%
3649 \hskip\csuse{Xafterlemmaseparator@#3}%
3650 \relax%
3651 }%
3652 }%
3653 }%
3654 %

```

XIII.7.4 Line number printing

\Xstorelineinfo The `\Xstorelineinfo` macro is used to store some data about line number of the current critical footnote, data which will be reused later for the `\Xnumberonlyfirstinline` and related setting.

#1 footnote specification for the current footnote ; #2 footnote series.

```

3655 \newcommand{\Xstorelineinfo}[2]{%
3656 \l@dp@rsefootspec#1}%
3657 \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3658 \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3659 \l@dparsedendline - \l@dparsedendsub}%
3660 }%
3661 {%
3662 \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3663 }%
3664 }%

```

\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`.

```

3665 \newcommand{\printlinefootnote}[2]{%
3666 \iftoggle{nonum@}{%Try if the line number must printed for this specific
3667 not (by default, yes)
3668 \hspace{\csuse{Xinplaceofnumber@#2}}}%
3669 }%
3670 {%

```

```

3671 \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
default, yes)
3672 {%
3673 \hspace{\csuse{Xinplaceofnumber@#2}}%
3674 }%
3675 {%
3676 {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
3677 {%
3678 \ifcsdef{prevline#2}%
3679 {%Be sure the \prevline exists.
3680 \ifcsequal{prevline#2}{\lineinfo@}%Try it
3681 {%
3682 \ifcsempy{Xsymlinenum@#2}%
3683 {%
3684 \hspace{\csuse{Xinplaceofnumber@#2}}%
3685 }%
3686 {\printsymlinefootnotearea{#2}}%
3687 }%
3688 {%
3689 \printlinefootnotearea{#1}{#2}%
3690 }%
3691 }%
3692 {%
3693 \printlinefootnotearea{#1}{#2}%
3694 }%
3695 }%
3696 {%
3697 \printlinefootnotearea{#1}{#2}%
3698 }%
3699 \csxdef{prevline#2}{\lineinfo@}%
3700 }%
3701 }%
3702 }%
3703 }%
3704 }
3705 %

```

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

```

3706 \newcommand{\printsymlinefootnotearea}[1]{%
3707 \hspace{\csuse{Xbeforesymlinenum@#1}}%
3708 \csuse{Xnotenumfont@#1}%
3709 \ifdimequal{\csuse{Xboxsymlinenum@#1}}{\z@}%
3710 {\csuse{Xsymlinenum@#1}}%
3711 {\hbox to \csuse{Xboxsymlinenum@#1}%
3712 {\csuse{Xsymlinenum@#1}\hfill}%
3713 }%
3714 \hspace{\csuse{Xaftersymlinenum@#1}}%

```

```

3715 }%
3716 %

```

\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.)

```

3717 \newcommand{\printlinefootnotearea}[2]{%
3718   \printXbeforenumber{#2}%
3719   \csuse{Xnotenumfont@#2}%
3720   \boxfootnotenumbers{#1}{#2}%
3721   \printXafternumber{#2}%
3722 }%
3723 %

```

\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.

```

3724 \newcommand{\boxfootnotenumbers}[2]{%
3725   \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
3726     \printlinefootnotenumbers{#1}{#2}%
3727   }%
3728   {%
3729     \hbox to \csuse{Xboxlinenum@#2}%
3730     {%
3731       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3732       \printlinefootnotenumbers{#1}{#2}%
3733       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3734     }%
3735   }%
3736 }%
3737 %

```

\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.

```

3738 \newcommand{\printlinefootnotenumbers}[2]{%
3739   \xdef\@currentseries{#2}%
3740   \ifboolexpr{%
3741     (togl{Xpstart@#2} and bool{numberpstart})%
3742     or togl{Xpstarteverytime@#2}}%
3743   {\printpstart}{}%
3744   \iftoggle{Xstanza@#2}{%
3745     \ifnumberstanza%
3746       \printstanza%
3747       \csuse{Xstanzaseparator@#2}%

```

```

3748 \fi%
3749 }{}%
3750 \iftoggle{Xonlypstart@#2}{%
3751 \csuse{Xtxtbeforenumber@#2}%
3752 \printlines#1|\ifledRcol@{\Rlineflag\fi}}%
3753 }%
3754 %

```

\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.)

```

3755 \newcommand{\printXbeforenumber}[1]{%
3756 \hspace{\csuse{Xbeforenumber@#1}}%
3757 }%
3758 %

```

\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

```

3759 \newcommand{\printXafternumber}[1]{%
3760 \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3761 \hspace{\csuse{Xafternumber@#1}}%
3762 }%
3763 %

```

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. 106: the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma.

edmac’ creator have defined six boolean in order to know which component of line number description we have to print:

- `\ifl@d@pnum` for page numbers;
- `\ifl@d@ssub` for starting sub-line;
- `\ifl@d@elin` for ending line;
- `\ifl@d@esl` for ending sub-line; and
- `\ifl@d@dash` for the dash between the starting and ending groups.

There is no boolean for the line number because it is always printed.

Maieul Rouquette has added `\ifl@d@Xtwolines` and `\ifl@d@Xmorethantwolines` to print a symbol which stands for “and subsequent” when there are two, three or more lines. Is also defines `\@annot@start@print` and `\@annot@end@print` which define annotations associated with the starting and ending line numbers in critical footnotes.

```

\ifl@d@pnum64 \newif\ifl@d@pnum
\ifl@d@ssub65 \newif\ifl@d@ssub
\ifl@d@elin66 \newif\ifl@d@elin
\ifl@d@esl67 \newif\ifl@d@esl
\ifl@d@dash68 \newif\ifl@d@dash
\ifl@d@Xtwolines69 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines70 \newif\ifl@d@Xmorethantwolines%
\@annot@start@print71 \let\@annot@start@print\relax%
\@annot@end@print72 \let\@annot@end@print\relax%
73 %

```

```

\l@d@parsefootsspec \l@d@parsefootsspec{<spec>}{<lemma>}{<text>} parses a footnote specification. <lemma>
\l@d@p@rsefootsspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@d@p@rsestartpage ber and lemma font specifier in \l@d@nums style format. The real work is done by
\l@d@p@rsestartline \l@d@p@rsefootsspec which defines macros holding the numeric values. In many cases,
\l@d@p@rsestartsub this last command is called directly. Just a reminder of the arguments:
\l@d@p@rsendpage \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@d@p@rsendline \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@d@p@rsendsub74 \newcommand*{\l@d@parsefootsspec}[3]{\l@d@p@rsefootsspec#1|}
75 \def\l@d@p@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
76 \gdef\l@d@p@rsestartpage{#1}%
77 \gdef\l@d@p@rsestartline{#2}%
78 \gdef\l@d@p@rsestartsub{#3}%
79 \gdef\l@d@p@rsendpage{#4}%
80 \gdef\l@d@p@rsendline{#5}%
81 \gdef\l@d@p@rsendsub{#6}%
82 }
83 %

```

Initialise the several number value macros.

```

84 \def\l@d@p@rsestartpage{0}%
85 \def\l@d@p@rsestartline{0}%
86 \def\l@d@p@rsestartsub{0}%
87 \def\l@d@p@rsendpage{0}%
88 \def\l@d@p@rsendline{0}%
89 \def\l@d@p@rsendsub{0}%
90
91 %

```

\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.

```

92 \newcommand*{\setprintlines}[6]{%
93 \let\@annot@start@print\relax%
94 \let\@annot@end@print\relax%
95 \l@d@pnumfalse%
96 \l@d@dashfalse%
97 \l@d@elinfalse%
98 %

```


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

```

3799 \ifbypage@
3800   \ifnum#4=#1 \else
3801     \l@d@pnumtrue
3802     \l@d@dashtrue
3803   \fi
3804 \fi
3805 %

```

We print the ending line number if: (1) we are printing the ending page number, or (2) the ending line number is different from the starting line number. If either of these conditions is true, we also print the annotation linked to the ending line number annotations.

```

3806 \ifboolexpr{%
3807   bool{\l@d@pnum}%
3808   or not test{\ifnumequal{#2}{#5}}%
3809 }{%
3810   \l@d@elintrue%
3811   \l@d@dashtrue%
3812   \unless\ifx\relax\annot@end%
3813     \def\@annot@end@print{%
3814       \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@end}%
3815     }%
3816   \fi%
3817   }{%}%
3818 %

```

We define the starting line number annotation as a merge of the starting annotation and ending annotation if we don't print the ending line number. Otherwise, it is only the starting annotation.

```

3819 \ifl@d@elin%
3820   \def\@annot@start@print{%
3821     \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
3822   }%
3823   \else%
3824     \ifx\annot@start\annot@end%
3825       \unless\ifx\@annot@start\relax%
3826         \def\@annot@start@print{%
3827           \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{\
annot@start}%
3828         }%
3829       \fi%
3830     \else%
3831       \ifx\@annot@end@print\relax%
3832         \def\@annot@start@print{%
3833           \l@wrapcs@ifnotemptybox{Xwraplinenumannotation@\@currentseries}{%

```

```

3834         \ifx\annot@start\empty\else%
3835             \annot@start%
3836             \ifdefined\linangesep@%
3837                 \linangesep@%
3838             \else%
3839                 \csuse{Xlinangeseparator@}\@currentseries}%
3840             \fi%
3841         \fi%
3842         \annot@end%
3843     }%
3844 }%
3845 \else%
3846     \let\@annot@start@print\@annot@end@print%
3847     \let\@annot@end@print\relax%
3848 \fi%
3849 \fi%
3850 \fi%
3851 %

```

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

```

3852 \l@d@ssubfalse
3853 \ifnum#3=0 \else
3854     \l@d@ssubtrue
3855 \fi
3856 %

```

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.

```

3857 \l@d@eslfalse
3858 \ifnum#6=0 \else
3859     \ifnum#6=#3
3860         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3861     \else
3862         \l@d@esltrue
3863         \l@d@dashtrue
3864     \fi
3865 \fi%
3866 %

```

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

```

3867 \ifl@d@dash%
3868     \ifbool{pr{togl{fulllines@} or test{\ifcempty{Xtwolines@}\@currentseries}}}%
3869     {%
3870     {%
3871         \setistwofollowinglines{#1}{#2}{#4}{#5}%
3872         \ifbool{pr{
3873             (%

```

```

3874         togl {Xtwolinesbutnotmore@\@currentseries}%
3875         and not%
3876         (%
3877             bool {istwofollowinglines@}%
3878         )%
3879     )%
3880 or%
3881 (%
3882     (not test{\ifnumequal{#1}{#4}})%
3883     and togl{Xtwolinesonlyinsamepage@\@currentseries}%
3884 )%
3885 }%
3886 {}%
3887 {%
3888     \l@d@dashfalse%
3889     \l@d@Xtwolinestrue%
3890     \l@d@elinfalse%
3891     \l@d@eslfalse%
3892     \ifcempty{Xmorethantwolines@\@currentseries}%
3893     {%
3894         {\ifistwofollowinglines\else%
3895             \l@d@Xmorethantwolinestrue%
3896         }%
3897     }%
3898 }%
3899 }%
3900 \fi%
3901 %

```

End of \setprintlines.

```

3902 }%
3903 %

```

\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.

```

3904 \newif\ifistwofollowinglines@%
3905 \newcommand{\setistwofollowinglines}[4]{%
3906   \ifcsdef{lastlinenumberon@#1}%
3907     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
3908     {\numdef{\tmp}{0}}%
3909   \istwofollowinglines@false%
3910   \ifnumequal{#4-#2}{1}%
3911     {\istwofollowinglines@true}%
3912     {\ifbypage@%
3913       \ifnumequal{#3-#1}{1}%
3914       {%
3915         \ifnumequal{#2}{\tmp}%
3916         {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
3917         {}%
3918       }%
3919       {}%
3920     \fi%
3921   }%
3922 }%
3923 %

```

`\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

```

3924 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3925   \begingroup%
3926   %

```

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

```

3927   \ifluatex%
3928     \edef\@tmp{\the\textdir}%
3929     \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
3930     spurious space (bug #397)
3931   \fi%
3932   %

```

Decide which part of line number components we will print.

```

3932   \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3933   %

```

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.

```

3934 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3935   {\bgroup}%
3936   {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
3937 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
3938   {\@annot@start@print}%
3939   {}%
3940 \ifl@d@pnum%
3941   \wrap@edcrossref{\@this@crossref@start}{#1}%
3942   \csuse{Xpagelinesep@\@currentseries}%
3943   \fi%
3944 \wrap@edcrossref{\@this@crossref@start}{%
3945   \linenumrep{#2}%
3946   \iftoggle{Xlineflag@\@currentseries}{#8}{}}%
3947   }%
3948 \ifl@d@ssub%
3949   \csuse{Xsublinesep@\@currentseries}%
3950   \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3951   \fi
3952 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
3953   {\@annot@start@print}%
3954   {}%
3955 \egroup%
3956 %

```

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

```

3957 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3958   {\bgroup}%
3959   {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
3960 \ifl@d@Xtwolines%
3961   \ifl@d@Xmorethantwolines%
3962     \csuse{Xmorethantwolines@\@currentseries}%
3963   \else%
3964     \csuse{Xtwolines@\@currentseries}%
3965   \fi%
3966 \else%
3967   \ifl@d@dash%
3968     \ifdefined\linerangesep%
3969       \linerangesep%
3970     \else%
3971       \csuse{Xlinerangeseparator@\@currentseries}%
3972     \fi%
3973   \fi%
3974 \ifcsstring{Xlinenumannotationposition@\@currentseries}{before}%
3975   {\@annot@end@print}%
3976   {}%
3977 \ifl@d@pnum%
3978   \wrap@edcrossref{\@this@crossref@end}{#4}%
3979   \csuse{Xpagelinesep@\@currentseries}%
3980   \fi%

```

```

3981 \ifl@d@elin%
3982   \wrap@edcrossref{\@this@crossref@end}{%
3983     \linenumrep{#5}%
3984     \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3985   }%
3986 \fi%
3987 \ifl@d@esl%
3988   \ifl@d@elin%
3989     \csuse{Xsublinesep@\@currentseries}%
3990   \fi%
3991   \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3992 \fi%
3993 \ifcsstring{Xlinenumannotationposition@\@currentseries}{after}%
3994   {\@annot@end@print}%
3995 {}%
3996 \fi%
3997 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3998 {}%
3999 {\hfill}%Prevent underfull hbox
4000 \egroup%
4001 \endgroup%
4002 }%
4003 %

```

XIII.7.5 Footnote grouped by line

`\prepare@Xgroupbyline` `\prepare@Xgroupbyline` is a macro called on the `\metaXXXvfootnote` if `\Xgroupbyline` is set to true, instead of calling `\insert` directly.

```

4004 \newcommand{\prepare@Xgroupbyline}[3]{%
4005   \iftoggle{Xgroupbylineseparetwolines@#1}{%
4006     \l@dparsespec#2%
4007     \ifcsdef{#1@forinserting@l@dparseendpage-\l@dparseendline-\l@dparseendsub}%
4008     {%
4009       \csgappto%
4010         {#1@forinserting@l@dparseendpage-\l@dparseendline-\l@dparseendsub}%
4011       {%
4012         \ifcsemt{Xsymlinenum@#1}%
4013         {\csuse{Xparafootsep@#1}}%
4014         {}%
4015         #3{#1}{#2}%
4016         \hskip\csuse{Xafternote@#1}\relax%
4017       }%
4018     }%
4019   }%
4020   \csdef%
4021     {#1@forinserting@l@dparseendpage-\l@dparseendline-\l@dparseendsub}%

```

```

4022     {%
4023       #3{#1}{#2}%
4024       \hskip\csuse{Xafternote@#1}\relax%
4025     }%
4026   }%
4027   \listcsxadd{#1@forinserting}{\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
4028   {%
4029     \ifcsdef{#1@forinserting@all}{%
4030       \csgappto%
4031       {#1@forinserting@all}}%
4032     {%
4033       \ifcseempty{Xsymlinenum@#1}%
4034         {\csuse{Xparafootsep@#1}}%
4035         {}%
4036       #3{#1}{#2}%
4037       \hskip\csuse{Xafternote@#1}\relax%
4038     }%
4039   }%
4040   {%
4041     \csdef%
4042       {#1@forinserting@all}%
4043     {%
4044       #3{#1}{#2}%
4045       \hskip\csuse{Xafternote@#1}\relax%
4046     }%
4047   }%
4048   \listcsgadd{#1@forinserting}{all}%
4049 }%
4050 }%
4051 %

```

XIV Familiar footnotes

XIV.1 Adjacent footnotes

The original edmac provided users with five series of critical footnotes (`\Afootnote` `\Bfootnote` `\Cfootnote` `\Dfootnote` `\Efootnote`), and \LaTeX provides a single numbered footnote. The `reledmac` package uses the edmac mechanism to provide six series of numbered footnotes.

First, though, the `footmisc` package has an option whereby two or more consecutive `\footnotes` have their marks separated by commas. This seemed to Peter Wilson such a useful ability that it was provided automatically by `eledmac`.

Maïeul Rouquette has maintained this feature in `reledmac`, despite he thought that is not directly in relationship with the aim of `reledmac`.

`\multiplefootnotemarker` These macros may have been defined by the memoir class, are provided by the `footmisc` package and perhaps by other footnote packages. That is why we use `\providecommand`

`\multfootsep`

and not `\newcommand`.

```
4052 \providecommand*{\multiplefootnotemarker}{3sp}
4053 \providecommand*{\multfootsep}{\textsuperscript{\normalfont,}}
4054
4055 %
```

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

```
4056 \providecommand*{\m@mmf@prepare}{%
4057   \kern-\multiplefootnotemarker
4058   \kern\multiplefootnotemarker\relax
4059 %}
```

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

```
4060 \providecommand*{\m@mmf@check}{%
4061   \ifdim\lastkern=\multiplefootnotemarker\relax
4062     \edef\x@sf{\the\spacefactor}%
4063     \unkern
4064     \multfootsep
4065     \spacefactor\x@sf\relax
4066   \fi}
4067
4068 %
```

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

```
4069 \@ifclassloaded{memoir}{}{%
4070 %}
```

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

```
4071 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
4072 %
```

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

```
4073
4074 \patchcmd{\@footnotemark}
4075   {\nobreak}
4076   {\m@mmf@check
4077    \nobreak
4078   }
4079   {}{}
4080 \patchcmd{\@footnotemark}
4081   {\makefnmark}
```



```

4082 {\@makefnmark
4083 \m@mmf@prepare
4084 }
4085 {}{}
4086 %

```

Finished the modifications for the non-memoir case.

```

4087 }
4088
4089 %

```

XIV.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.

```

4090 \pretocmd{\@footnotetext}{%
4091   \ifnumberedpar@
4092   \edtext{}{\l@dbfnote{#1}}}%
4093 \else
4094   {}{}
4095 \apptocmd{\@footnotetext}{\fi}{}{}%
4096 %

```

`\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
4097 \patchcmd%
4098   {\footnote}%
4099   {\stepcounter\@mpfn}%
4100   {%
4101   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
4102     l@dprintingcolumns}}{%
4103     \global\advance\footnote@reading by \@ne%
4104     \get@thisfootnote%
4105     \get@fnmark{\thisc@footnote}%
4106     \ifcsdef{footnotereading\the\footnote@reading=typeset}%
4107       {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%
4108       {\setcounter{\@mpfn}{\footnote@reading}}}%
4109     }{%
4110     \stepcounter\@mpfn%
4111     }%
4112   }%

```

```

4113 {}
4114 {}
4115
4116 \newcommand{\get@thisfootnote}{%
4117   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
4118     l@dprintingcolumns}}{%
4119     \protected@xdef\thisc@footnote{\the\footnote@reading}%
4120   }{%
4121     \protected@xdef\thisc@footnote{\the\c@footnote}%
4122   }%
4123
4124 \newcommand{\l@dbfnote}[1]{%
4125   \get@thisfootnote%
4126   \gdef\@tag{#1\relax}%
4127   \ifledRcol%
4128     \xright@appenditem{%
4129       \ifdefined\Hy@footnote@currentHref%
4130         \noexpand\def\noexpand\Hy@footnote@currentHref{
4131           Hy@footnote@currentHref}%
4132       \fi%
4133       \noexpand\vl@dbfnote{\expandonce\@tag}}{\thisc@footnote}%
4134     }%
4135     \to\inserts@listR
4136     \global\advance\insert@countR \@ne%
4137   \else%
4138     \xright@appenditem{%
4139       \ifdefined\Hy@footnote@currentHref%
4140         \noexpand\def\noexpand\Hy@footnote@currentHref{
4141           Hy@footnote@currentHref}%
4142       \fi%
4143       \noexpand\vl@dbfnote{\expandonce\@tag}}{\thisc@footnote}%
4144     }%
4145     \to\inserts@list
4146     \global\advance\insert@count \@ne%
4147   \fi
4148   \ignorespaces%
4149 }%
4150
4151 \newcommand{\get@fnmark}[1]{%
4152   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
4153     l@dprintingcolumns}}{%
4154     {%
4155       \stepcounter{footnote@typeset}%
4156       \setcounter{footnote}{\c@footnote@typeset}%
4157       \immediate\write\@mainaux{%
4158         \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
4159       }%
4160       \def\@thefnmark{\thefootnote}%
4161     }%
4162   }%

```

```

4159 {%
4160   \setcounter{footnote}{#1}%
4161   \def\@thefnmark{\thefootnote}%
4162 }%
4163 }%
4164
4165 \newcommand{\vl@dbfnote}[2]{%
4166   \get@fnmark{#2}%
4167   \@footnotetext{#1}%
4168 }%
4169 %

```

XIV.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`

```

4170 \newcommand*\prebodyfootmark{%
4171   \leavevmode
4172   \ifhmode
4173     \edef\@x@sf{\the\spacefactor}%
4174     \m@mmlf@check
4175     \nobreak
4176   \fi}
4177 \newcommand*\postbodyfootmark{%
4178   \m@mmlf@prepare
4179   \ifhmode\spacefactor\@x@sf\fi\relax}
4180
4181 %

```

XIV.4 Footnote arrangement

XIV.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.

```

4182 \newcommandx{\arrangementX}[2][1,usedefault]{%
4183   \def\do##1{%
4184     \csname arrangementX@#2\endcsname{##1}%
4185   }%
4186   \ifstrepty{#1}%
4187     {%
4188       \dolistloop{\@series}%
4189     }%
4190   {
4191     \docsvlist{#1}%

```

```

4192 }%
4193 }%
4194 %

```

XIV.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.

```

4195 \newcommand*{\normal@footnotemarkX}[1]{%
4196   \prebodyfootmark
4197   \wrapped@bodyfootmarkX{#1}%
4198   \postbodyfootmark}
4199
4200 %

```

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

```

4201 \newcommand*{\normalbodyfootmarkX}[1]{%
4202   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}%
4203 %

```

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

```

4204 \notbool{parapparatus@}{\newcommand*}{\newcommand*}{\normalvfootnoteX}[2]{%
4205   \beforeinsertion@X{#1}%
4206   \insert\@nameuse{footins#1}\bgroup
4207     \reset@font%
4208     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4209     \noindent\csuse{bhooknoteX@#1}%
4210     \csuse{notefontsizeX@#1}%
4211     \footsplitskips
4212     \ifl@dpairing\ifl@dpaging\else%
4213       \setnoteswidthliketwocolumnsX@{#1}%
4214     \fi\fi%
4215     \setnotesXpositionliketwocolumns@{#1}%
4216     \spaceskip=\z@skip \xspaceskip=\z@skip
4217     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
4218
4219 %

```

`\mpnormalvfootnoteX` The minipage version.

```

4220 \newcommand*{\mpnormalvfootnoteX}[2]{%
4221   \get@thisfootnoteX{#1}%
4222   \get@fnmarkX{#1}{\thisc@footnote}%
4223   \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4224   \global\setbox\@nameuse{mpfootins#1}\vbox{%

```

```

4225 \unvbox\@nameuse{mpfootins#1}
4226 \noindent\csuse{bhooknoteX@#1}%
4227 \csuse{notefontsizeX@#1}%
4228 \hsize\columnwidth
4229 \@parboxrestore
4230 \color@begingroup
4231 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
4232 %
4233 %

```

\normalfootfmtX `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

4234 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmtX}[2]{%
4235 \ifluatex%
4236 \texdir\footnote@luatextextdir%
4237 \pardir\footnote@luatexpardir%
4238 \par%
4239 \fi%
4240 \protected@edef\@currentlabel{%
4241 \@nameuse{@thefnmark#1}%
4242 }%
4243 \ledsetnormalparstuffX{#1}%
4244 \hangindent=\csuse{hangindentX@#1}%
4245 \everypar{\hangindent=\csuse{hangindentX@#1}}%
4246 \rule{z@\splittopskip}%
4247 {\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}}%
4248 \csuse{wrapcontentX@#1}{#2}%
4249 \strut\par}}
4250 %
4251 %

```

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

```

4252 \newcommand*{\normalfootfootmarkX}[1]{%
4253 \textsuperscript{\@nameuse{@thefnmark#1}}}
4254 %
4255 %

```

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

```

4256 \newcommand*{\normalfootstartX}[1]{%
4257 \ifdimequal{Opt}{\prenotesX@}{}%
4258 {%
4259 \iftoggle{prenotesX@}{%
4260 \togglefalse{prenotesX@}%
4261 \skip\csname footins#1\endcsname=%
4262 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4263 }%

```

```

4264     {}%
4265   }%
4266   \vskip\skip\csname footins#1\endcsname%
4267   \leftskip=\z@
4268   \rightskip=\z@
4269   \ifl@dpairing\else%
4270     \hsize=\old@hsize%
4271   \fi%
4272   \setnoteswidthliketwocolumnsX@{#1}%
4273   \setnotesXpositionliketwocolumns@{#1}%
4274   \print@footnoteXrule{#1}%
4275 }%
4276
4277 %

```

\normalfootnoteruleX The rule drawn before the footnote series group.

```

4278 \let\normalfootnoteruleX=\footnoterule
4279
4280 %

```

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

```

4281 \newcommand*{\normalfootgroupX}[1]{%
4282   \csuse{bhookgroupX@#1}%
4283   \unvbox\@nameuse{footins#1}%
4284   \hsize=\old@hsize%
4285 }%
4286
4287 %

```

\mpnormalfootgroupX The minipage version.

```

4288 \newcommand*{\mpnormalfootgroupX}[1]{%
4289   \vskip\skip\@nameuse{mpfootins#1}
4290   \ifl@dpairing\ifparledgroup%
4291     \leavevmode\marks\parledgroup@{begin}%
4292     \marks\parledgroup@series{#1}%
4293     \marks\parledgroup@type{footnoteX}%
4294   \fi\fi\normalcolor
4295   \ifparledgroup%
4296     \ifl@dpairing%
4297     \else%
4298       \setnoteswidthliketwocolumnsX@{#1}%
4299       \setnotesXpositionliketwocolumns@{#1}%
4300       \print@footnoteXrule{#1}%
4301     \fi%
4302   \else%
4303     \setnoteswidthliketwocolumnsX@{#1}%

```

```

4304 \setnotesXpositionliketwocolumns@{#1}%
4305 \print@footnoteXrule{#1}%
4306 \fi%
4307 \csuse{bhookgroupX@#1}%
4308 \unvbox\@nameuse{mpfootins#1}}
4309
4310 %

```

```

\normalbfnoteX
4312 \newcommand{\normalbfnoteX}[2]{%
4313 \get@thisfootnoteX{#1}%
4314 \ifledRcol%
4315 \ifluatex
4316 \footnotelang@lua[R]%
4317 \fi
4318 \@ifundefined{xpg@main@language}%if polyglossia
4319 {}%
4320 {\footnotelang@poly[R]}%
4321 \xright@appenditem{%
4322 \noexpand\led@set@index@fornote{#1}%
4323 \noexpand\prepare@edindex@fornote{\led@nums}%
4324 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4325 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
4326 \noexpand\led@reinit@index@fornote%
4327 \unexpanded{\advance\@edindex@fornote@m@ne}%
4328 }%
4329 \to\inserts@listR
4330 \global\advance\insert@countR \@ne%
4331 \else%
4332 \ifluatex
4333 \footnotelang@lua%
4334 \fi
4335 \@ifundefined{xpg@main@language}%if polyglossia
4336 {}%
4337 {\footnotelang@poly}%
4338 \xright@appenditem{%
4339 \noexpand\led@set@index@fornote{#1}%
4340 \noexpand\prepare@edindex@fornote{\led@nums}%
4341 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4342 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
4343 \noexpand\led@reinit@index@fornote%
4344 \unexpanded{\advance\@edindex@fornote@m@ne}%
4345 }%
4346 \to\inserts@list
4347 \global\advance\insert@count \@ne%
4348 \fi
4349 \ignorespaces}

```

```
4350
4351 %
```

\get@thisfootnoteX The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```
4352 \newcommand{\get@thisfootnoteX}[1]{%
4353   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4354     \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
endcsname}%
4355   }{%
4356     \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}%
4357   }%
4358 }%
4359 %
```

\vbfnoteX This command calls the correct footnote-inserting commands.

```
4360 \newcommand{\vbfnoteX}[3]{%
4361   \get@fnmarkX{#1}{#3}%
4362   \@nameuse{regvfootnote#1}{#1}{#2}%
4363 }%
4364
4365 %
```

\get@fnmarkX This command gets the correct footnote number when typesetting parallel texts.

```
4366 \newcommand{\get@fnmarkX}[2]{%
4367   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4368     {%
4369       \stepcounter{footnote#1@typeset}%
4370       \setcounter{footnote#1}{\value{footnote#1@typeset}}%
4371       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4372       \immediate\write\@mainaux{%
4373         \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
@typeset\endcsname}%
4374       }%
4375     }%
4376     {%
4377       \setcounter{footnote#1}{#2}%
4378       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4379     }%
4380   }
4381   %
4382   %
```



```

\newcommand{\vnumfootnoteX}[2]{%
  \ifnumberedpar@
    \edtext{}{\normalbfnoteX{#1}{#2}}%
  \else
    \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
    \get@thisfootnoteX{#1}%
    \get@fnmarkX{#1}{\expandonce\thisc@footnote}%
    \@nameuse{regvfootnote#1}{#1}{#2}%
  \fi}
%
```

arrangementX@normal `\arrangementX@normal{<series>}` initialises the settings for the <series> footnotes. This should always be called for each series.

```

\newcommand*{\arrangementX@normal}[1]{%
  \csgdef{series@displayX#1}{normal}
  \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
  \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
  \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
  \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
  \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
  \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
  \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
  \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
  \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
  \count\csname footins#1\endcsname=1000
  \csxdef{default@footins#1}{1000}%Use to have note only for one side
  \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
  \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
  \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
}%
%
```

Additions for minipages.

```

\ifnoledgroup@%
  \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
  \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
  \count\csname mpfootins#1\endcsname=1000
  \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
  \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
  \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
\fi
}
%
```

XIV.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 22 \newcommand*{\arrangementX@twocol}[1]{%
4423 \csgdef{series@displayX#1}{twocol}
4424 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4425 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4426 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4427 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4428 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4429 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4430 \twocolfootsetupX{#1}
4431 \ifnoledgroup@else%
4432 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4433 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4434 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4435 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4436 \mptwocolfootsetupX{#1}
4437 \fi%
4438 }
4439
4440 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
4441 \newcommand*{\twocolfootsetupX}[1]{%
4442 \count\csname footins#1\endcsname 500
4443 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
4444 \multiply\dimen\csname footins#1\endcsname by \tw@}
4445 \newcommand*{\mptwocolfootsetupX}[1]{%
4446 \count\csname mpfootins#1\endcsname 500
4447 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
4448
4449 %

```

```

\twocolvfootnoteX \twocolvfootnoteX{<series>}
4450 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
4451 \beforeinsertionX{#1}%
4452 \insert\csname footins#1\endcsname\bgroup%
4453 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4454 \noindent\csuse{bhooknoteX@#1}%
4455 \csuse{notefontsizeX@#1}%
4456 \footsplitskips%
4457 \spaceskip=\z@skip \xspaceskip=\z@skip%
4458 \@nameuse{footfmt#1}{#1}{#2}\egroup}
4459
4460 %

```

```

\twocolfootfmtX \twocolfootfmtX{<series>}

```

```

4461 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4462   \protected@edef\@currentlabel{%
4463     \@nameuse{@thefnmark#1}%
4464   }%
4465   \normal@pars%
4466   \hangindent=\csuse{hangindentX@#1}%
4467   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4468   \hsize \csuse{hsizeX@#1}%
4469   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4470   \tolerance=5000\relax%
4471   \par%
4472   \@tempdima=\parindent%
4473   \csuse{colalignX@#1}%
4474   \parindent=\@tempdima%
4475   {\hspace{\parindent}}%
4476   \csuse{notenumpfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4477   \csuse{wrapcontentX@#1}{#2}%
4478   \strut\par}%
4479 \allowbreak%
4480 }%
4481
4482 %

```

```

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX
4483 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notenumpfontsizeX@#1}
4484   \splittopskip=\ht\strutbox
4485   \expandafter
4486   \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4487
4488 \newcommand*{\mptwocolfootgroupX}[1]{%
4489   \vskip\skip\@nameuse{mpfootins#1}
4490   \ifl@dpairing\ifparledgroup%
4491     \leavevmode\marks\parledgroup@{begin}%
4492     \marks\parledgroup@series{#1}%
4493     \marks\parledgroup@type{footnoteX}%
4494     \fi\fi\normalcolor
4495     \ifparledgroup%
4496       \ifl@dpairing%
4497       \else%
4498         \setnoteswidthliketwocolumnsX@{#1}%
4499         \setnotesXpositionliketwocolumns@{#1}%
4500         \print@footnoteXrule{#1}%
4501       \fi%
4502     \else%
4503       \setnoteswidthliketwocolumnsX@{#1}%
4504       \setnotesXpositionliketwocolumns@{#1}%
4505       \print@footnoteXrule{#1}%
4506     \fi%

```

```

4507 \csuse{bhookgroupX@#1}%
4508 \splittopskip=\ht\strutbox
4509 \expandafter
4510 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4511
4512 %

```

XIV.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 13 \newcommand*\arrangementX@threecol}[1]{%
4514 \csgdef{series@displayX#1}{threecol}
4515 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4516 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4517 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4518 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4519 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4520 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4521 \threecolfootsetupX{#1}
4522 \ifnoledgroup@else%
4523 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4524 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4525 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4526 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4527 \mpthreecolfootsetupX{#1}
4528 \fi%
4529 }
4530
4531 %

```

```

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX
4532 \newcommand*\threecolfootsetupX}[1]{%
4533 \count\csname footins#1\endcsname 333
4534 \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4535 \multiply\dimen\csname footins#1\endcsname by \thr@@}
4536 \newcommand*\mpthreecolfootsetupX}[1]{%
4537 \count\csname mpfootins#1\endcsname 333
4538 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
4539
4540 %

```

```

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
4541 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%

```

```

4542 \beforeinsertion@X{#1}%
4543 \insert\csname footins#1\endcsname\bgroup%
4544   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4545   \noindent\csuse{bhooknoteX@#1}%
4546   \csuse{notefontsizeX@#1}%
4547   \footsplitskips%
4548   \@nameuse{footfmt#1}{#1}{#2}\egroup}
4549
4550 %

```

`\threecolfootfmtX` `\threecolfootfmtX{<series>}`

```

4551 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4552   \protected@edef\@currentlabel{%
4553     \@nameuse{@thefnmark#1}%
4554   }%
4555   \hangindent=\csuse{hangindentX@#1}%
4556   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4557   \normal@pars%
4558   \hsize \csuse{hsizethreecolX@#1}%
4559   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4560   \tolerance=5000\relax%
4561   \@tempdima=\parindent%
4562   \csuse{colalignX@#1}%
4563   \parindent=\@tempdima%
4564   {\hspace{\parindent}}%
4565   \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4566   \csuse{wrapcontentX@#1}{#2}%
4567   \strut\par}\allowbreak}
4568
4569 %

```

`\threecolfootgroupX` `\threecolfootgroupX{<series>}`
`\mpthreecolfootgroupX`

```

4570 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4571   \splittopskip=\ht\strutbox
4572   \expandafter
4573   \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4574
4575 \newcommand*{\mpthreecolfootgroupX}[1]{%
4576   \vskip\skip\@nameuse{mpfootins#1}
4577   \ifl@dpairing\ifparledgroup
4578     \leavevmode\marks\parledgroup@{begin}%
4579     \marks\parledgroup@series{#1}%
4580     \marks\parledgroup@type{footnoteX}%
4581   \fi\fi\normalcolor
4582   \ifparledgroup%
4583     \ifl@dpairing%
4584     \else%

```

```

4585 \setnoteswidthliketwocolumnsX@{#1}%
4586 \setnotesXpositionliketwocolumns@{#1}%
4587 \print@footnoteXrule{#1}%
4588 \fi%
4589 \else%
4590 \setnoteswidthliketwocolumnsX@{#1}%
4591 \setnotesXpositionliketwocolumns@{#1}%
4592 \print@footnoteXrule{#1}%
4593 \fi%
4594 \csuse{bhookgroupX@#1}%
4595 \splittopskip=\ht\strutbox
4596 \expandafter
4597 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4598
4599 %

```

XIV.4.5 Paraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@threecol` `\footparagraphX{<series>}`

```

4600 \newcommand*{\arrangementX@paragraph}[1]{%
4601 \csgdef{series@displayX#1}{paragraph}%
4602 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4603 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4604 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4605 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4606 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4607 \count\csname footins#1\endcsname=1000
4608 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
4609 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4610 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4611 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4612 \para@footsetupX{#1}
4613 \ifnoledgroup@else
4614 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4615 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4616 \count\csname mpfootins#1\endcsname=1000
4617 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4618 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4619 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4620 \fi
4621 }
4622
4623 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

4624 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4625 \setnoteswidthliketwocolumnsX@{#1}%
4626 \ifcsempy{widthX@#1}%
4627 {}%
4628 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4629 \dimen0=\baselineskip
4630 \multiply\dimen0 by 1024
4631 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
4632 %
4633 \expandafter
4634 \xdef\csname footfudgefactor#1\endcsname{%
\expandafter\strip@pt\dimen0 }}
4635 %
4636 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

4637 \newcommand*{\parafootstartX}[1]{%
4638 \ifdimequal{0pt}{\prenotesX@}{}%
4639 {%
4640 \iftoggle{prenotesX@}{%
4641 \togglefalse{prenotesX@}%
4642 \skip\csname footins#1\endcsname=%
4643 \glueexpr\csuse{prenotesX@}+\csuse{afterterruleX@#1}\relax%
4644 }%
4645 }%
4646 }%
4647 \leftskip=\z@
4648 \rightskip=\z@
4649 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4650 \vskip\skip\@nameuse{footins#1}%
4651 \setnoteswidthliketwocolumnsX@{#1}%
4652 \setnotesXpositionliketwocolumnsX@{#1}%
4653 \print@footnoteXrule{#1}%
4654 }
4655 %
4656 %

```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`
`\mppara@vfootnoteX`

```

4657 \newcommand*{\para@vfootnoteX}[2]{%
4658 \csuse{beforeinsertingX@#1}%
4659 \insert\csname footins#1\endcsname%
4660 \bgroup
4661 \csuse{notefontsizeX@#1}
4662 \footsplitskips
4663 \setbox0=\vbox{\hsize=\maxdimen%
4664 \let\bidir@RTL@everypar\@empty%
4665 \insert@txtbeforenotesX{#1}%

```

```

4666 \noindent\csuse{bhooknoteX@#1}%
4667 \@nameuse{footfmt#1}{#1}{#2}}%
4668 \setbox0=\hbox{\unvXH{0}{#1}}%
4669 \dp0=\z@
4670 \ht0=\csname footfudgefactor#1\endcsname\wd0
4671 \box0
4672 \penalty0
4673 \egroup}
4674 \newcommand*\mppara@vfootnoteX}[2]{%
4675 \get@thisfootnoteX{#1}%
4676 \get@fnmarkX{#1}{\thisc@footnote}%
4677 \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4678 \global\setbox\@nameuse{mpfootins#1}\vbox{%
4679 \unvbox\@nameuse{mpfootins#1}
4680 \csuse{notefontsizeX@#1}
4681 \footsplitskips
4682 \setbox0=\vbox{\hsize=\maxdimen%
4683 \let\bidirTL@everypar\@empty%
4684 \noindent\color@begingroup%
4685 \csuse{bhooknoteX@#1}%
4686 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
4687 \setbox0=\hbox{\unvXH{0}{#1}}%
4688 \dp0=\z@
4689 \ht0=\csname footfudgefactor#1\endcsname\wd0
4690 \box0
4691 \penalty0}}
4692
4693 %

```

```

\unvXH694 \newcommand*\unvXH}[2]{% 2th is optional for retro-compatibility
4695 \setbox0=\vbox{\unvbox#1%
4696 \global\setbox1=\lastbox}%
4697 \unhbox1
4698 \unskip % remove \rightskip,
4699 \unskip % remove \parfillskip,
4700 \unpenalty % remove \penalty of 10000,
4701 \hskip\csuse{afternoteX@#2}%
4702 \relax}% but add the glue to go between the notes
4703
4704 %

```

\parafootfmtX \parafootfmtX{<series>}

```

4705 \newcommand*\parafootfmtX}[2]{%
4706 \protected@edef\@currentlabel{%
4707 \@nameuse{@thefnmark#1}%
4708 }%
4709 \insertparafootsepX{#1}%
4710 \ledsetnormalparstuff@common%

```



```

4711 {\csuse{notenumfontX@#1}%
4712 \csuse{notenumfontX@#1}%
4713 \wrapped@footfootmarkX{#1}%
4714 \strut%
4715 \csuse{wrapcontentX@#1}{#2}%
4716 \penalty-10}}
4717
4718 %

```

```

\para@footgroupX \para@footgroupX{<series>}
\mppara@footgroupX
4719 \newcommand*{\para@footgroupX}[1]{%
4720 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4721 \unvbox\csname footins#1\endcsname
4722 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4723 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4724 \makehboxofhboxes
4725 \setbox0=\hbox{\unhbox0 \removehboxes}%
4726 \csuse{bhookgroupX@#1}
4727 \csuse{notefontsizeX@#1}
4728 \unhbox0\par}
4729
4730 \newcommand*{\mppara@footgroupX}[1]{%
4731 \setnoteswidthliketwocolumnsX@{#1}%
4732 \vskip\skip\@nameuse{mpfootins#1}
4733 \ifl@dpairing\ifparledgroup
4734 \leavevmode%
4735 \leavevmode\marks\parledgroup@{begin}%
4736 \marks\parledgroup@series{#1}%
4737 \marks\parledgroup@type{footnoteX}%
4738 \fi\fi\normalcolor
4739 \ifparledgroup%
4740 \ifl@dpairing%
4741 \else%
4742 \setnoteswidthliketwocolumnsX@{#1}%
4743 \setnotesXpositionliketwocolumns@{#1}%
4744 \print@footnoteXrule{#1}%
4745 \fi%
4746 \else%
4747 \setnoteswidthliketwocolumnsX@{#1}%
4748 \setnotesXpositionliketwocolumns@{#1}%
4749 \print@footnoteXrule{#1}%
4750 \fi%
4751 \unvbox\csname mpfootins#1\endcsname
4752 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4753 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4754 \makehboxofhboxes
4755 \setbox0=\hbox{\unhbox0 \removehboxes}%
4756 \csuse{bhookgroupX@#1}%
4757 \csuse{notefontsizeX@#1}%

```

```

4758 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4759 \unhbox0\par}}
4760
4761 %

```

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

```

\insertparafootsepX%2 \newcommand{\insertparafootsepX}[1]{%
4763   \ifledRcol@%
4764     \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
4765       {\csuse{Xparafootsep@#1}}%
4766       {}%
4767     \global\csname prevpage#1@numR\endcsname=\page@numR%
4768   \else%
4769     \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
4770       {\csuse{Xparafootsep@#1}}%
4771       {}%
4772     \global\csname prevpage#1@num\endcsname=\page@num%
4773   \fi%
4774 }
4775 %

```

XIV.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.

```

4776 \newcommand{\wrapped@footfootmarkX}[1]{%
4777   \ifdefined\hypertarget%
4778     \hyperlink%
4779       {@bodyfootmark#1@this@footnoteX@reading}%
4780       {@nameuse{footfootmark#1}}%
4781     \Hy@raisedlink{%
4782       \hypertarget%
4783       {@footnotemark#1@this@footnoteX@reading}%
4784       {}%
4785     }%
4786   \else%
4787     \@nameuse{footfootmark#1}%
4788   \fi%
4789 }%
4790 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package's commands, if needed.

```

4791 \newcommand{\wrapped@bodyfootmarkX}[1]{%
4792   \ifdefined\hypertarget%
4793     \hyperlink%
4794       {@footnotemark#1\expandafter\the\csname footnote#1@reading\
endcsname}%
4795     {\@nameuse{bodyfootmark#1}}%
4796   \Hy@raisedlink{%
4797     \hypertarget%
4798       {@bodyfootmark#1\expandafter\the\csname footnote#1@reading\
endcsname}%
4799     {}%
4800   }%
4801   \else%
4802     \@nameuse{bodyfootmark#1}%
4803   \fi%
4804 }%
4805 %

```

XV Code common to both critical and familiar footnote in normal arrangement

\par should always be redefined to \endgraf within the format macro (this is what \normal@pars does), to override tricky material in the main text to get the lines numbered automatically (as set up by \autopar, for example).

In the case of footnote arranged in a “normal” way, we also must set some setting for paragraph indent and text direction when using Lua[®]TeX.

That why we have defined \ledsetnormalparstuff@common in order to make this setting for both familiar and critical notes. This command is called by command to make specific setting to critical or familiar footnote.

```

\ledsetnormalparstuff@common\newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff\ifluatex%
\ledsetnormalparstuffX\ifdefstring{\footnote@luatextextdir}{TLT}{}%
4809   {\textdir\footnote@luatextextdir}%
4810   \pardir\footnote@luatexpardir%
4811   \fi%
4812   \csuse{\csuse{footnote@dir}}%
4813   \normal@pars%
4814   \parfillskip \z@ \@plus 1fil}%
4815
4816 \newcommand*{\Xledsetnormalparstuff}[1]{%
4817   \ledsetnormalparstuff@common%
4818   \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
4819 }%
4820
4821 \newcommand*{\ledsetnormalparstuffX}[1]{%
4822   \ledsetnormalparstuff@common%

```

```

4823 \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}}%
4824 }%
4825 %

```

XVI 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.

```

4826
4827 \newdimen\old@hsize%
4828 \AtBeginDocument{\old@hsize=\hsize}%
4829
4830 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
4831   \global\let\hsize@fornote=\hsize%
4832   \global\old@hsize=\hsize%
4833   \let\old@columnwidth=\columnwidth%
4834   \iftoggle{Xnoteswidthliketwocolumns@#1}%
4835     {%
4836       \setwidthliketwocolumns%
4837       \global\let\hsize@fornote=\hsize%
4838     }%
4839     {}%
4840   \let\hsize=\hsize@fornote%
4841   \let\columnwidth=\old@columnwidth%
4842 }%
4843
4844 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4845   \global\let\hsize@fornote=\hsize%
4846   \global\old@hsize=\hsize%
4847   \let\old@columnwidth=\columnwidth%
4848   \iftoggle{noteswidthliketwocolumnsX@#1}%
4849     {%
4850       \setwidthliketwocolumns%
4851       \global\let\hsize@fornote=\hsize%
4852     }%
4853     {}%
4854   \let\hsize=\hsize@fornote%
4855   \let\columnwidth=\old@columnwidth%
4856 }%
4857
4858 %

```

`\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`.

```

4859 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
4860   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4861     \csuse{setnotespositionliketwocolumns@\columns@position}%
4862   }{}%
4863 }%
4864
4865 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
4866   \iftoggle{noteswidthliketwocolumnsX@#1}{%
4867     \csuse{setnotespositionliketwocolumns@\columns@position}%
4868   }{}%
4869 }%
4870
4871 %

```

XVII Footnotes' order

`\fnpos` The `\fnpos` and `\mpfnpos` simply place their arguments in `\@fnpos` and `\@mpfnpos`, which will be used later in the output routine.

```

4872 \def\@fnpos{familiar-critical}
4873 \def\@mpfnpos{critical-familiar}
4874 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
4875 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
4876 %

```

XVIII 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@Xfootnoterule77 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoteXrule78   \vskip-\csuse{Xafterterrule@#1}%Because count in \dimen\csuse{#1footins}
4879   \nointerlineskip%
4880   \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
4881   \nointerlineskip%
4882   \vskip\csuse{Xafterterrule@#1}%
4883 }%
4884
4885 \newcommand{\print@footnoteXrule}[1]{%
4886   \vskip-\csuse{afterterruleX@#1}%Because count in \dimen\csuse{footins#1}
4887   \nointerlineskip%
4888   \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%

```

```

4889 \nointerlineskip%
4890 \vskip\csuse{afterruleX@#1}%
4891 }%
4892
4893 %

```

XIX Specific skip for first series of footnotes

XIX.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 !

XIX.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.

```

4894 \newtoggle{Xprenotes@}%
4895 \toggletrue{Xprenotes@}%

```

```

4896 \newcommand{\Xprenotes@}{Opt}%
4897 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
4898 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
4899 %

```

The same, but for familiar footnotes.

```

\Xprenotes@ \newtoggle{prenotesX@}
\Xprenotes@ \toggletrue{prenotesX@}
4902 \newcommand{\prenotesX@}{Opt}
4903 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
4904 %

```

XIX.3 Internal commands

```

firstXseries@ \gdef\firstXseries@{}
prepare@Xprenotes@ \newcommand{\prepare@Xprenotes}[1]{%
4907 \ifdimequal{Opt}{\Xprenotes@}%
4908 {}%
4909 {%
4910 \IfStrEq{\firstXseries@}{}%
4911 \global\skip\csuse{#1footins}=\Xprenotes@%
4912 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4913 \gdef\firstXseries@{#1}%
4914 }%
4915 {%
4916 \ifseriesbefore{#1}{\firstXseries@}%
4917 {%
4918 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
4919 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4920 \gdef\firstXseries@{#1}%
4921 }%
4922 {%
4923 }%
4924 }%
4925 }
4926 %

```

The same thing is required for familiar notes and \prenotesX.

```

firstseriesX@ \gdef\firstseriesX@{}
prepare@prenotesX@ \newcommand{\prepare@prenotesX}[1]{%
4929 \ifdimequal{Opt}{\prenotesX@}%
4930 {}%

```

```

4931 {%
4932   \IfStrEq{\firstseriesX@}{}{%
4933     \global\skip\csuse{footins#1}=\prenotesX@%
4934     \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4935     \gdef\firstseriesX@{#1}%
4936   }%
4937   {%
4938     \ifseriesbefore{#1}{\firstseriesX@}%
4939     {%
4940       \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
4941       \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4942       \gdef\firstXseries@{#1}%
4943     }%
4944     {}%
4945   }%
4946 }%
4947 }
4948 %

```

XX Endnotes

First, check the noend option.

```

4949 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
4950 %

```

XX.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.

```

4951 \newcommand{\l@dend@open}[1]{%
4952   \global\booltrue{l@dend@#1}%
4953   \expandafter\immediate%
4954   \expandafter\openout%
4955   \csname l@d@#1end\endcsname%
4956   =\l@auxdir\jobname.#1end\relax%
4957 }%
4958 \newcommand{\l@dend@close}[1]{%
4959   \global\boolfalse{l@dend@#1}%
4960   \expandafter\immediate%
4961   \expandafter\closeout\csname l@d@#1end\endcsname%
4962 }%

```



```
4963
4964 %
```

\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.

```
4965 \newcommand{\l@dend@stuff}{%
4966   \def\do##1{%
4967     \ifbool{l@dend@##1}{}%
4968     {\l@dend@open{##1}}%
4969     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
4970   }%
4971   \dolistloop{\@series}%
4972 }%
4973
4974 %
```

\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.

```
4975 \global\notbool{parapparatus@}{\long}\def\endprint#1#2#3#4#5#6#7{%
4976   \iftoggle{Xendfirstnote@#4}{%
4977     \global\togglefalse{Xendfirstnote@#4}%
4978     \csuse{Xendtxtbeforenotes@#4}%
4979   }{%
4980     \csuse{Xendbhooknote@#4}%
4981     \csuse{Xendnotefontsize@#4}%
4982     \hangindent=\csuse{Xendhangindent@#4}%
4983     \ifXendinsertsep%
4984       \hskip\csuse{Xendafternote@#4}\relax%
4985     \csuse{Xendsep@#4}%

```

```

4986 \else%
4987   \iftoggle{Xendparagraph@#4}%
4988     {\global\Xendinsertsep@true}%
4989     {}%
4990 \fi%
4991 \xdef\@currentseries{#4}%
4992 \def\do##1{%
4993   \setkeys[mac]{truefootnoteoption}{##1}%
4994 }%
4995 \notblank{#5}{\docsvlist{#5}}{}%
4996 \IfStrEq{#6}{R}{\ledRcol@true}{}%
4997 \def\@this@crossref@start{#7:start}%
4998 \def\@this@crossref@end{#7:end}%
4999 \printlineendnote{#1}{#4}%
5000 \IfStrEq{#6}{R}{\ledRcol@false}{}%
5001 \undef\@this@crossref@start%
5002 \undef\@this@crossref@end%
5003 \nottoggle{Xendlemmadisablefontselection@#4}%
5004   {\select@lemmafont#1|}%
5005   {}%
5006 \bgroup%
5007   \csuse{Xendlemmafont@#4}%
5008   \csuse{Xendwraplemma@#4}{#2}%
5009 \egroup%
5010 \ifboolexpr{%
5011   togl {nosep@}%
5012   or test{\ifcsempy{Xendlemmaseparator@#4}}%
5013 }%
5014   {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
5015   {\nobreak%
5016     \hskip\csuse{Xendbeforelemmaseparator@#4}%
5017     \csuse{Xendlemmaseparator@#4}%
5018     \hskip\csuse{Xendafterlemmaseparator@#4}%
5019     \relax%
5020   }%
5021 \csuse{Xendwrapcontent@#4}{#3}%
5022 \nottoggle{Xendparagraph@#4}{\par}{}%
5023 \def\do##1{%
5024   \setkeys[mac]{falsefootnoteoption}{##1}%
5025 }%
5026 \notblank{#5}{\docsvlist{#5}}{}%
5027 }}%
5028
5029 \let\l@d@section=\@gobble
5030
5031 %

```

`\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.

```

5032 \newcommand{\printlineendnote}[2]{%
5033   \l@dp@rsefootspec#1}%
5034   \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
5035     \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
l@dparsedendsub}%
5036   }%
5037   {%
5038     \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
l@dparsedstartsub}%
5039   }%
5040   \ifboolexpr{%
5041     togl {nonum@}%
5042     or togl {Xendnonumber@#2}%
5043   }%
5044   {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5045   {%
5046     \iftoggle{Xendnumberonlyfirstinline@#2}%
5047     {\ifcsdef{prevendline#2}%
5048       {\ifcsequal{prevendline#2}{\lineinfo@}%
5049         {%
5050           \csuse{Xendbhookinplaceofnumber@#2}%
5051           \ifcsempy{Xendsymnum@#2}%
5052             {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
5053             {\printsymlineendnotearea{#2}}}%
5054           \csuse{Xendahookinplaceofnumber@#2}%
5055         }%
5056         {\printlineendnotearea{#1}{#2}}}%
5057       {\printlineendnotearea{#1}{#2}}}%
5058     }%
5059     {\printlineendnotearea{#1}{#2}}%We keep every time line
5060     \csxdef{prevendline#2}{\lineinfo@}%
5061   }%
5062 }%
5063 %

```

```
\printsymlineendnotearea{
5065 \hspace{\csuse{Xendbeforesymlinenum@#1}}%
5066 \csuse{Xendnotenumfont@#1}%
5067 \ifdimequal{\csuse{Xendboxsymlinenum@#1}}{\z@}%
5068 {\csuse{Xendsymlinenum@#1}}%
5069 {\hbox to \csuse{Xendboxsymlinenum@#1}%
5070 {\csuse{Xendsymlinenum@#1}\hfill}}%
5071 }%
5072 \hspace{\csuse{Xendaftersymlinenum@#1}}%
5073 }%
5074 %
```

`\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.)

```

5075 \newcommand{\printlineendnotearea}[2]{%
5076   \csuse{Xendbhooklinenumber@#2}%
5077   \hspace{\csuse{Xendbeforenumber@#2}}%
5078   \bgroup%
5079   \csuse{Xendnotenumfont@#2}%
5080   \ifdimequal{\csuse{Xendboxlinenum@#2}}{0pt}%
5081     {\printendlines#1||\ifledRcol@\@Rlineflag\fi}%
5082     {\leavevmode%
5083       \hbox to \csuse{Xendboxlinenum@#2}%
5084       {%
5085         \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}}%
5086         \printendlines#1||\ifledRcol@\@Rlineflag\fi%
5087         \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}}%
5088       }%
5089   \egroup%
5090   \hspace{\csuse{Xendafternumber@#2}}%
5091   \csuse{Xendahooklinenumber@#2}%
5092 }%
5093 %

```

XX.2 User level commands

XX.2.1 Inserting contents to endnotes

The `\Xendnotes` commands are defined above, when defining apparatus commands by series. Here, we define only `\toendnotes` command not specific to a series, in order to insert arbitrary code. The regular version writes an unexpanded argument, while the regular version writes a once-expanded argument.

```

\toendnotes94 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes*95 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5096   \def\do##1{%
5097     \expandafter\immediate\expandafter\write\csname l@d##1end\endcsname%
5098     {\unexpanded{#2}\@percentchar}%
5099   }%
5100   \ifstrempy{#1}%
5101     {\dolistloop{\@series}}%
5102     {\docsvlist{#1}}%
5103   }{\led@err@toendnotes@outsidenumbering}%
5104 }%
5105 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
5106   \ifboolexpr{bool{numbering} or bool{numberingR}}{%
5107     \def\do##1{%
5108       \expandafter\immediate\expandafter\write\csname l@d##1end\endcsname%

```

```

5109     {#2\@percentchar}%
5110   }%
5111   \ifstrempy{#1}%
5112     {\dolistloop{\@series}}%
5113     {\docsvlist{#1}}%
5114   }\led@err@toendnotes@outsidenumbering}%
5115 }%
5116 %

```

XX.2.2 Printing endnotes

`\doendnotes` `\doendnotes` is the command you use to print one series of endnotes; it takes one argument: the series letter of the note series you want to print. `\Xendinsertsep@` is set to true at the first note of the series, and to false at the last one.

```

5117 \newif\ifXendinsertsep@%
5118 \newcommand*{\doendnotes}[1]{%
5119   \l@dend@close{#1}%
5120   \begingroup
5121     \csxdef{prevpagenum@#1}{}%
5122     \csxdef{prevpagerange@#1}{}%
5123     \global\toggletrue{Xendfirstnote@#1}%
5124     \makeatletter
5125     \expandafter\let\csname #1end\endcsname=\endprint
5126     \input\l@auxdir\jobname.#1end%
5127     \global\Xendinsertsep@false%
5128   \endgroup}
5129 %

```

`\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.

```

5130 \newcommand*{\doendnotesbysection}[1]{%
5131   \l@dend@close{#1}%
5132   \csxdef{prevpagenum@#1}{}%
5133   \csxdef{prevpagerange@#1}{}%
5134   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
5135   \begingroup%
5136     \makeatletter%
5137     \def\l@d@section##1{%
5138       \global\toggletrue{Xendfirstnote@#1}%
5139       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
5140         {\cslet{#1end}{\endprint}}%
5141         {\cslet{#1end}{\@gobbleseven}}%
5142     }%
5143   \input\l@auxdir\jobname.#1end%

```

```

5144 \global\Xendinsertsep@false%
5145 \endgroup%
5146 }%
5147 %

```

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.

```

5148 }%
5149 %

```

`\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`.

from the starting page number.

```

5150 \newcommand*{\setprintendlines}[6]{%
5151 \let\@annot@start@print\relax%
5152 \let\@annot@end@print\relax%
5153 \l@dpnumfalse%
5154 \l@ddashfalse%
5155 \l@delinfalse%
5156 %

```

First of all, we print the second page number only if the ending page number is different

```

5157 \ifnum#4=#1 \else
5158 \l@dpnumtrue
5159 \l@ddashtrue
5160 \fi
5161 %

```

We print the ending line number if: (1) we are printing the ending page number, or (2) the ending line is different from the starting line number.

```

5162 \ifboolexpr{%
5163   bool{\l@dpnum}%
5164   or not test{\ifnumequal{#2}{#5}}%
5165 }{%
5166   \l@delintrue%
5167   \l@ddashtrue%
5168   \l@delintrue
5169   \l@ddashtrue
5170   \unless\ifx\relax\@annot@end%
5171     \def\@annot@end@print{%
5172       \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
5173 }{\@annot@end}%

```

```

5173     }%
5174     \fi%
5175   }%
5176   {}%
5177 %

```

We define the starting line number annotation as a merge of the starting annotation and ending annotation if we don't print the ending line number. Otherwise, it is only the starting annotation.

```

5178 \ifl@d@elin%
5179   \def\@annot@start@print{%
5180     \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries}{\
annot@start}%
5181   }%
5182   \else%
5183     \ifx\@annot@start\annot@end%
5184       \unless\ifx\@annot@start\relax%
5185         \def\@annot@start@print{%
5186           \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
}{\@annot@start}%
5187         }%
5188       \fi%
5189     \else%
5190       \ifx\@annot@end@print\relax%
5191         \def\@annot@start@print{%
5192           \l@wrapcs@ifnotemptybox{Xendwraplinenumannotation@\@currentseries
}{%
5193             \ifx\@annot@start\empty\else%
5194               \annot@start%
5195               \ifdefined\linangesep%
5196                 \linangesep%
5197               \else%
5198                 \csuse{Xendlinangeseparator@\@currentseries}%
5199               \fi%
5200             \fi%
5201             \annot@end%
5202           }%
5203         }%
5204       \else%
5205         \let\@annot@start@print\@annot@end@print%
5206         \let\@annot@end@print\relax%
5207       \fi%
5208     \fi%
5209   \fi%
5210 %

```

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

```

5211 \l@d@ssubfalse
5212 \ifnum#3=0 \else

```

```

5213 \l@d@ssubtrue
5214 \fi
5215 %

```

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.

```

5216 \l@d@eslfalse
5217 \ifnum#6=0 \else
5218   \ifnum#6=#3
5219     \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
5220   \else
5221     \l@d@esltrue
5222     \l@d@dashtrue
5223   \fi
5224 \fi%
5225 %

```

```

5226 \ifl@d@dash%
5227   \ifboolexpr{togl{fulllines@} or test{\ifcempty{Xendtwolines@}\
@currentseries}}}%
5228   {%
5229   {%
5230     \setistwofollowinglines{#1}{#2}{#4}{#5}%
5231     \ifboolexpr{%
5232       (%
5233         togl {Xendtwolinesbutnotmore@\@currentseries}%
5234         and not%
5235         (%
5236           bool {istwofollowinglines@}%
5237         )%
5238       )%
5239       or%
5240       (%
5241         (not test{\ifnumequal{#1}{#4}})%
5242         and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
5243       )%
5244     }%
5245   }%
5246   {%
5247     \l@d@dashfalse%
5248     \l@d@Xtwolinesttrue%
5249     \l@d@elinfalse%
5250     \l@d@eslfalse%
5251     \ifcempty{Xendmorethantwolines@\@currentseries}%
5252     {%
5253       \ifistwofollowinglines@\else%
5254         \l@d@Xmorethantwolinesttrue%
5255       \fi%
5256     }%
5257   }%

```



```

5258 }%
5259 \fi%
5260 %

```

End of \setprintendlines.

```

5261 }%
5262 %

```

\printendlines Now we are ready to print it all.

```

5263 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
5264   \begingroup
5265   \setprintendlines{#1}-{#2}-{#3}-{#4}-{#5}-{#6}%
5266   %

```

The only subtlety left here is when to print a period between numbers. But the only instance in which this is tricky is for the ending sub-line number: it could be coming after the starting sub-line number (in which case we want only the dash) or after an ending line number (in which case we need to insert a period).

So, first, start the starting line box, if needed.

```

5267   \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
5268   {\bgroup}%
5269   {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}}%
5270   %

```

Then, print the starting page number-

```

5271   \ifboolexpr{%
5272     (%
5273     test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
5274     and not%
5275     (togl{Xendpagenumberonlyfirstifsingle@\@currentseries} and bool{
1@d@pnum}})%
5276   )%
5277   or%
5278   (%
5279   test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
5280   )%
5281   }%
5282   {%
5283   \ifcsempy{Xendsympagenum@\@currentseries}%
5284   {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
5285   {\csuse{Xendsympagenum@\@currentseries}}%
5286   }%
5287   {%
5288   \wrap@edcrossref{\@this@crossref@start}{\printpnum{#1}}%
5289   }%
5290   %

```

Then, determine what must be printed before the start line.

```

5291 \ifl@d@dash%
5292 \ifl@d@pnum%
5293 \csuse{Xendlineprefixsingle@ \@currentseries}%
5294 \else%
5295 \ifcseempty{Xendlineprefixmore@ \@currentseries}%
5296 {\csuse{Xendlineprefixsingle@ \@currentseries}}%
5297 {\csuse{Xendlineprefixmore@ \@currentseries}}%
5298 \fi%
5299 \else%
5300 \csuse{Xendlineprefixsingle@ \@currentseries}%
5301 \fi%
5302 %

```

Then print the starting line, followed, if needed, by the side flag and the starting sub line number, then the line number annotation.

```

5303 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{before}%
5304 {\@annot@start@print}%
5305 {}%
5306 \wrap@edcrossref{\@this@crossref@start}{%
5307 \ifledRcol@%
5308 \linenumrepR{#2}%
5309 \else%
5310 \linenumrep{#2}%
5311 \fi%
5312 }%
5313 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi}{}%
5314 \ifl@d@ssub%
5315 \csuse{Xendsublinesep@ \@currentseries}%
5316 \wrap@edcrossref{\@this@crossref@start}{%
5317 \ifledRcol@%
5318 \sublinenumrepR{#3}%
5319 \else%
5320 \sublinenumrep{#3}%
5321 \fi%
5322 }%
5323 \fi%
5324 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{after}%
5325 {\@annot@start@print}%
5326 {}%
5327 %

```

Close the box.

```

5328 \egroup%
5329 %

```

Open the box for the ending line number.

```

5330 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
5331 {\bgroup}%
5332 {\hbox to \csuse{Xendboxendlinenum@ \@currentseries}\bgroup}%
5333 %

```

Print the dash + the ending line number, or the line number range symbol.

```

5334 \ifl@d@Xtwolines%
5335 \ifl@d@Xmorethantwolines%
5336 \csuse{Xendmorethantwolines@}\@currentseries}%
5337 \else%
5338 \csuse{Xendtwolines@}\@currentseries}%
5339 \fi%
5340 \else%
5341 \ifl@d@dash%
5342 \ifdefined\linrangesep%
5343 \linrangesep%
5344 \else%
5345 \csuse{Xendlinrangeseparator@}\@currentseries}%
5346 \fi%
5347 \fi%
5348 %

```

Print the ending page number.

```

5349 \ifl@d@pnum%
5350 \ifcsstring{prevpagerange@}\@currentseries}{#1-#4}%
5351 {%
5352 \ifcempty{Xendsympagenum@}\@currentseries}%
5353 {\hspace{\csuse{Xendinplaceofpagenumber@}\@currentseries}}}%
5354 {\csuse{Xendsympagenum@}\@currentseries}}}%
5355 }%
5356 {%
5357 \wrap@edcrossref{\@this@crossref@end}\printpnum{#4}%
5358 }%
5359 \fi%
5360 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag, the subline number, and line number annotation.

```

5361 \ifcsstring{Xendlinenumannotationposition@}\@currentseries}{before}%
5362 {\@annot@end@print}%
5363 }%
5364 \ifl@d@elin%
5365 \ifl@d@pnum\csuse{Xendlineprefixsingle@}\@currentseries}\fi%
5366 \wrap@edcrossref{\@this@crossref@end}{%
5367 \ifledRcol%
5368 \linenumrepR{#5}%
5369 \else%
5370 \linenumrep{#5}%
5371 \fi%
5372 }%
5373 \iftoggle{Xendlineflag@}\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
5374 \fi%
5375 \ifl@d@esl%
5376 \ifl@d@elin%

```

```

5377 \csuse{Xendsublinesep@ \@currentseries}%
5378 \fi%
5379 \wrap@edcrossref{\@this@crossref@end}{%
5380 \ifledRcol@%
5381 \sublinenumrepR{#6}%
5382 \else%
5383 \sublinenumrep{#6}%
5384 \fi%
5385 }%
5386 \fi%
5387 \ifcsstring{Xendlinenumannotationposition@ \@currentseries}{after}%
5388 {\@annot@end@print}%
5389 {}%
5390 \fi%
5391 %

```

Close the ending line box.

```

5392 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
5393 {}%
5394 {\hfill}%Prevent underfull hbox
5395 \egroup%
5396 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

5397 \iftoggle{Xendpagenumberonlyfirst@ \@currentseries}%
5398 {\iftoggle{Xendpagenumberonlyfirstintwo@ \@currentseries}%
5399 {\csxdef{prevpagerange@ \@currentseries}{#1-#4}}%
5400 {\csxdef{prevpagenum@ \@currentseries}{#4}}%
5401 }%
5402 {}%
5403 %

```

Now, the end of \printendlines macro.

```

5404 \endgroup%
5405 }%
5406
5407 %

```

\printnpnum A macro to print a page number in an endnote. Should not be override anymore

```

5408 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@ \@currentseries
5409 }#1\csuse{Xendafterpagenumber@ \@currentseries}}
5410 %

```

XXI **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@`.

```

5411 \newcommand{\newseries}[1]{%
5412   \def\do##1{\newseries@{##1}}%
5413   \docsvlist{#1}
5414 }
5415 %

```

\@series The `\series@` macro is an etoolbox list, which contains the name of all series.

```

5416 \newcommand{\@series}{}
5417 %

```

The command `\newseries@\series` creates a new series of the footnote.

```

\newseries@18 \newcommand{\newseries@}[1]{
5419 %

```

XXI.1 Test if series is still existing

```

5420 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
5421 {%
5422 %

```

XXI.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```

5423 \ifdefined\newseries@par%
5424   \newseries@par{#1}%
5425 \fi%
5426 %

```

XXI.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```

5427 \unless\ifnocritical@
5428 %

```

XXI.3.1 Options

```

5429 \newtoggle{Xlineflag@#1}
5430 \newtoggle{Xparindent@#1}
5431 \newtoggle{Xlemmadisablefontselection@#1}
5432 \csgdef{Xwrapcontent@#1}{}%
5433 \csgdef{Xbeforeinserting@#1}{}%
5434 \csgdef{Xhangindent@#1}{Opt}%
5435 \csgdef{Xragged@#1}{}%
5436 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
5437 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
5438 \csgdef{Xcolalign@#1}{\raggedright}%
5439 \csgdef{Xnotenumfont@#1}{\normalfont}%
5440 \csgdef{Xnotefontsize@#1}{\footnotesize}%
5441 \csgdef{Xbhooknote@#1}{}%
5442 \csgdef{Xbhookgroup@#1}{}%
5443
5444 \csgdef{Xboxlinenum@#1}{Opt}%
5445 \csgdef{Xboxlinenumalign@#1}{L}%
5446
5447 \csgdef{Xboxstartlinenum@#1}{Opt}%
5448 \csgdef{Xboxendlinenum@#1}{Opt}%
5449
5450 \csgdef{Xboxsymlinenum@#1}{Opt}%
5451 \newtoggle{Xnumberonlyfirstinline@#1}%
5452 \newtoggle{Xgroupbyline@#1}%
5453 \newtoggle{Xgroupbylineseparatetwolines@#1}%
5454 \newtoggle{Xnumberonlyfirstintwolines@#1}%
5455 \csgdef{Xtwolines@#1}{}%
5456 \csgdef{Xmorethantwolines@#1}{}%
5457 \csgdef{Xsublinesep@#1}{\fullstop}%
5458 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
5459 \newtoggle{Xtwolinesbutnotmore@#1}%
5460 \newtoggle{Xtwolinesonlyinsamepage@#1}%
5461 \newtoggle{Xonlypstart@#1}%
5462 \newtoggle{Xpstarteverytime@#1}%
5463 \newtoggle{Xpstart@#1}%
5464 \newtoggle{Xstanza@#1}%
5465 \csgdef{Xstanzaseparator@#1}{}%
5466 \csgdef{Xsymlinenum@#1}{}%
5467 \newtoggle{Xnonumber@#1}%
5468 \csgdef{Xbeforenumber@#1}{Opt}%
5469 \csgdef{Xtxtbeforenumber@#1}{}%
5470 \csgdef{Xafternumber@#1}{0.5em}%
5471 \newtoggle{Xnonbreakableafternumber@#1}%
5472 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
5473 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
5474 \csgdef{Xinplaceofnumber@#1}{1em}%
5475 \global\cslet{Xlemmaseparator@#1}{\rbracket}%

```

```

5476 \csgdef{Xbeforelemmaseparator@#1}{0em}%
5477 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
5478 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
5479 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}
5480 \csgdef{Xafterrule@#1}{0pt}
5481
5482 \csgdef{Xtxtbeforenotes@#1}{}
5483 \newtoggle{Xtxtbeforenotes@#1@typeset}%Not directly used by user,
but internal
5484 \newtoggle{Xtxtbeforenotesonlyonce@#1}%
5485
5486 \csgdef{Xmaxhnotes@#1}{0.8\vsizel}
5487 \newtoggle{Xnoteswidthliketwocolumns@#1}%
5488 \csgdef{Xparafootsep@#1}{}%
5489 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}
5490 \csgdef{Xlinrangeseparator@#1}{\endashchar}%
5491
5492 \csgdef{Xlemmafont@#1}{}%
5493 \csgdef{Xwraplemma@#1}{}
5494 \csgdef{Xwidth@#1}{\hsizel}
5495 \csgdef{Xwraplinenumannotation@#1}{\textsuperscript}%
5496 %

```

XXI.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

5497 \expandafter\newinsert\csname #1footins\endcsname%
5498 \unless\ifnoledgroup@%
5499 \expandafter\newinsert\csname mp#1footins\endcsname%
5500 \fi%
5501 %

```

XXI.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.

```

5502 \global\newbool{parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand}\csname #1footnote\endcsname[2][]{%
5503 \ifedtext@secondarg@%
5504 \ifledRcol%
5505 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}}%
5506 \else%
5507 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}}%
5508 \fi%
5509 \begingroup%
5510 \newcommand{\content}{##2}%
5511 \ifnumberedpar@%

```

```

5512         \ifledRcol%
5513         \ifluatex%
5514             \footnotelang@lua[R]%
5515         \fi%
5516         \@ifundefined{xpg@main@language}%if polyglossia
5517             {}%
5518             {\footnotelang@poly[R]}%
5519         \footnoteoptions@{R}{#1}{true}%
5520         \xright@appenditem{%
5521             \ifbool{indtl@innote}%
5522                 {\unexpanded{\let\index\nindex}}%
5523                 {}%
5524             \ifbool{indtl@notenumber}%
5525                 {\unexpanded{\let\index\nindex}}%There is no note
...number so
5526             {}%
5527             \noexpand\Xnote@true%
5528             \noexpand\prepare@Xprenotes{#1}%
5529             \noexpand\prepare@edindex@fornote{\l@d@nums}%
5530             \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.
5531             \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
5532             \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5533             \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5534             \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
5535             \noexpand\parse@annot\l@current@annotR|%
5536             \noexpand\csuse{v#1footnote}{#1}%
5537             {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
5538             \noexpand\Xnote@false%
5539             \unexpanded{\advance\@edindex@fornote@m@ne}%
5540             \unexpanded{\undef\@this@crossref@start}%
5541             \unexpanded{\undef\@this@crossref@end}%
5542             \ifbool{indtl@innote}%
5543                 {\unexpanded{\let\index\orig@@index}}%
5544                 {}%
5545             \ifbool{indtl@notenumber}%
5546                 {\unexpanded{\let\index\orig@@index}}%
5547                 {}%
5548             }\to\inserts@listR
5549             \footnoteoptions@{R}{#1}{false}%
5550             \global\advance\insert@countR \@ne%
5551         \else%
5552             \ifluatex%
5553                 \footnotelang@lua%

```



```

5554         \fi%
5555         \@ifundefined{xpg@main@language}%if polyglossia
5556             {}%
5557             {\footnotelang@poly}%
5558         \footnoteoptions@{L}{##1}{true}%
5559         \xright@appenditem{%
5560             \ifbool{indtl@innote}%
5561                 {\unexpanded{\let\index\nindex}}%
5562                 {}%
5563         \ifbool{indtl@notenumber}%
5564             {\unexpanded{\let\index\nindex}}%There is no note
...number so
5565             {}%
5566             \noexpand\Xnote@true%
5567             \noexpand\prepare@Xprenotes{##1}%
5568             \noexpand\prepare@edindex@fornote{\l@d@nums}%
5569             \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.
5570             \ifl@dpairing%
5571             \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
5572             \fi%
5573             \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5574             \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5575             \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
5576             \noexpand\parse@annot\l@current@annot|%
5577             \noexpand\csuse{v#1footnote}%
5578                 {##1}%
5579                 {\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
5580             \unexpanded{\undef\@this@crossref@start}%
5581             \unexpanded{\undef\@this@crossref@end}%
5582             \noexpand\Xnote@false%
5583             \unexpanded{\advance\@edindex@fornote@m@ne}%
5584             \ifbool{indtl@innote}%
5585                 {\unexpanded{\let\index\orig@@index}}%
5586                 {}%
5587             \ifbool{indtl@notenumber}%
5588                 {\unexpanded{\let\index\orig@@index}}%
5589                 {}%
5590             }\to\inserts@list
5591             \global\advance\insert@count \@ne%
5592             \footnoteoptions@{L}{##1}{false}%
5593         \fi
5594     \else
5595         \csuse{v#1footnote}{##1}{0|0|0|0|0|0|0|0}{##1}%

```

```

5596         \fi%
5597         \endgroup%
5598     \else%
5599         \led@err@FootnoteNotInSecondArgEdtext{#1}%
5600     \fi%
5601 \ignorespaces%
5602 }
5603 %

```

Create counter used to determine on which page the previous note was called.

```

5604     \expandafter\newcount\csname #1prevpage@num\endcsname%
5605     \expandafter\newcount\csname #1prevpage@numR\endcsname%
5606 %

```

We need to be able to modify `reledmac`'s footnote macros and restore their

```

5607     \global\csletcs{#1@@footnote}{#1footnote}
5608 %

```

XXI.3.4 Set standard display

```

5609     \Xarrangement@normal{#1}%
5610 %

```

End of for critical footnotes.

```

5611     \fi
5612 %

```

XXI.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```

5613     \unless\ifnofamiliar@
5614 %

```

XXI.4.1 Options

```

5615     \newtoggle{parindentX@#1}
5616     \csgdef{wrapcontentX@#1}{}%
5617     \csgdef{hangindentX@#1}{Opt}%
5618     \csgdef{beforeinsertingX@#1}{}%
5619     \csgdef{raggedX@#1}{}%
5620     \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
5621     \csgdef{hsizethreecolX@#1}{.3 \hsize}%
5622     \csgdef{colalignX@#1}{\raggedright}%
5623     \csgdef{notenumfontX@#1}{\normalfont}%
5624     \csgdef{notefontsizeX@#1}{\footnotesize}%
5625     \csgdef{bhooknoteX@#1}{}%

```

```

5626 \csgdef{bhookgroupX@#1}{}%
5627 \csgdef{afterruleX@#1}{Opt}
5628 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
5629 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
5630 \newtoggle{noteswidthliketwocolumnsX@#1}%
5631 \csgdef{parafootsepX@#1}{}%
5632 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
5633 \csgdef{widthX@#1}{\hsizex}%
5634 \csgdef{txtbeforenotesX@#1}{}%
5635 \newtoggle{txtbeforenotesX@#1@typeset}%Not directly used by user,
but internal
5636 \newtoggle{txtbeforenotesonlyonceX@#1}%
5637 \csgdef{Xlinenumannotationposition@#1}{after}%
5638 % End of for familiar footnotes.
5639 % \subsubsection{Create inserts, needed to add notes in foot}
5640 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
5641 % \begin{macrocode}
5642 \expandafter\newinsert\csname footins#1\endcsname%
5643 \unless\ifnoledgroup%
5644 \expandafter\newinsert\csname mpfootins#1\endcsname%
5645 \fi%
5646 %

```

XXI.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.

```

5647 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
5648 \begingroup%
5649 \prepare@prenotesX{#1}%
5650 \newcommand{\content}{##1}%
5651 %
5652 %

```

If we use the \csquotes package, we reset the quotation level.

```

5653 \ifdefined\csq@qllevel%
5654 \csq@qllevel=0\relax%
5655 \fi%
5656 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read reledpar's handbook about that (V.1.2 p. 53).

```

5657 \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
5658 \ifboolexpr{bool{1@dpairing} or bool{1@dprintingpages} or
bool{1@dprintingcolumns}}{%
5659 \ifcsdef{footnote#1@reading\the\csname footnote#1@reading\
endcsname=typeset}%
5660 {\setcounter{footnote#1}{\csuse{footnote#1@reading\the\
csname footnote#1@reading\endcsname=typeset}}}%

```

```

5661         {\setcounter{footnote#1}{\the\csname footnote#1@reading
\endcsname}}}%
5662         }{%
5663         \stepcounter{footnote#1}%
5664         }%
5665 %

```

We also have to check consistency with `\onlysideX` setting.

```

5666         \ifledRcol%
5667         \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}{}%
5668         \else%
5669         \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}{}%
5670         \fi%
5671 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

5672         \protected@csxdef{the#fnmark#1}{\csuse{thefootnote#1}}%
5673         \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
5674         {\csuse{@footnotemark#1}}%
5675         {}%
5676         \ifluatex%
5677         \xdef\footnote@luatextextdir{\the\textdir}%
5678         \xdef\footnote@luatexpardir{\the\pardir}%
5679         \fi%
5680         \if@ledgroup%
5681         \led@set@index@fornote{#1}%
5682         \fi%
5683         \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
5684         \ifbool{indtl@innote}%
5685         {\let\index\orig@@index}%
5686         {}%
5687         \ifbool{indtl@notenumber}%
5688         {\let\index\orig@@index}%
5689         {}%
5690         \endgroup%
5691     }
5692 %

```

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. 53).

```

5693         \newcounter{footnote#1}
5694         \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
5695         \expandafter\newcount\csname footnote#1@reading\endcsname%
5696 %

```

Create counter used to determine on which page the previous note was called.

```

5697 \expandafter\newcount\csname prevpage#1@num\endcsname%
5698 \expandafter\newcount\csname prevpage#1@numR\endcsname%
5699 %

```

Add `\let\footnoteX\@gobble` to `\no@expands`.

```

5700 \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
5701 %

```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to classical `\footnotemark` and `\footnotetext`.

```

5702 \expandafter\newcommand\csname footnote#1mark\endcsname{%
5703 \begingroup%
5704 \prepare@prenotesX{#1}%
5705 \stepcounter{footnote#1}%
5706 \protected@csxdef{theftnmark#1}{\csuse{thefootnote#1}}%
5707 \csuse{@footnotemark#1}%
5708 \m@mmlf@prepare%
5709 \endgroup%
5710 }%
5711 \expandafter\newcommand\csname footnote#1text\endcsname[1]{%
5712 \begingroup%
5713 \csuse{vfootnote#1}{#1}{\expandonce{##1}}%
5714 \endgroup%
5715 }%
5716 %

```

Do not forget to initialize the series.

```

5717 \arrangementX@normal{#1}%
5718 \fi
5719 %

```

XXI.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

5720 \unless\ifnoend@
5721 %

```

XXI.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`

```

5722 \expandafter\newwrite\csname l@d@#1end\endcsname%
5723 \expandafter\newif\csname ifl@dend@#1\endcsname%
5724 %

```

XXI.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.

```

5725
5726 \global\expandafter\newcommand\csname #1endnote\endcsname[2][1,
usedefault]{%
5727 \bgroup%
5728 \newlinechar='40%
5729 \global\@noneed@Footnotetrue%
5730 \newcommand{\content}{##2}%
5731 \stepcounter{labidx}%
5732 \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
5733 \unexpanded{\def\sw@list@inedtext}%
5734 {\expandafter\unexpanded\expandafter{\sw@inthisedtext}}}%
5735 \@percentchar\space%Explicit space, to add a linebreak in the
output file
5736 \noexpand\parse@annot\l@current@annot|\@percentchar\space%
5737 \expandafter\string\csname #1end\endcsname%
5738 {\ifnumberedpar@l@d@nums\fi}%
5739 {\ifnumberedpar@\expandonce\@tag\fi}%
5740 {\expandonce\content}%
5741 {\#1}%
5742 {\unexpanded{##1}}%
5743 {\ifledRcol R\else L\fi}%
5744 {\theedtext}%
5745 \@percentchar%
5746 }%
5747 \egroup%
5748 \ignorespaces%
5749 }%
5750 %

```

XXI.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

5751
5752 \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
5753 \toendnotes[#1]{##1}%

```

```

5754 }%
5755
5756 \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
endcsname*[1]{%
5757 \toendnotes*[#1]{##1}%
5758 }%
5759
5760 %

```

XXI.5.4 Internal commands

\Xendnote commands called \Xend commands on to the endnote file; these are analogous to the various footfmt commands above, and they take the same arguments. When we process this file, we want to pick out the notes of one series and ignore all the rest. To do that, we equate the end command for the series we want to \endprint, and leave the rest equated to \@gobbleseven, which just skips over its seven arguments.

```

5761
5762 \global\cslet{#1end}{\@gobbleseven}
5763 %

```

We need to store the number of times \doendnotesbysection is called for one series.

```

5764 \global\expandafter\newcount\csname #1end@bysection\endcsname%
5765 %

```

XXI.5.5 The options

```

5766 \csgdef{Xendwraplemma@#1}{%
5767 \csgdef{Xendwrapcontent@#1}{}%
5768 \csgdef{Xendtwolines@#1}{}%
5769 \csgdef{Xendmoreethantwolines@#1}{}%
5770 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
5771 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
5772 \newtoggle{Xendlemmadisablefontselection@#1}%
5773 \csgdef{Xendnotenumfont@#1}{\normalfont}%
5774 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
5775 \csgdef{Xendbhooknote@#1}{}%
5776
5777 \csgdef{Xendsublinesep@#1}{\fullstop}%
5778
5779 \csgdef{Xendbeforenumber@#1}{Opt}
5780 \csgdef{Xendafternumber@#1}{0.5em}
5781
5782 \csgdef{Xendboxlinenum@#1}{Opt}%
5783 \csgdef{Xendboxlinenumalign@#1}{L}%
5784
5785 \csgdef{Xendboxstartlinenum@#1}{Opt}%
5786 \csgdef{Xendboxendlinenum@#1}{Opt}%

```

```

5787 \csgdef{Xendlemmaseparator@#1}{}%
5788 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
5789 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
5790 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
5791
5792
5793 \newtoggle{Xendparagraph@#1}%
5794 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
5795 \csgdef{Xendsep@#1}{}%
5796
5797 \csgdef{Xendinplaceofnumber@#1}{Opt}%
5798 \newtoggle{Xendnonnumber@#1}%
5799
5800 \csgdef{Xendhangindent@#1}{Opt}%
5801 \newtoggle{Xendnumberonlyfirstinline@#1}%
5802 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
5803
5804 \csgdef{Xendbeforesymlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
5805 \csgdef{Xendaftersymlinenum@#1}{\csuse{Xendafternumber@#1}}%
5806 \csgdef{Xendsymlinenum@#1}{}%
5807 \csgdef{Xendboxsymlinenum@#1}{Opt}%
5808
5809 \csgdef{Xendbhooklinenumber@#1}{}%
5810 \csgdef{Xendehooklinenumber@#1}{}%
5811 \csgdef{Xendbhookinplaceofnumber@#1}{}%
5812 \csgdef{Xendehookinplaceofnumber@#1}{}%
5813
5814 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%
5815
5816 \csgdef{Xendbeforepagenumber@#1}{p.}%
5817 \csgdef{Xendafterpagenumber@#1}{) }%
5818 \csgdef{Xendlineprefixsingle@#1}{}%
5819 \csgdef{Xendlineprefixmore@#1}{}%
5820
5821 \newtoggle{Xendlineflag@#1}
5822
5823 \csgdef{Xendlemmafont@#1}{}%
5824
5825 \csgdef{Xendlinenumannotationposition@#1}{after}%
5826 \csgdef{Xendwraplinenumannotation@#1}{\textsuperscript}%
5827
5828 \newtoggle{Xendpagenumberonlyfirst@#1}%
5829 \newtoggle{Xendpagenumberonlyfirstifsingle@#1}%
5830 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
5831 \csgdef{Xendsympagenum@#1}{}%
5832 \csgdef{Xendinplaceofpagenumber@#1}{Opt}%
5833
5834 \csgdef{Xendtxtbeforenotes@#1}{}%
5835 \newtoggle{Xendfirstnote@#1}%Not a hook, but used to apply
Xendtxtbeforenotes

```



```

5836
5837 %
End of endnotes declaration
5838 \fi%
5839 %
Dump series in \@series
5840 \listxadd{\@series}{#1}
5841 }
5842 }% End of \newseries
5843 %

```

XXI.6 Init standards series (A,B,C,D,E)

```

5844 \expandafter\newseries\expandafter{\default@series}
5845 %

```

XXII Setting series display

XXII.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.

```

5846 \newcommand{\seriesatbegin}[1]{%
5847   \StrDel{\@series}{#1}[\@series]%
5848   \edef\@new{%
5849     \listead{\@new}{#1}%
5850     \listead{\@new}{\@series}%
5851     \xdef\@series{\@new}%
5852   }
5853 %

```

\seriesatend And `\seriesatend` moves the series to the end of the list.

```

5854 \newcommand{\seriesatend}[1]{%
5855   \StrDel{\@series}{#1}[\@series]%
5856   \edef\@new{%
5857     \listead{\@new}{\@series}%
5858     \listead{\@new}{#1}%
5859     \xdef\@series{\@new}%
5860   }
5861 %

```

XXII.2 Test series order

\ifseriesbefore `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands `<true>` if `<seriesA>` is printed before `<seriesB>`, expands `<false>` otherwise.

```

5862 \newcommand{\ifseriesbefore}[4]{%
5863   \StrPosition{\@series}{#1}[\@first]%
5864   \StrPosition{\@series}{#2}[\@second]%
5865   \ifnumgreater{\@second}{\@first}{#3}{#4}%
5866 }
5867 %

```

XXII.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries 5868 \newcommand{\@getfirstseries}{%
5869   \ifdefempty{\@series}%
5870     {\xdef\@firstseries{}}%
5871     {\StrChar{\@series}{1}[\@firstseries]}%
5872 }%
5873 %

```

XXII.3 Series setting

XXII.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.

XXII.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`.

```

5874 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
5875   \def\do##1{%
5876     \global\settoggle{#2@##1}{#3}%
5877     \ifstrequal{#4}{critical}{
5878       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
5879     }{
5880       \ifstrequal{#4}{familiar}{
5881         \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
5882       }{
5883       }%
5884     \ifstreempty{#1}{%
5885       \dolistloop{\@series}%
5886       \ifstreempty{#5}{%
5887         \docsvlist{#5}%
5888       }
5889     }%
5890     {%
5891       \docsvlist{#1}%
5892     }%
5893   }
5894   %

```

`\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`.

```

5895 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
5896   \def\do##1{
5897     \csgdef{#2@##1}{#3}
5898     \ifstrequal{#4}{critical}{%
5899       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
5900     }{}
5901     \ifstrequal{#4}{familiar}{%
5902       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
5903     }{}%
5904   }%
5905   \ifstreempty{#1}{%
5906     \dolistloop{\@series}%
5907     \ifstreempty{#5}{}%
5908     \docsvlist{#5}
5909   }
5910 }%
5911 {%
5912   \docsvlist{#1}%
5913 }%
5914 }%
5915 %

```

XXII.3.3 Tools to generate options commands

\newhookcommand@series \newhookcommand@series\command names is a generic command to add new commands for hooks, like \Xhsizetwocol. The first argument is the name of the hook, the second a comma-separated list of pseudo-series where the hook can be used, like appref in the case of \Xtwolines. The second argument is also used to create commands named \<hookname><pseudoserries>, like \Xtwolinesappref.

```

5916 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
5917   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5918     \setcommand@series{##1}{#1}{##2}[][#2]%
5919   }%
5920   \ifstreempty{#2}{}%
5921   \def\do##1{%
5922     \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
5923     [1]{%
5924       \csuse{#1}[##1]{####1}%
5925     }%
5926     \docsvlist{#2}%
5927   }%
5928 }
5929 %

```

\newhooktoggle@series \newhooktoggle@series\command names is a generic command to add new commands for a new toggle hook, like \Xnumberonlyfirstinline. The second argu-

ment is also used to create commands named `\<hookname><pseudoseris>`, like `\Xtwolinesbutnotmoreappref`.

```

5930 \newcommand{\newhooktoggle@series}[2][2,usedefault]{%
5931   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
5932     \settoggle@series{##1}{#1}{##2}[][#2]%
5933   }%
5934   \ifstrempy{#2}{}{%
5935     \def\do##1{%
5936       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
5937         \csuse{#1}{##1}%
5938       }%
5939     }%
5940     \docsvlist{#2}%
5941   }%
5942 }
5943 %

```

\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

```

5944 \newcommand{\newhooktoggle@series@reload}[2]{%
5945   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
5946     \settoggle@series{##1}{#1}{##2}[#2]%
5947   }%
5948 }%
5949 %

```

\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.

```

5950 \newcommand{\newhookcommand@series@reload}[2]{%
5951   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5952     \setcommand@series{##1}{#1}{##2}[#2]%
5953   }%
5954 }
5955 %

```

XXII.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.

```

5956 \unless\ifnocritical@
5957   \newhookcommand@series{Xwrapcontent}%
5958   \newhookcommand@series{Xbeforeinserting}%

```

```

5959 \newhookcommand@series{Xlemmafont}%
5960 \newhookcommand@series{Xwraplemma}%
5961 \newhooktoggle@series{Xparindent}
5962 \newhookcommand@series{Xhangindent}
5963 \newhookcommand@series{Xragged}
5964 \newhookcommand@series{Xhsizetwocol}
5965 \newhookcommand@series{Xhsizethreecol}
5966 \newhookcommand@series{Xcolalign}%
5967 \newhookcommand@series{Xnotenumfont}
5968 \newhookcommand@series{Xbhooknote}
5969 \newhookcommand@series@reload{Xbhookgroup}{critical}
5970 \newhookcommand@series{Xboxsymlinenumber}%
5971 \newhookcommand@series{Xsymlinenumber}
5972 \newhookcommand@series{Xbeforenumber}
5973 \newhookcommand@series{Xtxtbeforenumber}
5974 \newhookcommand@series{Xafternumber}
5975 \newhookcommand@series{Xbeforesymlinenumber}
5976 \newhookcommand@series{Xaftersymlinenumber}
5977 \newhookcommand@series{Xinplaceofnumber}
5978 \newhookcommand@series{Xlemmaseparator}
5979 \newhookcommand@series{Xbeforelemmaseparator}
5980 \newhookcommand@series{Xafterlemmaseparator}
5981 \newhookcommand@series{Xinplaceoflemmaseparator}
5982 \newhookcommand@series{Xtxtbeforenotes}
5983 \newhooktoggle@series{Xtxtbeforenotesonlyonce}%
5984 \newhookcommand@series@reload{Xafterrule}{critical}
5985 \newhooktoggle@series{Xnumberonlyfirstinline}
5986 \newhooktoggle@series{Xnumberonlyfirstintwolines}
5987 \newhooktoggle@series{Xgroupbyline}%
5988 \newhooktoggle@series{Xgroupbylineseparetwolines}%
5989 \newhooktoggle@series{Xnonumber}
5990 \newhooktoggle@series{Xpstart}
5991 \newhooktoggle@series{Xpstarteverytime}%
5992
5993 \newhooktoggle@series{Xstanza}%
5994 \newhookcommand@series{Xstanzaseparator}%
5995
5996 \newhooktoggle@series{Xonlypstart}
5997 \newhooktoggle@series{Xnonbreakableafternumber}
5998 \newhooktoggle@series{Xlemmadisablefontselection}
5999 \newhookcommand@series@reload{Xmaxhnotes}{critical}
6000 \newhookcommand@series@reload{Xbeforenotes}{critical}
6001 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
6002 \newhookcommand@series@reload{Xnotefontsize}{critical}
6003
6004 \newhookcommand@series{Xboxlinenum}%
6005 \newhookcommand@series{Xboxlinenumalign}%
6006
6007 \newhookcommand@series{Xboxstartlinenum}%
6008 \newhookcommand@series{Xboxendlinenum}%

```

```

6009 \newhookcommand@series{Xafternote}%
6010 \newhookcommand@series{Xparafootsep}
6011 \newhookcommand@series@reload{Xwidth}{critical}%
6012
6013 \ifundef{\Xhsize}%
6014 {%
6015   \newcommandx{\Xhsize}[2][1,usedefault]{%
6016     \led@warning@Xhsize@deprecated%
6017     \Xwidth[#1]{#2}%
6018   }%
6019 }%
6020 {}%
6021 \fi
6022 %
6023 %

```

Now, hooks for critical notes which also apply to crossreferencing and line numbering at the sides of the page.

```

6024 \newhooktoggle@series{Xlineflag}[appref,SEref]
6025 \newhookcommand@series{Xtwolines}[appref,SEref]
6026 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
6027 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
6028 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
6029 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
6030 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
6031 \newhookcommand@series{Xlinerrangeseparator}[appref,SEref]
6032 \newhookcommand@series{Xlinenumannotationposition}[side,appref,SEref]%
6033 \newhookcommand@series{Xwraplinenumannotation}[side,ref,appref,SEref]%
6034 %

```

XXII.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

6035 \unless\ifnofamiliar@
6036   \newhookcommand@series{wrapcontentX}%
6037   \newhookcommand@series{beforeinsertingX}%
6038   \newhooktoggle@series{parindentX}
6039   \newhookcommand@series{hangindentX}
6040   \newhookcommand@series{raggedX}
6041   \newhookcommand@series{hsizetwocolX}
6042   \newhookcommand@series{hsizethreecolX}
6043   \newhookcommand@series{colalignX}%
6044   \newhookcommand@series{notenumfontX}
6045   \newhookcommand@series{bhooknoteX}
6046   \newhookcommand@series@reload{bhookgroupX}{familiar}
6047   \newhookcommand@series@reload{beforenotesX}{familiar}
6048   \newhookcommand@series@reload{maxhnotesX}{familiar}

```

```

6049 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
6050 \newhookcommand@series@reload{afterruleX}{familiar}
6051 \newhookcommand@series@reload{notefontsizeX}{familiar}
6052 \newhookcommand@series{afternoteX}
6053 \newhookcommand@series{parafootsepX}
6054 \newhookcommand@series{txtbeforenotesX}%
6055 \newhooktoggle@series{txtbeforenotesonlyonceX}%
6056 \newhookcommand@series@reload{widthX}{familiar}%
6057 \ifundef{\hsizeX}%
6058 {%
6059   \newcommandx{\hsizeX}[2][1,usedefault]{%
6060     \led@warning@hsizeX@deprecated%
6061     \widthX[#1]{#2}%
6062   }%
6063 }%
6064 {}%
6065 \fi
6066 %

```

XXII.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.

```

6067 \unless\ifnoend@
6068   \newhookcommand@series{Xendwraplemma}
6069   \newhookcommand@series{Xendwrapcontent}
6070   \newhookcommand@series{Xendnotenumfont}
6071   \newhookcommand@series{Xendlemmafont}%
6072   \newhookcommand@series{Xendbhooknote}
6073
6074   \newhookcommand@series{Xendboxlinenum}%
6075   \newhookcommand@series{Xendboxlinenumalign}%
6076
6077   \newhookcommand@series{Xendboxstartlinenum}%
6078   \newhookcommand@series{Xendboxendlinenum}%
6079
6080   \newhookcommand@series{Xendnotefontsize}
6081   \newhooktoggle@series{Xendlemmadisablefontselection}
6082   \newhookcommand@series{Xendlemmaseparator}
6083   \newhookcommand@series{Xendbeforelemmaseparator}
6084   \newhookcommand@series{Xendafterlemmaseparator}
6085   \newhookcommand@series{Xendinplaceoflemmaseparator}
6086
6087   \newhookcommand@series{Xendbeforenumber}%
6088   \newhookcommand@series{Xendafternumber}%
6089
6090   \newhooktoggle@series{Xendparagraph}
6091   \newhookcommand@series{Xendafternote}
6092   \newhookcommand@series{Xendsep}

```



```

6093 \newhookcommand@series{Xendinplaceofnumber}%
6094 \newhooktoggle@series{Xendnonumber}%
6095
6096
6097 \newhooktoggle@series{Xendnumberonlyfirstinline}%
6098 \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
6099
6100 \newhookcommand@series{Xendsymlinenum}%
6101 \newhookcommand@series{Xendbeforesymlinenum}%
6102 \newhookcommand@series{Xendaftersymlinenum}%
6103 \newhookcommand@series{Xendboxsymlinenum}%
6104
6105 \newhookcommand@series{Xendbhooklinenumber}%
6106 \newhookcommand@series{Xendahooklinenumber}%
6107 \newhookcommand@series{Xendbhookinplaceofnumber}%
6108 \newhookcommand@series{Xendahookinplaceofnumber}%
6109
6110 \newhookcommand@series{Xendhangindent}%
6111
6112 \newhooktoggle@series{Xendpagenumberonlyfirst}%
6113 \newhooktoggle@series{Xendpagenumberonlyfirstifsingle}%
6114 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
6115 \newhookcommand@series{Xendsympagenum}%
6116 \newhookcommand@series{Xendinplaceofpagenumber}%
6117
6118 \newhookcommand@series{Xendtxtbeforenotes}%
6119
6120 \fi
6121 \newhooktoggle@series{Xendlineflag}[apprefwithpage,Serefwithpage]
6122 \newhookcommand@series{Xendtwolines}[apprefwithpage,Serefwithpage]
6123 \newhookcommand@series{Xendmorethantwolines}[apprefwithpage,Serefwithpage]
6124 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,Serefwithpage]
6125 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
6126 Serefwithpage]
6127 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,Serefwithpage
6128 ]
6129 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,Serefwithpage,
6130 Serefonlypage]
6131 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,Serefwithpage]
6132 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,Serefwithpage]
6133 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,Serefwithpage]
6134 \newhookcommand@series{Xendsublinesep}[apprefwithpage,Serefwithpage]
6135
6136 \newhookcommand@series{Xendlinenumannotationposition}[apprefwithpage,
6137 Serefwithpage]%
6138 \newhookcommand@series{Xendwraplinenumannotation}[apprefwithpage,
6139 Serefwithpage]%
6140 %

```

XXII.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

6136 \newcommand{\newhooktoggle@specific}[1]{%
6137   \newtoggle{#1}%
6138   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
6139   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false}
}}
6140 }
6141 %

```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

6142 \newcommand{\newhookarg@specific}[1]{%
6143   \define@key[mac]{truefootnoteoption}{#1}{\csgdef{#1}{##1}}%When enabling
footnote option
6144   \define@key[mac]{falsefootnoteoption}{#1}{\global\csundef{#1}}%When
disabling footnote option
6145 }
6146 %

```

And now, we define some hooks specific to a note.

```

6147 \newhooktoggle@specific{fulllines}%
6148 \newhooktoggle@specific{nonum}
6149 \newhooktoggle@specific{nosep}
6150 \newhookarg@specific{linrangesep}
6151 %

```

`linrangesep@` `\linrangesep@` is defined by the option `linrangesep` of critical notes to change temporarily the line range separator for a specific line. As we have to define it before typesetting the line and undefine it after, we use the family of `xkeyval` package's key.

```

6152 %

```

`\nomk@` `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```

6153 \newtoggle{nomk}%
6154 %

```

XXII.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```

6155 \newcommandx*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]}
6156 %

```

XXIII Output routine

Now we begin the output routine and associated things.

XXIII.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.

```

6157 \newcommand*{\l@ddoxtrafeet}{%
6158   \IfStrEq{familiar-critical}{\@fnpos}
6159   {\do@feetX\do@Xfeet}%
6160   {%
6161     \IfStrEq{critical-familiar}{\@fnpos}%
6162     {\do@Xfeet\do@feetX}%
6163     {%
6164       \setbox\@outputbox \vbox{%
6165         \unvbox\@outputbox%
6166         \do@feet@custom@order}{\@fnpos}%
6167       }%
6168     }%
6169   }%
6170 }%
6171 %
6172 %

```

`\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.

```

6173 \newcommand{\do@feet@custom@order}[2]{%
6174   \def\do##1{%
6175     \edef\@notesseries{\@firstoftwo##1}%
6176     \edef\@notetype{\@secondoftwo##1}%
6177     \ifdefstring{\@notetype}{critical}%
6178       {\csuse{#1append@Xnotes}{\@notesseries}}%
6179       {\ifdefstring{\@notetype}{familiar}%
6180        {\csuse{#1append@notesX}{\@notesseries}}%
6181        {}}%
6182     }%
6183   }%
6184   \expandafter\docsvlist\expandafter{#2}%
6185 }%

```

```
6186 %
```

\do@Xfeet \do@Xfeet is the code extending \@makecol to cater to the extra critical feet.

```
6187 \newcommand*{\do@Xfeet}{%
6188   \setbox\@outputbox \vbox{%
6189     \unvbox\@outputbox
6190     \opXfeet}}
6191 %
```

\@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

```
6192 \newcommand{\append@Xnotes}[1]{%
6193   \ifvoid\csuse{#1footins}\else%
6194     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
6195     \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
6196     \print@Xnotes{#1}%
6197   \fi%
6198 }%
6199 %
```

The normal way to add one series, \print@Xnotes, is replaced by reledpar when using \Pages.

```
6200 \newcommand\print@Xnotes[1]{%
6201   \xdef\@currentseries{#1}%
6202   \csuse{#1footstart}{#1}%
6203   \csuse{#1footgroup}{#1}%
6204 }%
6205 %
```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```
6206 \newcommand*{\@opXfeet}{%
6207   \unless\ifnocritical@%
6208     \gdef\firstXseries@{}%
6209     \def\do##1{%
6210       \append@Xnotes{##1}%
6211     }%
6212     \dolistloop{\@series}%
6213   \fi%
6214 }%
6215 %
```

\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.

```
6216 \newcommand*{\l@ddodoreinextrafeet}{%
6217   \IfStrEq{familiar-critical}{\@fnpos}
```

```

6218 {\@doreinfeetX\X@doreinfeet}%
6219 {%
6220 \IfStrEq{critical-familiar}{\@fnpos}%
6221 {\X@doreinfeet\@doreinfeetX}%
6222 {\@doreinfeetX\X@doreinfeet}%
6223 }%
6224 }
6225
6226 %

```

\X@doreinfeet \X@doreinfeet is the code for catering for the extra critical footnotes within \@reinserts.

```

6227 \newcommand*{\X@doreinfeet}{%
6228 \unless\ifnocritical@%
6229 \def\do##1{%
6230 \ifvoid\csuse{##1footins}\else%
6231 \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
6232 \fi}%
6233 \dolistloop{\@series}
6234 \fi%
6235 }
6236
6237 %

```

\print@notesX We have to add all the new kinds of familiar footnotes to the output routine. A macro which appends the familiar footnotes of one series onto the output routine, also adding vertical skip before notes.

\append@notesX

\do@feetX

\@doreinfeetX

```

6238 \newcommand{\append@notesX}[1]{%
6239 \ifvoid\csuse{footins#1}\else%
6240 \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
6241 \global\advance\skip\csuse{footins#1} by\csuse{afterterruleX@#1}%
6242 \print@notesX{#1}%
6243 \fi%
6244 }%
6245 %

```

The normal way to print one series of notes. \print@Xnotes is replaced by reledpar when using \Pages.

```

6246 \newcommand\print@notesX[1]{%
6247 \xdef\@currentseries{#1}%
6248 \csuse{footstart#1}{#1}%
6249 \csuse{footgroup#1}{#1}%
6250 }%
6251 %

```

We print all the series of notes by looping on them. We check before printing them that they are not voided.

```

6252 \newcommand*{\do@feetX}{%

```

```

6253 \unless\ifnofamiliar{%
6254   \gdef\firstseriesX@{%
6255     \setbox\@outputbox \vbox{%
6256       \unvbox\@outputbox%
6257       \def\do##1{%
6258         \append@notesX{##1}%
6259       }%
6260       \dolistloop{\@series}}%
6261   \fi%
6262 }%
6263
6264 \newcommand{\@doreinfeetX}{%
6265   \unless\ifnofamiliar{%
6266     \def\do##1{%
6267       \ifvoid\csuse{footins##1}\else
6268         \insert%
6269           \csuse{footins##1}
6270           {\unvbox\csuse{footins##1}}%
6271       \fi%
6272     }%
6273     \dolistloop{\@series}%
6274   \fi%
6275 }%
6276
6277 %

```

XXIII.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.)

```

6278 \@ifclassloaded{memoir}{%
6279 %

```

memoir is loaded so we use memoir's built in hooks.

```

6280   \g@addto@macro{\m@mddoextrafeet}{\l@dddoextrafeet}%
6281   \g@addto@macro{\m@mddodoreinextrafeet}{\l@ddodoreinextrafeet}%
6282 }{%
6283 %

```

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`.

```

6284 \ifboolexpr{%
6285   test{\@ifpackageloaded{fancyhdr}}%
6286   and test {\ifdef{\latex@makecol}}%
6287 }{%
6288   \patchcmd%
6289     {\latex@makecol}%
6290     {\xdef\@freelist{\@freelist\@midlist}}%
6291     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6292     {}%
6293     {\led@error@fail@patch@makecol}%
6294 }{%
6295   \patchcmd%
6296     {\@makecol}%
6297     {\xdef\@freelist{\@freelist\@midlist}}%
6298     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
6299     {}%
6300     {\led@error@fail@patch@makecol}%
6301 }%
6302
6303 \patchcmd%
6304   {\@reinserts}%
6305   {\ifvbox}%
6306   {\l@ddodoreinextrafeet\ifvbox}%
6307   {}%
6308   {\led@error@fail@patch@reinserts}%
6309 }
6310
6311 %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

6312 \newif\if@led@nofoot
6313
6314 %

```

```

6315 \@ifclassloaded{memoir}{%
6316 %

```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet 17 \g@addto@macro{\@mem@extranofeet}{%%
6318 \def\do#1{%
6319   \unless\ifnocritical@%
6320     \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
6321   \fi%
6322   \unless\ifnofamiliar@%
6323     \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
6324   \fi%

```

```

6325 }
6326 \dolistloop{\@series}%
6327 }%
6328 }%
6329 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot%30 \newcommand*{\@led@testifnofoot}{%
\@doclearpage%31 \@led@nofoottrue%
6332 \ifvoid\footins\else%
6333 \@led@nofootfalse%
6334 \fi%
6335 \def\do##1{%
6336 \unless\ifnocritical@%
6337 \ifvoid\csuse{##1footins}\else%
6338 \@led@nofootfalse%
6339 \fi%
6340 \fi%
6341 \unless\ifnofamiliar@%
6342 \ifvoid\csuse{footins##1}\else%
6343 \@led@nofootfalse%
6344 \fi%
6345 \fi%
6346 }%
6347 \dolistloop{\@series}%
6348 }%
6349
6350 \pretocmd%
6351 {\@doclearpage}%
6352 {\@led@testifnofoot}%
6353 {}%
6354 {\led@error@fail@patch@@doclearpage}%
6355
6356 \patchcmd%
6357 {\@doclearpage}%
6358 {\ifvoid\footins}%
6359 {\if@led@nofoot}%
6360 {}%
6361 {\led@error@fail@patch@@doclearpage}%
6362
6363 }
6364
6365 %

```


XXIV Page numbering in parallel typesetting

The `reledpar` package has two options which change the way the page numbering works. We need to implement these options on `reledmac` and not on `reledpar` because they have some consequences on the `reledmac` auxiliaries files (numbered file see V.12 p. 127). The `sameparallelpagenumbers` option allows the same page number on both left and right side. The `prevpgnotnumbered` option allows an empty (not numbered) right-side page before `\Pages`.

We cannot implement these two options by changing the value of the `page` counter, since its value is used by many \LaTeX features to determine whether a page is left (even-numbered) or right (odd-numbered). Consequently, we have to do it by patching `\thepage`, in order to use the value of the `par@page` counter instead of value of `page` counter.

This counter will be increased in a patched version of the \LaTeX 's `\@outputpage` macro, as is the `page` counter in this macro. However, this increase will take account of the options.

`\par@patch@thepage` `\par@patch@thepage` patches `\thepage` in order to use the value of `par@page` counter and not the value of `page`. It must be called after any redefinition of `\thepage`. That is why we insert it at the end of the \LaTeX macro `\pagenumbering`, which is called by some `\xxxmatter` commands. In cases when we are using the `memoir` class, we insert it at the end of `\@memmpnum`. When using `\pagenumbering`, we also need to restart `par@page` counter. Consequently, we have wrapped `\par@patch@thepage` and counter restart in `\par@patch@pagenumbering`. We also call `\par@patch@thepage` at the beginning of the document.

```

6366
6367 \newcommand{\par@patch@thepage}{%
6368   \ifboolexpr{%
6369     bool{sameparallelpagenumbers}%
6370     or bool{prevpgnotnumbered}%
6371   }%
6372   {%
6373     \patchcmd{\thepage}%
6374       {page}{par@page}%
6375       {}%
6376       {\led@error@fail@patch@thepage}%
6377   }{}%
6378 }%
6379
6380 \newcommand{\par@patch@pagenumbering}{%
6381   \ifboolexpr{%
6382     bool{sameparallelpagenumbers}%
6383     or bool{prevpgnotnumbered}%
6384   }%
6385   {%
6386     \setcounter{par@page}{1}%
6387   }%
6388   {}%

```

```

6389 \par@patch@thepage%
6390 }%
6391
6392 \ifl@dmemoir%
6393 \apptocmd{\@mempnum}%
6394 {\par@patch@pagenumbering}%
6395 {}%
6396 {\led@error@fail@patch@@mempnum}%
6397
6398 \else%
6399 \apptocmd{\pagenumbering}%
6400 {\par@patch@pagenumbering}%
6401 {}%
6402 {\led@error@fail@patch@pagenumbering}%
6403 \fi%
6404
6405 \AtBeginDocument{\par@patch@thepage}%
6406 %

```

\@outputpage As its name says, \@outputpage is a L^AT_EX's macro called in the output routine. It is this macro which increases the page counter. We patch it in order to increase, conditionally, the par@page counter.

```

6407 \AtBeginDocument{%
6408 \apptocmd{\@outputpage}{%
6409 \ifsameparallelpagelength%
6410 \ifl@dprintingpages%
6411 \ifodd\c@page\else%
6412 \stepcounter{par@page}%
6413 \fi%
6414 \else%
6415 \stepcounter{par@page}%
6416 \fi%
6417 \else%
6418 \stepcounter{par@page}%
6419 \fi%
6420 }%
6421 {}%
6422 {\led@error@fail@patch@@outputpage}%
6423 }%
6424 %

```

\thepar@page The par@page counter.

```

6425 \newcounter{par@page}%
6426 \setcounter{par@page}{1}%
6427 %

```

XXV 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.

```
6428 \list@create{\labelref@list}
6429 %
```

`\zz@@@` Two convenience macros to zero three / four labeling counters in one go.

```
\zz@@@
6430 \newcommand*\zz@@@{000|000|000}% Set three counters to zero in one go
6431 \newcommand*\zz@@@{000|000|000|000}% Set four counters to zero in one go
6432
6433 %
```

`\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.³⁶

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 Billett³⁷ found that the original code could be off by several pages. This version, hopefully cures that, and also allows for non-arabic page numbering.

```
6434 \newcommand*\edlabel}[1]{%
6435   \leavevmode%
6436   \@bsphack%
6437   \ifbool{expr{bool{ledRcol} or bool{ledRcol@}}}{%
6438     \ifXnote@%
6439     \protected@write\@auxout{%
6440       {\string\ledmake@labelsR\space\thepage|\ledparsedstartline|\ledparsedstartsub||\annot@start|\the\c@pstartR|{#1}}%
6441       \ifdef{\hypertarget}%
6442         {\Hy@raisedlink{\hypertarget{#1}}}%
6443       {}%
6444     }
```

³⁶The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

³⁷(jdb43@cam.ac.uk) via the ctt thread ‘ledmac cross referencing’, 25 August 2003.

```

6444 \else%
6445 \write\linenum@outR{\string\@lab}%
6446 \ifx\labelref@listR\empty%
6447 \xdef\label@refs{\zz@@@}%
6448 \else%
6449 \gl@p\labelref@listR\to\label@refs%
6450 \fi%
6451 \ifvmode%
6452 \advancelabel@refs%
6453 \fi%
6454 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

6455 \protected@write\@auxout{%
6456 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartR
|{#1}}%
6457 \ifdef{\hypertarget}%
6458 {\Hy@raisedlink{\hypertarget{#1}{}}}%
6459 {}%
6460 \fi%
6461 }{%
6462 \ifXnote@
6463 \ifl@dpairing%pstart or pstartL?
6464 \protected@write\@auxout{%
6465 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1}}%
6466 \ifdef{\hypertarget}%
6467 {\Hy@raisedlink{\hypertarget{#1}{}}}%
6468 {}%
6469 \else%
6470 \protected@write\@auxout{%
6471 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\annot@start|\the\c@pstart|{#1}}%
6472 \ifdef{\hypertarget}%
6473 {\Hy@raisedlink{\hypertarget{#1}{}}}%
6474 {}%
6475 \fi%
6476 \else%
6477 \write\linenum@out{\string\@lab}%
6478 \ifx\labelref@list\empty%
6479 \xdef\label@refs{\zz@@@}%
6480 \else%
6481 \gl@p\labelref@list\to\label@refs%
6482 \fi%
6483 \ifvmode%
6484 \advancelabel@refs%
6485 \fi%
6486 \ifl@dpairing%Pstart or PstartL?
6487 \protected@write\@auxout{%

```

```

6488      {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}}%
6489      \ifdef\hypertarget}%
6490      {\Hy@raisedlink{\hypertarget{#1}{}}}%
6491      {}%
6492      \else%
6493      \protected@write\auxout{}%
6494      {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}}%
6495      \ifdef\hypertarget}%
6496      {\Hy@raisedlink{\hypertarget{#1}{}}}%
6497      {}%
6498      \fi%
6499      \fi%
6500      }%
6501      \@esphack}%
6502      %
6503      %

```

`\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.

```

6504 \newcounter{line}%
6505 \newcounter{subline}%
6506 \newcounter{absline}%
6507 \newcommand{\advancelabel@refs}{%
6508   \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
6509   \stepcounter{line}%
6510   \setcounter{absline}{\expandafter\labelrefsparseabsline\label@refs}%
6511   \stepcounter{absline}%
6512   \ifsublines@%
6513     \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
6514     \stepcounter{subline}{1}%
6515     \def\label@refs{\theline|\thesubline|\theabsline}%
6516   \else%
6517     \def\label@refs{\theline|0|\theabsline}%
6518   \fi%
6519 }
6520 \def\labelrefsparseline#1|#2|#3{#1}%
6521 \def\labelrefsparsesubline#1|#2|#3{#2}%
6522 \def\labelrefsparseabsline#1|#2|#3{#3}%
6523 %

```

`\l@dmake@labels` The `\l@dmake@labels` macro gets executed when the labels file is read. For each label it defines a macro, whose name is made up partly from the label you supplied, that

contains the page, line and sub-line numbers. But first it checks to see whether the label has already been used (and complains if it has).

The initial use of `\newcommand` is to catch if `\l@dmake@labels` has been previously defined (by a class or package).

#1 page number, #2 line number, #3 sub-line number, #4 absolute line number, #5 line number annotation, #6 pstart number, #7 label.

```

6524 \newcommand*{\l@dmake@labels}{}
6525 \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
6526   \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
6527   \relax%
6528   \else%
6529     \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
6530   \fi
6531   \global\providetoggle{label@#7@ledRcol}%False is the default value of
this toggle, which tells us whether a label is linked to the right or left
side
6532   \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\relax}%
6533   \ignorespaces}
6534
6535 %

```

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.

```

6536 \AtBeginDocument{%
6537   \def\l@dmake@labels#1|#2|#3|#4|#5|#6|#7{%
6538   }
6539
6540 %

```

\@lab The `\@lab` command, which appears in the `\linenum@out` file, appends the current value of page, line, sub-line, and absolute line to the `\labelref@list`. These values are defined by the earlier `\@page`, `\@nl`, and the `\sub@on` and `\sub@off` commands appearing in the `\linenum@out` file.

TeX uses the page counter for page numbers. However, it appears that this is not the right place to grab the page number. That task is now done in the `\edlabel` macro. This version of `\@lab` appends just the current line and sub-line numbers to `\labelref@list`.

```

6541
6542 \newcommand*{\@lab}{%
6543   \ifledRcol
6544     \xright@appenditem{\linenumr@p{\line@numR}}{|%
6545       \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|
6546       \the\absline@numR}%
6547     \current@annot%
6548   }%
6549   \to\labelref@listR

```

```

6550 \else
6551 \xright@appenditem{%
6552 \linenumr@p{\line@num}|%
6553 \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|%
6554 \the\absline@num|}%
6555 \current@annot%
6556 }%
6557 \to\labelref@list
6558 \fi}
6559 %

```

\applabel \applabel, if called in \edtext will insert automatically both a starting and an ending label for the current edtext lines.

```

6560 \newcommand*{\applabel}[1]{%
6561 \if@edtext@secondarg@%
6562 %

```

Label should not be already defined.

```

6563 \ifcsundef{the@label#1}{%
6564 \csdef{the@label#1}{\applabel}%
6565 }%
6566 {%
6567 \led@warn@DuplicateLabel{#1 (\applabel)}%
6568 }%
6569 %

```

Parse the \edtext line numbers and annotations.

```

6570 \expandafter\l@dp@rsefootspec\l@d@nums|%
6571 \expandafter\parse@annot\l@current@annot|%
6572 %

```

Use the \TeX standard hack for label.

```

6573 \@bsphack%
6574 %

```

And now, write the data in the auxiliary file.

```

6575 \ifledRcol%
6576 \protected@write\@auxout{}%
6577 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\annot@start|\the\c@pstartR|{#1:start}}%
6578 \ifdef{\hypertarget}%
6579 {\Hy@raisedlink{\hypertarget{#1:start}}}%
6580 {}%
6581 \protected@write\@auxout{}%
6582 {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|\annot@end|\the\c@pstartR|{#1:end}}%
6583 \else%

```

```

6584 \ifl@dpairing%pstart or pstartL?
6585 \protected@write\@auxout{}%
6586 {\string\l@dmake@labels\space\l@dparsedstartpage|\
l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1:
start}}}%
6587 \ifdef{\hypertarget}%
6588 {\Hy@raisedlink{\hypertarget{#1:start}}{}}}%
6589 {}%
6590 \protected@write\@auxout{}%
6591 {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub||\annot@end|\the\c@pstartL|{#1:end}}}%
6592 \else%
6593 \protected@write\@auxout{}%
6594 {\string\l@dmake@labels\space\l@dparsedstartpage|\
l@dparsedstartline|\l@dparsedstartsub||\annot@start|\the\c@pstartL|{#1:start
}}}%
6595 \ifdef{\hypertarget}%
6596 {\Hy@raisedlink{\hypertarget{#1:start}}{}}}%
6597 {}%
6598 \protected@write\@auxout{}%
6599 {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub||\annot@end|\the\c@pstartL|{#1:end}}}%
6600 \fi%
6601 \fi%
6602 %

```

Use the \TeX standard hack for label.

```

6603 \@esphack%
6604 %

```

Warning if `\applabel` is called outside of `\edtext`.

```

6605 \else%
6606 \led@warn@AppLabelOutSecondArgEdtext{#1}%
6607 \fi%
6608 %

```

End of `\applabel`

```

6609 }%
6610 %

```

`\edlabels` `\edlabels` and `\edlabelE` are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
6611 \newcommand{\edlabels}[1]{%
6612 \edlabel{#1:start}%
6613 }
6614 \newcommand{\edlabelE}[1]{%
6615 \edlabel{#1:end}%
6616 }
6617 \newcommand{\edlabelSE}[1]{%

```



```

6618 \edlabelS{#1}%
6619 \edlabelE{#1}%
6620 }
6621 %

```

\wrap@edcrossref \wrap@edcrossref is called around all reledmac crossref commands, except those which start with x. It adds the hyperlink.

```

6622 \newrobustcmd{\wrap@edcrossref}[2]{%
6623 \ifdef{\hyperlink}%
6624 {\hyperlink{#1}{#2}}%
6625 {#2}%
6626 }
6627 %

```

\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.

TeX already defines a \pageref, so changing the name to \edpageref.

```

6628 \newcommand*\edpageref[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
6629 \newcommand*\xpageref[1]{\l@dgetref@num{1}{#1}}
6630
6631 %

```

\edlineref If the specified label exists, \lineref gives its line number.

```

6632 \newcommand*\edlineref[1]{%
6633 \l@dref@undefined{#1}%
6634 \wrap@edcrossref{#1}{%
6635 \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
6636 \iftoggle{label@#1@ledRcol}%
6637 {\linenumrepR{\l@dgetref@num{2}{#1}}}%
6638 {\linenumrep{\l@dgetref@num{2}{#1}}}%
6639 \xflagref{#1}%
6640 }%
6641 }%
6642 \newcommand*\xlineref[1]{\l@dgetref@num{2}{#1}}%
6643
6644 %

```

\sublineref If the specified label exists, \sublineref gives its sub-line number.

\xsublineref

```

6645 \newcommand*{\sublineref}[1]{%
6646   \l@dref@undefined{#1}%
6647   \wrap@edcrossref{#1}{%
6648     \providetoggle{label@#1@dRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
6649     \iftoggle{label@#1@dRcol}%
6650       {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
6651       {\sublinenumrep{\l@dgetref@num{3}{#1}}}%
6652     }%
6653   }%
6654 \newcommand*{\xsublineref}[1]{\l@dgetref@num{3}{#1}}
6655
6656 %

```

\xabslineref If the specified label exists, \xabslineref gives its absolute line number. This is generally used only by reledmac internal macros.

```

6657 \newcommand*{\xabslineref}[1]{\l@dgetref@num{4}{#1}}%
6658 %

```

\annotationref If the specified label exists, \edannotationref gives the line number annotation with which it is associated.

\xannotationref

```

6659 \newcommand*{\annotationref}[1]{%
6660   \l@dref@undefined{#1}%
6661   \wrap@edcrossref{#1}{%
6662     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@ref}{%
6663       \l@dgetref@num{5}{#1}%
6664     }%
6665   }%
6666 }%
6667 \newcommand*{\xannotationref}[1]{%
6668   \l@dgetref@num{5}{#1}%
6669 }%
6670 %

```

\pstartref If the specified label exists, \pstartref gives its pstart number.

\xpstartref

```

6671 \newcommand*{\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{6}{#1}}}%
6672 \newcommand*{\xpstartref}[1]{\l@dgetref@num{6}{#1}}%
6673
6674 %

```

\xflagref \xflagref finds the side flag of any ref defined with \edlabel.

```

6675 \newcommand*{\xflagref}[1]{\l@dgetref@num{7}{#1}}%
6676 %

```

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.

```

6677 \newcommand*{\l@dref@undefined}[1]{%
6678   \expandafter\ifx\csname the@label#1\endcsname\relax
6679   \led@warn@RefUndefined{#1}%
6680   \fi}
6681
6682 %

```

\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.

```

6683 \newcommand*{\l@dgetref@num}[2]{%
6684   \expandafter
6685   \ifx\csname the@label#2\endcsname \relax
6686     000%
6687   \else
6688     \expandafter\expandafter\expandafter
6689     \l@dlabel@parse\csname the@label#2\endcsname|1%
6690   \fi}
6691
6692 %

```

\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 argument. The `|`-delimited arguments consist of the expanded label-macro (three reference numbers), followed by the switch-number (1, 2, 3, 4, 5, 6) which defines which of the former seven numbers to extract. (It was given earlier as the first argument of `\l@dgetref@num`.)

```

6693 \newcommand*{\l@dlabel@parse}{%
6694   \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7|#8{%
6695     \ifcase #8%
6696       \or #1%
6697       \or #2%
6698       \or #3%
6699       \or #4%
6700       \or #5%
6701       \or #6%

```

```

6702 \or #7%
6703 \fi}
6704 %

```

\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.

```

6705 \newcommand*\xxref}[2]{%
6706 {%
6707   \expandafter\ifx\csname the@label#1\endcsname\relax%
6708     \let\@tempa\zz@@@%
6709     \def\@tempc{%
6710       \else%
6711         \def\@tempa{%
6712           \ldgetref@num{1}{#1}|%
6713           \ldgetref@num{2}{#1}|%
6714           \ldgetref@num{3}{#1}%
6715         }%
6716         \edef\@tempc{%
6717           \ldgetref@num{5}{#1}%
6718         }%
6719       \fi%
6720       \expandafter\ifx\csname the@label#2\endcsname\relax%
6721         \let\@tempb\zz@@@%
6722         \def\@tempd{%
6723           \else%
6724             \def\@tempb{%
6725               \ldgetref@num{1}{#2}|%
6726               \ldgetref@num{2}{#2}|%
6727               \ldgetref@num{3}{#2}%
6728             }%
6729             \edef\@tempd{%
6730               \ldgetref@num{5}{#2}%
6731             }%
6732           \fi%
6733           \global\appto\@beforeinsertofthisedtext{%
6734             \def\@this@crossref@start{#1}%
6735           }%
6736           \global\appto\@beforeinsertofthisedtext{%
6737             \def\@this@crossref@end{#2}%
6738           }%
6739           \linenum{\@tempa|\@tempb}%
6740           \edef\@tempe{\@tempc|\@tempd}%

```

```

6741 \expandafter\lineannot\expandafter{\@tempe}%
6742 }%
6743 }%
6744
6745
6746 %

```

`\appref` `\appref`, `\Seref`, `\apprefwithpage`, `\Serefwithpage` and `\SEonlypage` print cross-ref to some start / end lines defined by specific commands. It prints the lines as they should be printed in the apparatus (critical notes for not suffixed versions, endnotes for suffixed versions).

`\apprefwithpage` `\Seref` 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`.

```

6747 \def\Xtwolines@appref{%
6748 \def\Xtwolines@Seref{%
6749
6750 \def\Xmorethantwolines@appref{%
6751 \def\Xmorethantwolines@Seref{%
6752
6753 \def\Xlinerangeseparator@appref{\endashchar}%
6754 \def\Xlinerangeseparator@Seref{\endashchar}%
6755
6756 \def\Xsublinesep@appref{\fullstop}%
6757 \def\Xsublinesep@Seref{\fullstop}%
6758
6759 \def\Xpagelinesep@appref{\fullstop}%
6760 \def\Xpagelinesep@Seref{\fullstop}%
6761
6762
6763 \newtoggle{Xtwolinesbutnotmore@appref}%
6764 \newtoggle{Xtwolinesbutnotmore@Seref}%
6765
6766 \newtoggle{Xtwolinesonlyinsamepage@appref}%
6767
6768 \newtoggle{Xtwolinesonlyinsamepage@Seref}%
6769
6770 \newtoggle{Xlineflag@appref}%
6771 \toggletrue{Xlineflag@appref}%Here exception
6772 \newtoggle{Xlineflag@Seref}%
6773 \toggletrue{Xlineflag@Seref}%%Here exception
6774
6775 \def\Xlinenumannotationposition@appref{after}%
6776 \def\Xlinenumannotationposition@Seref{after}%
6777
6778 \def\Xwraplinenumannotation@appref{\textsuperscript}%
6779 \def\Xwraplinenumannotation@Seref{\textsuperscript}%
6780

```

```

6781 \def\Xendtwolines@apprefwithpage{}%
6782 \def\Xendtwolines@Serefwithpage{}%
6783
6784 \def\Xendmoreethantwolines@apprefwithpage{}%
6785 \def\Xendmoreethantwolines@Serefwithpage{}%
6786
6787 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
6788 \def\Xendlinerangeseparator@Serefwithpage{\endashchar}
6789 \def\Xendlinerangeseparator@Serefonlypage{\endashchar}
6790
6791 \def\Xendbeforepagenumber@apprefwithpage{p.}%
6792 \def\Xendbeforepagenumber@Serefwithpage{p.}%
6793 \def\Xendbeforepagenumber@SEonlypage{p.}%
6794
6795 \def\Xendafterpagenumber@apprefwithpage{) }%
6796 \def\Xendafterpagenumber@Serefwithpage{) }%
6797
6798
6799 \def\Xendlineprefixsingle@apprefwithpage{}%
6800 \def\Xendlineprefixsingle@Serefwithpage{}%
6801
6802 \def\Xendlineprefixmore@apprefwithpage{}%
6803 \def\Xendlineprefixmore@Serefwithpage{}%
6804
6805 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
6806 \newtoggle{Xendtwolinesbutnotmore@Serefwithpage}%
6807
6808 \def\Xendsublinesep@apprefwithpage{\fullstop}%
6809 \def\Xendsublinesep@Serefwithpage{\fullstop}%
6810
6811 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
6812 \newtoggle{Xendtwolinesonlyinsamepage@Serefwithpage}%
6813
6814 \newtoggle{Xendlineflag@apprefwithpage}
6815 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
6816 \newtoggle{Xendlineflag@Serefwithpage}
6817 \toggletrue{Xendlineflag@Serefwithpage}%Here, exception
6818
6819 \def\Xendlinenumannotationposition@apprefwithpage{after}%
6820 \def\Xendlinenumannotationposition@apprefwithpage@Serefwithpage{after}%
6821
6822 \def\Xendwraplinenumannotation@apprefwithpage{\textsuperscript}%
6823 \def\Xendwraplinenumannotation@Serefwithpage{\textsuperscript}%
6824 %

```

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.

```

6825
6826 \gdef\Xboxstartlinenum@appref{Opt}

```

```

6827 \gdef\Xboxstartlinenum@SEref{Opt}
6828
6829 \gdef\Xboxendlinenum@appref{Opt}
6830 \gdef\Xboxendlinenum@SEref{Opt}
6831
6832 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}
6833 \gdef\Xendboxstartlinenum@SErefwithpage{Opt}
6834
6835 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
6836 \gdef\Xendboxendlinenum@SErefwithpage{Opt}
6837
6838 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
6839 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
6840
6841 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
6842 \newtoggle{Xendpagenumberonlyfirstifsingle@SErefwithpage}
6843
6844 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
6845 \newtoggle{Xendpagenumberonlyfirstintwo@SErefwithpage}
6846
6847 \gdef\Xendsympagenum@apprefwithpage{}
6848 \gdef\Xendsympagenum@SErefwithpage{}
6849
6850 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
6851 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
6852
6853 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

6854 \newcommand\@apprefprefixsingle{}%
6855 \newcommand\@SErefprefixsingle{}%
6856
6857 \newcommand\@apprefprefixmore{}%
6858 \newcommand\@SErefprefixmore{}%
6859
6860 \newcommand{\setapprefprefixsingle}[1]{%
6861   \gdef\@apprefprefixsingle{#1}%
6862 }
6863 \newcommand{\setSErefprefixsingle}[1]{%
6864   \gdef\@SErefprefixsingle{#1}%
6865 }
6866
6867 \newcommand{\setapprefprefixmore}[1]{%
6868   \gdef\@apprefprefixmore{#1}%
6869 }
6870 \newcommand{\setSErefprefixmore}[1]{%
6871   \gdef\@SErefprefixmore{#1}%
6872 }
6873

```

```

6874 %
And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.
6875 \newcommand{\setSErefonlypageprefixsingle}[1]{%
6876   \gdef\SErefonlypage@prefixsingle{#1}%
6877 }%
6878 \newcommand{\setSErefonlypageprefixmore}[1]{%
6879   \gdef\SErefonlypage@prefixmore{#1}%
6880 }%
6881 %

```

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.

```

6882
6883 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
6884 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
6885
6886 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
6887   {#1}{#2}{appref}}
6888 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
6889   {#1}{#2}{SEref}}
6890
6891 \newcommand{\reformatted@}[3]{%
6892   \def\do##1{%
6893     \setkeys[mac]{truefootnoteoption}{##1}%
6894   }%
6895   \notblank{#1}{\docsvlist{#1}}}%
6896   \xdef\@currentseries{#3}%
6897   \ifcempty{@#3prefixmore}%
6898     {\@apprefprefixsingle}%
6899     {%
6900       \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
6901       {\csuse{@#3prefixsingle}}%
6902       {\csuse{@#3prefixmore}}%
6903     }%
6904   \ifboolexpr{%
6905     test{\ifcsundef{the@label#2:start}}%
6906     or test{\ifcsundef{the@label#2:end}}%
6907   }%
6908   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6909   {%
6910     \def\@this@crossref@start{#2:start}%
6911     \def\@this@crossref@end{#2:end}%
6912     \xdef\annot@start{\xannotationref{#2:start}}%

```



```

6913 \xdef\annot@end{\xannotationref{#2:end}}%
6914 \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax\
xflagref{#2:start}|%
6915 \undef\@this@crossref@end%
6916 \undef\@this@crossref@start%
6917 }%
6918 \def\do##1{%
6919 \setkeys[mac]{falsefootnoteoption}{##1}%
6920 }%
6921 \notblank{#1}{\docsvlist{#1}}{ }%
6922 }%
6923
6924 \newcommand{\reformattedwithpage@}[3]{%
6925 \def\do##1{%
6926 \setkeys[mac]{truefootnoteoption}{##1}%
6927 }%
6928 \notblank{#1}{\docsvlist{#1}}{ }%
6929 \xdef\@currentseries{#3withpage}%
6930 \ifboolexpr{%
6931 test{\ifcsundef{the@label#2:start}}%
6932 or test{\ifcsundef{the@label#2:end}}%
6933 }%
6934 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6935 {%
6936 \def\@this@crossref@start{#2:start}%
6937 \def\@this@crossref@end{#2:end}%
6938 \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax\
xflagref{#2:start}|%
6939 \undef\@this@crossref@end%
6940 \undef\@this@crossref@start%
6941 }%
6942 \def\do##1{%
6943 \setkeys[mac]{falsefootnoteoption}{##1}%
6944 }%
6945 \notblank{#1}{\docsvlist{#1}}{ }%
6946 }%
6947
6948 \newcommand{\reformattedonlypage@}[3]{%
6949 \def\do##1{%
6950 \setkeys[mac]{truefootnoteoption}{##1}%
6951 }%
6952 \notblank{#1}{\docsvlist{#1}}{ }%
6953 \xdef\@currentseries{#3onlypage}%
6954 \ifboolexpr{%
6955 test{\ifcsundef{the@label#2:start}}%
6956 or test{\ifcsundef{the@label#2:end}}%
6957 }%
6958 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%

```

```

6959 {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}}%
6960     {%
6961     \ifcsvoid{#3onlypage@prefixsingle}%
6962     {}%
6963     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
}}}%
6964     \printnpnum{%
6965     \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
6966     }%
6967     }%
6968     {%
6969     \ifcsvoid{#3onlypage@prefixmore}%
6970     {}%
6971     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}}%
6972     \ifdefined\linangesep@%
6973     \printnpnum{%
6974     \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
6975     \linangesep@%
6976     \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
6977     }%
6978     \else%
6979     \printnpnum{%
6980     \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
6981     \csuse{Xendlinangeseparator@ \@currentseries}%
6982     \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
6983     }%
6984     \fi%
6985     }%
6986     }%
6987     \def\do##1{%
6988     \setkeys{mac}{falsefootnoteoption}{##1}%
6989     }%
6990     \notblank{#1}{\docsvlist{#1}}}%
6991     }%
6992     %

```

`\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`.

```

6993 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\
endcsname{#2}}
6994
6995 %

```

(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. 142 and V.9 p. 106), since `\xxref` makes a call to `\linenum` in order to do its work.)

XXV.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`.

```
6996 \pretocmd{\XR@test}{%
6997   {\XR@test@mac+++#1#2#3#4+++}%
6998   {}%
6999   {}%
7000   {%
```

`\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`.

```
7001 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
7002 %
```

`\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 `\ldmake@labels` and `\ldmake@labelsR`.

```
7003 \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
7004   \ifx#1\ldmake@labels%
7005     \ldmake@labels#2%
7006   \else
7007     \ifx#1\ldmake@labelsR%
7008       \ldmake@labelsR #2%
7009     \fi%
7010   \fi%
7011 }%
7012 %
```

XXVI Sidenotes

Regular `\marginpars` do not work inside numbered text — they do not produce any note but do put an extra unnumbered blank line into the text.

`\@xympar` Changing `\@xympar` a little at least ensures that `\marginpars` in numbered text do not disturb the flow.

```

7013 \pretocmd{\@xympar}%
7014   {\ifnumberedpar@
7015     \led@warn@NoMarginpars
7016     \@esphack
7017   \else}%
7018   {}%
7019   {}%
7020
7021 \apptocmd{\@xympar}%
7022   {\fi}%
7023   {}
7024   {}
7025
7026 %

```

We provide sidenotes 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 sidenote margin:

left: 0

right: 1

outer: 2

inner: 3

```

7027 \newcount\sidenote@margin
7028 \newcommand*{\sidenotemargin}[1]{%
7029   \l@dgetsidenote@margin{#1}%
7030   \ifnum\@l@dttempcntb>\m@ne
7031     \ifledRcol
7032       \global\sidenote@marginR=\@l@dttempcntb
7033     \else
7034       \global\sidenote@margin=\@l@dttempcntb
7035     \fi
7036   \fi}}
7037 \newcommand*{\l@dgetsidenote@margin}[1]{%
7038   \def\@tempa{#1}\def\@tempb{left}%
7039   \ifx\@tempa\@tempb
7040     \@l@dttempcntb \z@
7041   \else
7042     \def\@tempb{right}%
7043     \ifx\@tempa\@tempb
7044       \@l@dttempcntb \@ne
7045     \else
7046       \def\@tempb{outer}%

```

```

7047 \ifx\@tempa\@tempb
7048 \l@dtempcntb \tw@
7049 \else
7050 \def\@tempb{inner}%
7051 \ifx\@tempa\@tempb
7052 \l@dtempcntb \thr@@
7053 \else
7054 \led@warn@BadSidenotemargin
7055 \l@dtempcntb \m@ne
7056 \fi
7057 \fi
7058 \fi
7059 \fi}
7060 \sidenotemargin{right}
7061
7062 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
7063 \newbox\l@dlp@rbox
7064 \newbox\l@drp@rbox
7065
7066 %

```

`\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 7067 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep 7068 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup 7069 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup 7070 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
7071 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
7072 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
7073
7074 %

```

`\ledleftnote` `\ledrightnote`, `\ledinnernote`, `\ledouternote` are the user commands for left, right, inner and outer sidenotes. The two last one are just alias for the two first one, depending of the page number. `\ledsidenote{<text>}` is the command for a moveable sidenote.

```

\ledsidenote 7075 \newcommand*\ledleftnote[1]{\edtext{}\l@dlsnote{#1}}
7076 \newcommand*\ledrightnote[1]{\edtext{}\l@drsnote{#1}}
7077 \newcommand*\ledsidenote[1]{\edtext{}\l@dcnote{#1}}%
7078 \newcommand*\ledinnernote[1]{\edtext{}\l@disnote{#1}}%
7079 \newcommand*\ledouternote[1]{\edtext{}\l@dosnote{#1}}%
7080 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.

`\l@drsnote`
`\l@dcnote`
`\l@desnote`
`\l@disnote`

```

7081 \newif\ifrightnoteup
7082 \rightnoteuptrue
7083
7084 \newcommand*{\l@dlsnote}[1]{%
7085 \begingroup%
7086 \newcommand{\content}{#1}%
7087 \ifnumberedpar@
7088 \ifledRcol%
7089 \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}{%
7090 \to\inserts@listR
7091 \global\advance\insert@countR \@ne%
7092 \else%
7093 \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}{%
7094 \to\inserts@list
7095 \global\advance\insert@count \@ne%
7096 \fi
7097 \fi}%
7098 \ignorespaces%
7099 \endgroup%
7100 }%
7101
7102 \newcommand*{\l@drsnote}[1]{%
7103 \begingroup%
7104 \newcommand{\content}{#1}%
7105 \ifnumberedpar@
7106 \ifledRcol%
7107 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}{%
7108 \to\inserts@listR
7109 \global\advance\insert@countR \@ne%
7110 \else%
7111 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}{%
7112 \to\inserts@list
7113 \global\advance\insert@count \@ne%
7114 \fi
7115 \fi\ignorespaces%
7116 \endgroup%
7117 }%
7118
7119 \newcommand*{\l@dcsnote}[1]{%
7120 \begingroup%
7121 \newcommand{\content}{#1}%
7122 \ifnumberedpar@
7123 \ifledRcol%
7124 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}{%
7125 \to\inserts@listR
7126 \global\advance\insert@countR \@ne%
7127 \else%
7128 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}{%
7129 \to\inserts@list
7130 \global\advance\insert@count \@ne%

```

```

7131 \fi
7132 \fi\ignorespaces%
7133 \endgroup%
7134 }%
7135
7136 \newcommand*{\l@disnote}[1]{%
7137 \begingroup%
7138 \newcommand{\content}{#1}%
7139 \ifnumberedpar%
7140 \ifledRcol%
7141 \xright@appenditem{\noexpand\l@disnote{\expandonce\content}}{%
7142 \to\inserts@listR%
7143 \global\advance\insert@countR \@ne%
7144 \else%
7145 \xright@appenditem{\noexpand\l@disnote{\expandonce\content}}{%
7146 \to\inserts@list%
7147 \global\advance\insert@count \@ne%
7148 \fi%
7149 \fi\ignorespaces%
7150 \endgroup%
7151 }%
7152
7153 \newcommand*{\l@dosnote}[1]{%
7154 \begingroup%
7155 \newcommand{\content}{#1}%
7156 \ifnumberedpar%
7157 \ifledRcol%
7158 \xright@appenditem{\noexpand\l@dosnote{\expandonce\content}}{%
7159 \to\inserts@listR%
7160 \global\advance\insert@countR \@ne%
7161 \else%
7162 \xright@appenditem{\noexpand\l@dosnote{\expandonce\content}}{%
7163 \to\inserts@list%
7164 \global\advance\insert@count \@ne%
7165 \fi%
7166 \fi\ignorespaces%
7167 \endgroup%
7168 }%
7169
7170 %

```

\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 sidenotes. 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 **\vl@disnote** 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.

```

7171 \newcommand*{\vl@dlsnote}[1]{%
7172   \ifledRcol@%
7173     \@l@tempcntb=\sidenote@marginR%
7174     \ifnum\@l@tempcntb>\@ne%
7175       \advance\@l@tempcntb by\page@numR%
7176     \fi%
7177   \else%
7178     \@l@tempcntb=\sidenote@margin%
7179     \ifnum\@l@tempcntb>\@ne%
7180       \advance\@l@tempcntb by\page@num%
7181     \fi%
7182   \fi%
7183   \ifodd\@l@tempcntb%
7184     \listgadd{\l@dcsnotetext@l}{#1}%
7185   \else%
7186     \listgadd{\l@dcsnotetext}{#1}%
7187   \fi
7188 }
7189 \newcommand*{\vl@drsnote}[1]{%
7190   \ifledRcol@%
7191     \@l@tempcntb=\sidenote@marginR%
7192     \ifnum\@l@tempcntb>\@ne%
7193       \advance\@l@tempcntb by\page@numR%
7194     \fi%
7195   \else%
7196     \@l@tempcntb=\sidenote@margin%
7197     \ifnum\@l@tempcntb>\@ne%
7198       \advance\@l@tempcntb by\page@num%
7199     \fi%
7200   \fi%
7201   \ifodd\@l@tempcntb%
7202     \listgadd{\l@dcsnotetext}{#1}%
7203   \else%
7204     \listgadd{\l@dcsnotetext@r}{#1}%
7205   \fi%
7206 }
7207 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
7208
7209 \newcommand{\vl@disnote}[1]{%

```



```

7210 \ifledRcol@%
7211 \@tempcnta=\page@numR%
7212 \else%
7213 \@tempcnta=\page@num%
7214 \fi%
7215 \ifodd\@tempcnta% ODD => right page => inner side = left side
7216 \vl@dlsnote{#1}%
7217 \else%
7218 \vl@drsnote{#1}%
7219 \fi%
7220 }%
7221
7222 \newcommand{\vl@dosnote}[1]{%
7223 \ifledRcol@%
7224 \@tempcnta=\page@numR%
7225 \else%
7226 \@tempcnta=\page@num%
7227 \fi%
7228 \ifodd\@tempcnta% ODD => right page => outer side = right side
7229 \vl@drsnote{#1}%
7230 \else%
7231 \vl@dlsnote{#1}%
7232 \fi%
7233 }%
7234
7235 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box. And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

7236 \newcommand*{\setl@dlp@rbox}[1]{%
7237 \begingroup%
7238 \parindent\z@\hspace=\ledlsnotewidth%
7239 \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7240 \global\setbox\l@dlp@rbox%
7241 \ifleftnoteup%
7242 =\vbox to\z@{\{\ledlsnotefontsetup\vss #1}\}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
7243 \else%
7244 =\vbox to 0.70\baselineskip{\{\ledlsnotefontsetup\strut#1\vss}\}%
7245 \fi%
7246 \endgroup%
7247 }
7248
7249 \newcommand*{\setl@drp@rbox}[1]{%
7250 \begingroup%
7251 \parindent\z@\hspace=\ledrsnotewidth%

```

```

7252 \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
7253 \global\setbox\l@drp@rbox%
7254 \ifrightnoteup%
7255 =\vbox to\z@{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
7256 \else%
7257 =\vbox to0.7\baselineskip{\ledrsnotefontsetup\strut#1\vss}}%
7258 \fi%
7259 \endgroup%
7260 }%
7261 \newif\ifleftnoteup
7262 \leftnoteuptrue
7263 %

```

\@sidenotesep This macro is used to separate sidenotes of the same line.

```

7264 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
7265 \newcommand{\@sidenotesep}{, }
7266 %

```

\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.

```

7267 \newcommand*\affixside@note{%
7268 \prepare@edindex@fornote{\the\page@num|\the\line@num|\the\subline@num|\
the\page@num|\the\line@num|\the\subline@num}%
7269 \def\sidenotecontent@{%
7270 \numgdef\itemcount@{0}%
7271 \def\do##1{%
7272 \ifnumequal{\itemcount@}{0}%
7273 {%
7274 \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
7275 {\appto\sidenotecontent@{\@sidenotesep ##1}%
7276 }%
7277 \numgdef\itemcount@{\itemcount@+\@ne}%
7278 }%
7279 \dolistloop{\l@dcnotesetext}%
7280 \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
7281 %

```

And we do the same for left and right notes (not movable).

```

7282 \gdef\@templ@d{%
7283 \gdef\@templ@n{\l@dcnotesetext\l@dcnotesetext@l\l@dcnotesetext@r}%

```

```

7284 \ifx\@templ@d\@templ@n \else%
7285 \if@twocolumn%
7286 \if@firstcolumn%
7287 \setl@dlp@rbox{##1}{\sidenotecontent@}%
7288 \else%
7289 \setl@drp@rbox{\sidenotecontent@}%
7290 \fi%
7291 \else%
7292 \l@dttempcntb=\sidenote@margin%
7293 \ifnum\l@dttempcntb>\@ne%
7294 \advance\l@dttempcntb by\page@num%
7295 \fi%
7296 \ifodd\l@dttempcntb%
7297 \setl@drp@rbox{\sidenotecontent@}%
7298 \gdef\sidenotecontent@{}%
7299 \numgdef{\itemcount@}{0}%
7300 \dolistloop{\l@dcsnotetext@l}%
7301 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
7302 \setl@dlp@rbox{\sidenotecontent@}%
7303 \else%
7304 \setl@dlp@rbox{\sidenotecontent@}%
7305 \gdef\sidenotecontent@{}%
7306 \numgdef{\itemcount@}{0}%
7307 \dolistloop{\l@dcsnotetext@r}%
7308 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
7309 \setl@drp@rbox{\sidenotecontent@}%
7310 \fi%
7311 \fi%
7312 \fi%
7313 \advance\@edindex@for@note@m@one%
7314 }
7315 %

```

XXVII Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfeetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage`.
`\l@dfeetendmini` They can be extended to handle other things if necessary.

```

7316 \ifnoledgroup@\else%
7317 \newcommand*{\l@dfeetbeginmini}{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
7318 \newcommand*{\l@dfeetendmini}{%
7319 \IfStrEq{critical-familiar}{\@mpfnpos}%
7320 {\l@dedendmini\l@dfamendmini}%

```

```

7321      {%
7322          \IfStrEq{familiar-critical}{\@mpfnpos}%
7323              {\l@dfamendmini\l@dedendmini}%
7324              {\do@feet@custom@order{mp@}{\@mpfnpos}}}%
7325      }%
7326  }%
7327  %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

`\l@dedendmini`

`\mp@append@Xnotes`

```

7328 \newcommand*{\l@dedbeginmini}{%
7329     \unless\ifnocritical@%
7330         \def\do##1{%
7331             \csletcs{v##1footnote}{mpv##1footnote}%
7332         }%
7333         \dolistloop{\@series}%
7334     \fi%
7335 }
7336 \newcommand*{\l@dedendmini}{%
7337     \unless\ifnocritical@%
7338         \ifl@dpairing%
7339             \ifledRcol%
7340                 \flush@notesR%
7341             \else%
7342                 \flush@notes%
7343             \fi%
7344         \fi
7345         \def\do##1{%
7346             \mp@append@Xnotes{##1}%
7347         }%
7348         \dolistloop{\@series}%
7349     \fi%
7350 }%
7351 \newcommand{\mp@append@Xnotes}[1]{%
7352     \ifvoid\csuse{mp#1footins}\else%
7353         \ifl@dpairing%
7354             \ifparledgroup%
7355                 \ifledRcol%
7356                     \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
7357 \@nameuse{mp#1footins}}%
7358                 \else%
7359                     \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\
7360 skip\@nameuse{mp#1footins}}%
7361                 \fi%
7362             \fi%
7363             \ifcsstring{series@display#1}{paragraph}{\}%
7364             \setbox\@nameuse{mp#1footins}=\vbox{%
7365                 \csuse{Xnotefontsize@#1}%
7366                 \ifcsdef{Xhsize\csuse{series@display#1}@#1}{%

```

```

7366 \hsize\csuse{Xhsize\csuse{series@display#1}@#1}%
7367 }{}%
7368 \noindent\csuse{Xtxtbeforenotes@#1}%
7369 \unvbox\@nameuse{mp#1footins}%
7370 \@parboxrestore%
7371 }%
7372 }%
7373 \csuse{mp#1footgroup}{#1}%
7374 \fi%
7375 }%
7376 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
7377 \newcommand*{\l@dfambeginmini}{%
7378 \unless\ifnofamiliar%
7379 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
7380 \dolistloop{\@series}%
7381 \fi%
7382 }%
7383
7384 \newcommand*{\l@dfamendmini}{%
7385 \unless\ifnofamiliar%
7386 \def\do##1{%
7387 \mp@append@notesX{##1}%
7388 }%
7389 \dolistloop{\@series}%
7390 \fi%
7391 }%
7392 \newcommand{\mp@append@notesX}[1]{%
7393 \ifvoid\csuse{mpfootins#1}\else%
7394 \ifcsstring{series@displayX#1}{paragraph}{%
7395 \setbox\@nameuse{mpfootins#1}=\vbox{%
7396 \csuse{notefontsizeX@#1}%
7397 \ifcsdef{hsize\csuse{series@display#1}X@#1}{%
7398 \hsize\csuse{hsize\csuse{series@display#1}X@#1}%
7399 }{}%
7400 \noindent\csuse{txtbeforenotesX@#1}%
7401 \unvbox\@nameuse{mpfootins#1}%
7402 \@parboxrestore%
7403 }%
7404 }%
7405 \csuse{mpfootgroup#1}{#1}%
7406 \fi%
7407 }%
7408 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

7409 \patchcmd%

```

```

7410 {\@iiiminipage}%
7411 {\let\@footnotetext\@mpfootnotetext}%
7412 {\let\@footnotetext\@mpfootnotetext\l@dfeetbeginmini}%
7413 {}%
7414 {\led@error@fail@patch@\@iiiminipage}%
7415 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

7416 \patchcmd%
7417 {\endminipage}%
7418 {\footnoterule}%
7419 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
7420 {}%
7421 {\led@error@fail@patch@endminipage}
7422
7423 \patchcmd%
7424 {\endminipage}%
7425 {\@minipagefalse}%
7426 {\l@dfeetendmini\@minipagefalse}%
7427 {}%
7428 {\led@error@fail@patch@endminipage}
7429
7430 %

```

`\l@dunboxmpfoot` `\l@dunboxmpfoot` insert normal footnotes for `ledgroup`.

`\l@advance@parledgroup@beforenormalnotes`

```

7431 \newcommand*{\l@dunboxmpfoot}{%
7432   \vskip\skip\@mpfootins
7433   \normalcolor
7434   \footnoterule
7435   \l@advance@parledgroup@beforenormalnotes
7436   \unvbox\@mpfootins%
7437 }
7438 %

```

When using parallel `ledgroup`, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

7439 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
7440   \ifparledgroup
7441     \ifl@pairing
7442       \ifledRcol
7443         \dimgdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
7444       \else
7445         \dimgdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
7446       \fi
7447     \fi
7448   \fi

```

```
7449 }
7450 %
```

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.

```
7451
7452 \newenvironment{ledgroup}{%
7453   \resetprevpage@num%
7454   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
7455   \let\@footnotetext\@mpfootnotetext
7456   \l@dfetbeginmini%
7457 }{%
7458   \par
7459   \unskip
7460   \ifvoid\@mpfootins\else
7461     \l@dunboxmpfoot
7462   \fi
7463   \l@dfetendmini%
7464   \@ledgroupfalse%
7465 }
7466
7467
7468 %
```

ledgroupsize $\begin{bmatrix} \text{ledgroupsize} \end{bmatrix} [\langle pos \rangle] \{ \langle width \rangle \}$

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.

```
7469 \newenvironment{ledgroupsize}[2][1]{%
7470 %
```

Set the various text measures.

```
7471 \hsize #2\relax
7472 %
```

Initialize fills for centering.

```
7473 \let\ledllfill\hfil
7474 \let\ledrlfill\hfil
7475 \def\@tempa{#1}\def\@tempb{1}%
7476 %
```

Left adjusted numbered lines

```
7477 \ifx\@tempa\@tempb
7478 \let\ledllfill\relax
7479 \else
7480 \def\@tempb{r}%
7481 \ifx\@tempa\@tempb
7482 %
```

Right adjusted numbered lines

```
7483 \let\ledrlfill\relax
7484 \fi
7485 \fi
7486 %
```

Set up the footnoting.

```
7487 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
7488 \let\@footnotetext\@mpfootnotetext
7489 \l@dfeetbeginmini%
7490 }{f%
7491 \par
7492 \unskip
7493 \ifvoid\@mpfootins\else
7494 \l@dunboxmpfoot
7495 \fi
7496 \l@dfeetendmini%
7497 }
7498
7499 %
```

Close the \ifnoledgroup@\else.

```
7500 \fi%
7501 %
```

`\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`.

```
7502 \newif\ifledgroupnotesL@
7503 \newif\ifledgroupnotesR@
7504 %
```

XXVIII Indexing

Here is some code for indexing using page and line numbers.

XXVIII.1 Looking on package order

First, ensure that `imakeidx` or `indextools` is loaded *before* `eledmac`.

```
7505 \AtBeginDocument{%
7506 \unless\ifl@imakeidx%
7507 \@ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
7508 %
7509 \fi%
7510 \unless\ifl@indextools%
7511 \@ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}%
7512 }
```



```

7511 \fi%
7512 \unless\ifl@footmisc%
7513 \@ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
7514 %
7515 \fi%
7516 }
7517 %

```

XXVIII.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
7517 \newcommand{\pagelinesep}{-}
7518 \newcommand{\edindexlab}{$&}
7519 \newcounter{labidx}
7520 \setcounter{labidx}{0}
7521
7522 %

```

`\doedindexlabel` This macro sets an `\edlabel`.

```

7523 \newcommand{\doedindexlabel}{%
7524 \stepcounter{labidx}%
7525 \edlabel{\edindexlab\thelabidx}%
7526 }
7527
7528 %

```

`\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`.

```

7529 \newcounter{pageline}%
7530 \renewcommand{\thepageline}{%
7531 \thepage%
7532 \pagelinesep%
7533 \xlineref{\edindexlab\thelabidx}%
7534 }
7535 %

```

`\thestartpageline` These macros make up the page/line start/end number when the `\edindex` command is called in critical notes.

```

\theendpageline
7536 \newcommand{\thestartpageline}{%
7537 \l@dparsedstartpage%
7538 \pagelinesep%
7539 \l@dparsedstartline%
7540 }

```

```

7541 \newcommand{\theendpageline}{%
7542   \l@dparsedendpage%
7543   \pagelinesep%
7544   \l@dparsedendline%
7545 }
7546 %

```

XXVIII.3 Code specific to \edindex in critical footnotes

\@edindex@fornote@ This counter is incremented at the beginning of each note (either a footnote or a sidenote), and decremented at the end of each note. If its value is greater than 0, that means we are inside a note.

```

7547 \newcount\@edindex@fornote@
7548 %

```

\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 |.

```

7549 \newcommand{\prepare@edindex@fornote}[1]{%
7550   \l@dp@rsefootspec#1|%
7551   \advance\@edindex@fornote@\@ne%
7552 }
7553 %

```

\get@edindex@ledinnote@command The \get@edindex@ledinnote@command macro defines a \@ledinnote@command command which is added as an attribute (text inserted after |) of the next index entry.

Consequently, we write the definition of the location reference attribute in the .xdy file.

```

7554 \newcommand{\get@edindex@ledinnote@command}{%
7555   \ifxindy@%
7556     \gdef\@ledinnote@command{%
7557       ledinnote\thelabidx%
7558     }%
7559     \ifxindyhyperref@%
7560       \immediate\write\eledmac@xindy@out{%
7561         (define-attributes ("ledinnote\thelabidx"))^^J
7562         \space\space(markup-locref^^J
7563         \eledmacmarkuplocrefdepth^^J
7564         :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
7565       }{"^^J
7566         :close "}"^^J
7567         :attr "ledinnote\thelabidx"^^J
7568       )
7569     }%
7570   \else%

```

```

7570 \immediate\write\eledmac@xindy@out{%
7571 (define-attributes ("ledinnote\thelabidx"))^^J
7572 \space\space(markup-locref^^J
7573 \eledmacmarkuplocdepth^^J
7574 :open "\string\ledinnote{\@index@command}{"}^^J
7575 :close "}"^^J
7576 :attr "ledinnote\thelabidx"^^J
7577 )
7578 }%
7579 \fi%
7580 %

```

If we do not use xindy option, \@ledinnote@command will produce something like ledinnote{formattingcommand}.

```

7581 \else%
7582 \gdef\@ledinnote@command{%
7583 ledinnote[\edindexlab\thelabidx]{\@index@command}%
7584 }%
7585 \fi%
7586 }
7587 %

```

XXVIII.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 |.

```

7588 \def\get@index@command#1|#2+{%
7589 \gdef\@index@txt{#1}%
7590 \gdef\@index@command{#2}%
7591 \xdef\@index@parenthesis{%
7592 \IfBeginWith{\@index@command}{(}{%
7593 \StrGobbleLeft{\@index@command}{1}{\@index@command}%
7594 \global\let\@index@command\@index@command%
7595 \xdef\@index@parenthesis{%
7596 }{}}%
7597 \IfBeginWith{\@index@command}{)}{%
7598 \StrGobbleLeft{\@index@command}{1}{\@index@command}%
7599 \global\let\@index@command\@index@command%
7600 \xdef\@index@parenthesis{)}}%
7601 }{}}%
7602 }
7603 %

```

XXVIII.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

```

7604 \newcommandx{\ledinnote}[3][1,usedefault]{%
7605   \ifboolexpr{%
7606     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
7607     or%
7608     bool {xindyhyperref@}%
7609   }%
7610   {%
7611     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
7612   }%
7613   {%
7614     \csuse{#2}{\ledinnotemark{#3}}%
7615   }%
7616 }%
7617 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage
7618 {#2}}}}%
7619 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
7620 %

```

XXVIII.6 Main code

Eledmac and ledmac were using the specific indexing tools of the memoir in order to allow multiple index. However, elledmac 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
\dummy@edindex
7620 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
7621   \begin{group}%
7622     \let\emph\@firstofone%
7623     \let\textbf\@firstofone%
7624     \let\textit\@firstofone%
7625     \let\textmd\@firstofone%
7626     \let\textnormal\@firstofone%
7627     \let\textrm\@firstofone%
7628     \let\textsc\@firstofone%
7629     \let\textsf\@firstofone%
7630     \let\textsl\@firstofone%
7631     \let\texttt\@firstofone%
7632     \let\textup\@firstofone%
7633     \xdef\@tmp{#2}%To be used in IfSubStr instead of #2 directly. Avoid
some expansion bugs (for example with \edindex{textsc{something}})
7634   \end{group}%
7635   \ifl@imakeidx%

```

```

7636 \ifnum\@edindex@fornote@>\z@%
7637 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
7638 \get@edindex@ledinnote@command%
7639 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
7640 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%
7641 \else%
7642 \get@edindex@hyperref{#2}%
7643 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
7644 \fi%
7645 \else%
7646 \ifnum\@edindex@fornote@>\z@%
7647 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
7648 \get@edindex@ledinnote@command%
7649 \expandafter\protected@write\@indexfile{}%
7650 {\string\indexentry{\@index@txt|(\@ledinnote@command}{\thestartpageline}
}%
7651 }%
7652 \expandafter\protected@write\@indexfile{}%
7653 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
}%
7654 }%
7655 \else%
7656 \protected@write\@indexfile{}%
7657 {\string\indexentry{#2}{\thepageline}
}%
7658 }%
7659 \fi%
7660 \fi%
7661 \endgroup
7662 \@esphack%
7663 }
7664 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

7665 \pretocmd{\makeindex}{%
7666 \def\edindex{%
7667 \ifboolexpr{bool{numbering} or bool{numberingR} or bool{
1@dprintingpages} or bool{1@dprintingcolumns}}{%
7668 \@bsphack%
7669 \doedindexlabel%
7670 \begingroup%
7671 \@sanitize%
7672 \@wredindex%
7673 }%
7674 {%
7675 \led@warn@edinde@outsidenumbering%
7676 \index%
7677 }%

```

```

7678 }%
7679 }%
7680 {}%
7681 {\led@error@fail@patch@makeindex}%
7682 \newcommand{\edindex}[1]{\@bsphack\@esphack}
7683 \newcommandx{\dummy@edindex}[2][1=\expandonce\jobname,usedefault]{}%
7684 %

```

XXVIII.7 Hyperlink

\hyperlinkformat \hyperlinkformat command is to be used to have both a internal hyperlink and a format, when indexing.

```

7685 \newcommand{\hyperlinkformat}[3]{%
7686   \ifstrempy{#1}%
7687     {\hyperlink{#2}{#3}}%
7688     {\csuse{#1}{\hyperlink{#2}{#3}}}%
7689 }%
7690 %

```

\hyperlinkR \hyperlinkR command is to be used to create a internal hyperlink and \ledRflag, when indexing.

```

7691 \newcommand{\hyperlinkR}[2]{%
7692   \hyperlink{#1}{#2\@Rlineflag}%
7693 }%
7694 %
7695 %

```

\hyperlinkformatR \hyperlinkformatR command is to be used to create a internal hyperlink, a format and a \@Rlineflag, when indexing.

```

7696 \newcommand{\hyperlinkformatR}[3]{%
7697   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
7698 }%
7699 %
7700 %

```

\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).

```

7701 \newcommand{\get@edindex@hyperref}[1]{%
7702 %

```

We have to disable temporary spaces to work through a xstring bug (or feature?)

```

7703 \edef\temp@{%
7704   \catcode`\ =9 %space need for catcode
7705   \detokenize{#1}%For active character in unicode
7706   \catcode`\ =10 % space need for catcode
7707 }%
7708 %

```

Now, we define `\@edindex@hyperref` if the `hyperindex` of `hyperref` is enabled.

```

7709 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
7710   \IfSubStr{\temp@}{|}%
7711     {\get@index@command#1+%
7712       \ifledRcol%
7713         \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7714           hyperlinkformatR{\@index@command}%
7715             {\edindexlab\thelabidx}}%
7716       \else%
7717         \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7718           hyperlinkformat{\@index@command}%
7719             {\edindexlab\thelabidx}}%
7720       \fi%
7721     }%
7722     {\get@index@command#1|+%
7723       \ifledRcol%
7724         \gdef\@edindex@hyperref{|\hyperlinkR{\edindexlab\thelabidx}}%
7725       \else%
7726         \gdef\@edindex@hyperref{|\hyperlink{\edindexlab\thelabidx}}%
7727       \fi%
7728     }%
7729   }%
7730 }%

7731 % If we use both xindy and hyperref, first get the \protect\cs{
7732   index@command} command.
7733 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
7734 % \begin{macrocode}
7735 {\ifxindyhyperref%
7736   \IfSubStr{\temp@}{|}%
7737     {\get@index@command#1+%
7738       {\get@index@command#1|+%
7739         \gdef\@edindex@hyperref{|\eledmac\thelabidx}%
7740       }%
7741     }%
7742   }%
7743 }%
7744 }%
7745 }%
7746 %

```

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`.

```

7740 \IfStrEq{\@index@parenthesis}{(%}%
7741   {%
7742     \csxdef{xindy@parenthesis@{\@index@txt}{\thelabidx}%
7743       \gdef\@edindex@hyperref{|(eledmac\thelabidx}%
7744     }%
7745   }%
7746 }%

```

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`.

```

7747 \IfStrEq{\@index@parenthesis}{})}%
7748 {%
7749 \xdef\@edindex@hyperref{|)\eledmac\csuse{xindy@parenthesis@
@index@txt}}}%
7750 \global\csundef{xindy@parenthesis@ \@index@txt}}%
7751 }%
7752 %

```

Write in the .xdy file the attributes of the location.

```

7753 {%
7754 \immediate\write\eledmac@xindy@out{%
7755 (define-attributes ("eledmac\thelabidx"))^^J
7756 \space\space(markup-locoref^^J
7757 \eledmacmarkuplocorefdepth^^J
7758 :open "\string\hyperlink%
7759 \ifledRcol R\fi%
7760 {\edindexlab\thelabidx}%
7761 {\ifdefempty{\@index@command}}%
7762 {}%
7763 {\@backslashchar \@index@command}}%
7764 {"^^J
7765 :close "}"^^J
7766 :attr "eledmac\thelabidx"^^J
7767 )
7768 }%
7769 }%
7770 %

```

And now, in any other case.

```

7771 \else%
7772 \gdef \@index@txt{#1}%
7773 \gdef \@edindex@hyperref{}%
7774 \fi%
7775 }%
7776 }
7777 %

```

XXVIII.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.

```

7778 \newcommand{\led@set@index@fornote}[1]{%
7779 \ifbool{indtl@innote}%
7780 {\let\index\nindex}%
7781 {}%

```



```

7782 \ifbool{indtl@notenumber}%
7783 {%
7784 \renewcommand{\index}[2][\indtl@jobname]{%
7785 \orig@@index[##1]{%
7786 ##2|innotenumber{\this@footnoteX@reading}%
7787 }%
7788 }%
7789 }%
7790 {}%
7791 }%
7792 %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

7793 \newcommand{\led@reinit@index@fornote}{%
7794 \ifbool{indtl@innote}%
7795 {\let\index\orig@@index}%
7796 {}%
7797 \ifbool{indtl@notenumber}%
7798 {\let\index\orig@@index}%
7799 {}%
7800 }%
7801 %

```

XXIX Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

7802 \AtBeginDocument{%
7803 \@ifpackageloaded{glossaries}{%
7804 %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

7805 \renewcommand{\do}[1]{%
7806 \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
7807 \doedindexlabel%
7808 \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
7809 }%
7810 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,3,usedefault]{%
7811 \doedindexlabel%
7812 \csname#1\endcsname*[counter=pageline,##1]{##2}[##3]%
7813 }%
7814 }%
7815 \docsvlist{%
7816 gls,%
7817 Gls,%

```

```

7818 GLS,%
7819 glspl,%
7820 Glspl,%
7821 GLSpl,%
7822 glstext,%
7823 Glstext,%
7824 GLStext,%
7825 Glsfirst,%
7826 GLSfirst,%
7827 glsplural%
7828 Glsplural,%
7829 GLSplural,%
7830 glsfirstplural,%
7831 Glsfirstplural,%
7832 GLSfirstplural,%
7833 glsname,%
7834 Glsname,%
7835 GLSname,%
7836 glssymbol,%
7837 Glssymbol,%
7838 GLSsymbol,%
7839 glsdesc,%
7840 Glsdesc,%
7841 GLSdesc,%
7842 glsuseri,%
7843 Glsuseri,%
7844 GLSuseri,%
7845 glsuserii,%
7846 Glsuserii,%
7847 GLSuserii,%
7848 glsuseriii,%
7849 Glsuseriii,%
7850 GLSuseriii,%
7851 glsuseriv,%
7852 Glsuseriv,%
7853 GLSuseriv,%
7854 glsuserv,%
7855 Glsuserv,%
7856 GLSuserv,%
7857 glsuservi,%
7858 Glsuservi,%
7859 GLSuservi%
7860 }%
7861 %

```

Then those which arguments are [*options*]{*label*}{*link text*}.

```

7862 \renewcommand{\do}[1]{%
7863 \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,
usedefault]{%
7864 \doedindexlabel%

```

```

7865 \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
7866 }%
7867 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,usedefault]{%
7868 \doedindexlabel%
7869 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
7870 }%
7871 }%
7872 \docsvlist{glsdisp,glslink}%
7873 %

```

Then those which arguments are [*options*]{*label*}.

```

7874 \renewcommand{\do}[1]{%
7875 \expandafter\DeclareRobustCommand\csname ed#1\endcsname[2][1,
usedefault]{%
7876 \doedindexlabel%
7877 \csname#1\endcsname[counter=pageline,##1]{##2}%
7878 }%
7879 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[2][1,usedefault]{%
7880 \doedindexlabel%
7881 \csname#1\endcsname*[counter=pageline,##1]{##2}%
7882 }%
7883 }%
7884 \docsvlist{glsadd}%
7885 }{}%
7886 }%
7887 %

```

XXX 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`.

XXX.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 typography, the symbol is '['. We obtain it by means of the following code:

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

The `\ifinstanza` boolean is used to ensure that we are in a stanza part.

```

7888 \def\@hangingsymbol{}
7889 \newcommand*\sethangingsymbol[1]{%
7890 \gdef\@hangingsymbol{#1}%

```

```

7891 }%
7892 \newif\ifinstanza
7893 %

```

`\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.

```

7894 \newif\ifinserthangingsymbol
7895 \newcommand{\inserthangingsymbol}{%
7896 \ifinserthangingsymbol%
7897   \ifinstanza%
7898     \@hangingsymbol%
7899   \fi%
7900 \fi%
7901 }
7902 %

```

XXX.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.

```

7903 \newcommand*{\ampersand}{\char`\&}
7904
7905 %

```

XXX.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.

```

7906 \chardef\body=\catcode`\@
7907 \catcode`\@=11
7908 \chardef\next=\catcode`\&
7909 \catcode`\&=\active
7910
7911 %

```

XXX.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.

```

7912 \newcount\stanza@count
7913 \newlength{\stanzaindentbase}
7914 \setlength{\stanzaindentbase}{20pt}
7915
7916 %

```

`\strip@szacnt` 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.

```

7917 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
7918 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,,|}%
7919   \stanza@count\z@
7920   \def\next{\expandafter\strip@szacnt\@tempa
7921     \ifx\@tempb\empty\let\next\relax\else
7922       \expandafter\mathchardef\csname #1@\number\stanza@count
7923       \@endcsname\@tempb\relax
7924       \advance\stanza@count\@ne\fi\next}%
7925   \next}
7926
7927 %

```

`\setstanzaindents` 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).

```

7928 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
7929 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
7930 %
7931 %

```

`\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.

```

7932 \newcounter{stanzaindentsrepetition}
7933 \newcount\stanza@modulo
7934
7935 \newcommand*\managestanza@modulo[0]{%
7936   \advance\stanza@modulo\@ne%
7937   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
7938     \stanza@modulo\@ne%
7939   \fi%

```

```
7940 }
7941 %
```

\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.

```
7942 \newcommand{\stanzaindent}[1]{%
7943   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
7944   \ignorespaces%
7945 }%
7946 \WithSuffix\newcommand\stanzaindent*[1]{%
7947   \stanzaindent{#1}%
7948   \global\advance\stanza@modulo-\@ne%
7949   \ifnum\stanza@modulo=0%
7950     \global\stanza@modulo=\value{stanzaidentsrepetition}%
7951   \fi%
7952   \ignorespaces%
7953 }%
7954 %
```

XXX.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```
\thestanza55 \newcounter{stanza}
7956 \renewcommand{\thestanza}{%
7957   \textbf{\arabic{stanza}}%
7958 }
7959 %
```

\ifnumberstanza Then, macro to activate automatically numbering of stanza.

```
7960 \newif\ifnumberstanza%
7961 %
```

\@insertstanzanumber Now, macro called at the first line of of verse of a stanza.

```
7962 \newcommand{\@insertstanzanumber}[0]{%
7963   \ifnumberstanza%
7964     \ifl@dpairing%
7965       \ifledRcol%
7966         \stanzanumwrapper{\thestanzaR}%
7967       \else%
7968         \stanzanumwrapper{\thestanzaL}%
7969       \fi%
7970     \else%
7971       \stanzanumwrapper{\thestanza}%
7972     \fi%
```

```

7973 \setline{1}%
7974 \fi%
7975 }%
7976 %

```

`\@advancestanzanumber` Also a command to advance the counter of stanza.

```

7977 \newcommand{\@advancestanzanumber}[0]{%
7978 \ifnumberstanza%
7979 \ifl@dpairing%
7980 \ifledRcol%
7981 \addtocounter{stanzaR}{1}%
7982 \else%
7983 \addtocounter{stanzaL}{1}%
7984 \fi%
7985 \else%
7986 \addtocounter{stanza}{1}%
7987 \fi%
7988 \fi%
7989 }%
7990 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

7991 \newcommand{\stanzanumwrapper}[1]{%
7992 \flagstanza{#1}%
7993 }%
7994 %

```

XXX.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

7995 \newcommand{\printstanza}[0]{%
7996 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
7997 l@dprintingcolumns}}{%
7998 \ifledRcol%
7999 \thestanzaR%
8000 \else%
8001 \thestanzaL%
8002 \fi%
8003 }{%
8004 \thestanza%
8005 }%
8006 }%

```

XXX.7 Main work

`\stanza@line` Now we arrive at the main works. `\stanza@line` sets the indentation for the line and starts a numbered paragraph—each line is treated as a paragraph. `\stanza@hang` sets the hanging indentation to be used if the stanza line requires more than one print line.

If it is known that each stanza line will fit on one print line, it is advisable to set the hanging indentation to zero. `\sza@penalty` places the specified penalty following each stanza line. By default, this facility is turned off so that no penalty is included. However, the user may initiate these penalties to indicate good and bad places in the stanza for page breaking.

```

8007 \newcommand{\stanza@line}[2][1,2,usedefault]{%
8008     \ifnum\value{stanzaindentrepetition}=0
8009         \ifcsdef{sza@\number\stanza@count @}{%
8010             {%
8011                 \parindent=\csname sza@\number\stanza@count @\endcsname\
stanzaindentbase%
8012             }{%
8013                 \led@err@StanzaIndentNotDefined%
8014             }%
8015         }else
8016             \ifcsdef{sza@\number\stanza@modulo @}{%
8017                 \parindent=\csname sza@\number\stanza@modulo @\endcsname\
stanzaindentbase%
8018                 \managestanza@modulo%
8019             }%
8020             {%
8021                 \led@err@StanzaIndentNotDefined%
8022             }%
8023         \fi
8024         \pstart[#1][#2]\stanza@hang\ignorespaces%
8025     }%
8026 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
8027     \hangindent\expandafter
8028     \noexpand\csname sza@0@\endcsname\stanzaindentbase
8029     \hangafter\@ne}
8030 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
8031     \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
8032     \penalty\fi\count@}
8033 %

```

`\@startstanza` Now we have the components of the `\stanza` macro, which appears at the start of a group of lines. This macro initializes the count and checks to see if hanging indentation and penalties are to be included. Hanging indentation suspends the line count, so that the enumeration is by verse line rather than by print line. If the print line count is desired, invoke `\let\startlock\relax` and do the same for `\endlock`. Here and above we have used `\xdef` to make the stored macros take up a bit less space, but it also makes them more obscure to the reader. Lines of the stanza are delimited by ampersands `&`. The last line of the stanza must end with `\&`.

`\stanza`

`\@stopstanza`

`\AtEveryStopStanza`

`\AtEveryStanza`

`\AtStartEveryStanza`

`\BeforeEveryStopStanza`

`\newverse`


```

8034 \xdef\@startstanza[#1][#2]{%
8035   \noexpand\instanzatrue\expandafter
8036   \begingroup%
8037   \catcode`\noexpand\&\active%
8038   \global\stanza@count\@ne\stanza@modulo\@ne
8039   \noexpand\ifnum\expandafter\noexpand
8040   \csname sz@0@\endcsname=z@\let\noexpand\stanza@hang\relax
8041   \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
8042   \@M\rightskipz@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
8043   \expandafter\noexpand\csname szp@0@\endcsname=z@
8044   \let\noexpand\sza@penalty\relax\noexpand\fi%
8045   \def\noexpand&{%
8046     \noexpand\newverse[] []}%
8047   \def\noexpand\&{\noexpand\@stopstanza}%
8048   \noexpand\@advancestanzanumber%
8049   \noexpand\stanza@line[#1][#2]%
8050   \noexpand\@insertstanzanumber%
8051   \let\par\relax\ignorespaces%No paragraph in verses
8052 }
8053
8054 \newcommandx{\stanza}[2][1,2,usedefault]{%
8055   \ifboolexpr{%
8056     not test{\ifdefvoid{\at@every@stanza}}%
8057     and test{\ifstrempy{#1}}%
8058     and test{\ifstrempy{#2}}}%
8059   {\@startstanza[] [\at@every@stanza]\at@start@every@stanza}%
8060   {\@startstanza[#1][#2]\at@start@every@stanza}%
8061 }%
8062
8063 \newcommandx{\@stopstanza}[2][1,2,usedefault]{%
8064   \unskip%
8065   \endlock%
8066   \ifboolexpr{%
8067     not test{\ifdefvoid{\at@every@stop@stanza}}%
8068     and test{\ifstrempy{#1}}%
8069     and test{\ifstrempy{#2}}}%
8070   {\before@every@stop@stanza\pend[] [\at@every@stop@stanza]}%
8071   {\before@every@stop@stanza\pend[#1][#2]}%
8072   \endgroup%
8073   \instanzafalse%
8074 }
8075
8076 \newcommand{\AtEveryStopStanza}[1]{%
8077   \ifstrempy{#1}%
8078   {\gdef\at@every@stop@stanza{}}%
8079   {\gdef\at@every@stop@stanza{\noindent#1}}%
8080 }%
8081 \WithSuffix\newcommand\AtEveryStopStanza*[1]{%
8082   \ifstrempy{#1}%
8083   {\gdef\at@every@stop@stanza{}}%

```

```

8084     {\gdef\at@every@stop@stanza{#1}}%
8085 }%
8086 \def\at@every@stop@stanza{}%
8087
8088 \newcommand{\AtEveryStanza}[1]{%
8089   \ifstrepty{#1}%
8090     {\gdef\at@every@stanza{}}%
8091     {\gdef\at@every@stanza{\noindent#1}}%
8092 }%
8093 \WithSuffix\newcommand\AtEveryStanza*[1]{%
8094   \ifstrepty{#1}%
8095     {\gdef\at@every@stanza{}}%
8096     {\gdef\at@every@stanza{#1}}%
8097 }%
8098
8099
8100
8101 \newcommand{\AtStartEveryStanza}[1]{%
8102   \ifstrepty{#1}%
8103     {\gdef\at@start@every@stanza{}}%
8104     {\gdef\at@start@every@stanza{#1}}%
8105 }%
8106 \def\at@start@every@stanza{}%
8107
8108 \newcommand{\BeforeEveryStopStanza}[1]{%
8109   \ifstrepty{#1}%
8110     {\gdef\before@every@stop@stanza{}}%
8111     {\gdef\before@every@stop@stanza{#1}}%
8112 }%
8113 \def\before@every@stop@stanza{}%
8114
8115 \newcommandx*\newverse}[4][1,2,3,4,usedefault]{%
8116   \unskip%
8117   \endlock\pend[#1][#3]\sza@penalty\global%
8118   \advance\stanza@count\@ne\stanza@line[#2][#4]%
8119 }
8120
8121 %

```

\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`.

```

8122 \newcommand*\flagstanza}[2][\stanzaindentbase]{%
8123   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
8124
8125 %

```

XXX.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.

```

8126 \catcode`\&=\next
8127 \catcode`\@=\body
8128 \setstanzavalues{szp}{0}
8129
8130 %

```

XXXI Apparatus of Manuscripts

XXXI.1 User level macro

\msdata The user level \msdata command only writes the manuscripts data in a numbered auxiliary file. There are two associated etoolbox counters.

```

\msdata@cR
\msdata@c
\msdata@cR
8131 \def\msdata@c{}%
8132 \def\msdata@cR{}%
8133 \newcommand{\msdata}[1]{%
8134   \leavevmode%
8135   \unless\ifstopmsdata@inserted@%
8136     \stopmsdata%
8137     \led@warning@msdatawithoutstop%
8138   \fi%
8139   \global\stopmsdata@inserted@false%
8140   \unless\ifledRcol%
8141     \numgdef{\msdata@c}{\msdata@c+1}%
8142     \ifdef{\hypertarget}{%
8143       \edlabel{\msdata@c:start:msdata}%
8144     }{}%
8145     \protected@write\linenum@out{}{%
8146       \string\@msd{#1}%
8147     }%
8148   \else%
8149     \numgdef{\msdata@cR}{\msdata@cR+1}%
8150     \ifdef{\hypertarget}{%
8151       \edlabel{\msdata@cR:start:msdata}%
8152     }{}%
8153     \protected@write\linenum@outR{}{%
8154       \string\@msd{#1}%
8155     }%
8156   \fi%
8157 }%

```

8158 %

\stopmsdata The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```
8159 \newcommand{\stopmsdata}[0]{%
8160   \leavevmode%
8161   \unless\ifledRcol%
8162     \protected@write\linenum@out{}{%
8163       \string\stopmsd%
8164     }%
8165     \ifdef{\hypertarget}{%
8166       \edlabel{\msdata@c:end:msdata}%
8167     }{}%
8168   \else%
8169     \protected@write\linenum@outR{}{%
8170       \string\stopmsd%
8171     }%
8172     \ifdef{\hypertarget}{%
8173       \edlabel{\msdata@cR:end:msdata}%
8174     }{}%
8175   \fi%
8176   \global\stopmsdata@inserted@true%
8177 }%
8178 %
```

\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`

```
8179 \newif\ifstopmsdata@inserted@%
8180 %
```

XXXI.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?

```
8181 \newcommand{\setmsdataseries}[1]{%
8182   \gdef\@msdata@series{#1}%
8183 }%
8184 \def\@msdata@series{A}%
8185 %
```

\setmsdataposition The label for the manuscripts data.

```

8186 \def\ms@data@position{msdata-regular}%
8187 \newcommand{\setmsdataposition}[1]{%
8188   \gdef\ms@data@position{#1}%
8189 }%
8190 %

```

\setmsdatalabel The label for the manuscripts data.

```

8191 \def\ms@data@label{Ms.}%
8192 \newcommand{\setmsdatalabel}[1]{%
8193   \gdef\ms@data@label{#1}%
8194 }%
8195 %

```

XXXI.3 Counters and lists

\@msd@c \@msd@c is a counter incremented at each \@msd read in auxiliary file.

```

8196 \numdef{\@msd@c}{0}
8197 \numdef{\@msd@cR}{0}
8198 %

```

\add@msd@ \add@msd@ is a counter incremented at each \add@msddata, that is at each time we prepare the insertion of manuscripts data footnote.

```

8199 \numdef{\add@msd@c}{0}%
8200 \numdef{\add@msd@cR}{0}%
8201 %

```

\@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.

```

8202 \def\@msdata@list{}%
8203 %

```

XXXI.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.

```

8204 \newcommand{\@msd}[1]{%
8205   \unless\ifledRcol%
8206     \global\numdef{\@msd@c}{\@msd@c+\@ne}%
8207     \csgdef{\@msdata@\@msd@c @data}{#1}%
8208     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
8209     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%

```

```

8210 \xright@appenditem{\the\absline@num}\to\actionlines@list%
8211 \xright@appenditem{-1010}\to\actions@list%
8212 \else%
8213 \global\newdef{\@msd@cR}{\@msd@cR+\@ne}%
8214 \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
8215 \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
8216 \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
8217 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
8218 \xright@appenditem{-1010}\to\actions@listR%
8219 \fi%
8220 }%
8221 %

```

\@stopmsd Inserted in the auxiliary file by \stopmsd, the \@stopmsd macro will store in two commands the line number and the absolute line number on which it is called.

```

8222 \newcommand{\@stopmsd}[0]{%
8223 \unless\ifledRcol%
8224 \ifcsundef{\@msdata@\@msd@c @stoplinenumber}{%
8225 \csxdef{\@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
8226 \csxdef{\@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
8227 }%
8228 \else%
8229 \ifcsundef{\@msdata@\@msd@cR @stoplinenumberR}{%
8230 \csxdef{\@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
8231 \csxdef{\@msdata@\@msd@cR @stoplinenumberR}{\the\line@numR}%
8232 }%
8233 }%
8234 \fi%
8235 }%
8236 %

```

XXXI.5 Action macro

\add@msdata \add@msdata is executed on each line when action code 1010 is seen. It will not insert immediately the manuscript data footnote, as action code are executed before the line be typeset, and, consequently, could be on the previous page. So it just stores the manuscript data footnote to \@msdata@list.

```

8237 \newcommand{\add@msdata}{%
8238 \bgroup%
8239 \normalfont%
8240 \unless\ifledRcol%
8241 \newdef{\add@msd@c}{\add@msd@c+\@ne}%
8242 \ifcsdef{\@msdata@\add@msd@c @data}{%
8243 \letcs{\@data}{\@msdata@\add@msd@c @data}%
8244 \edef\l@d@nums{%
8245 000| Start page = we don't print it
8246 \csuse{\@msdata@\add@msd@c @linenumber}| Start line number

```

```

8247         000|% Start subline number, for now, not used
8248         000|% End page number, we don't print it
8249         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}%
8250         {\csuse{@msdata@\add@msd@c @stoplinenumber}}%End line number if
in the same page
8251         {\csuse{@lastline@forpage@\the\page@num}}%Otherwiser, last
number of the page
8252         |%
8253         000|% End sub line number, for now, not used
8254         \edfont@info%Font
8255         }%
8256         \@msd@options@fullpagefalse%
8257         \if@firstlineofpage%Try if the data are for the full page. If yes
, will add options to the list.
8258         \unless\if@msdata@insertedfrompreviouspage%
8259         \ifnumless{\csuse{@lastabsline@forpage@\the\page@num}}{\csuse
{@msdata@\add@msd@c @stopabslinenumber}+\@ne}%
8260         {%
8261         \numdef{\@tmp}{\add@msd@c+\@ne}%
8262         \ifcsdef{@msdata@\@tmp @abslinenumber}%
8263         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
8264         {}%
8265         {\@msd@options@fullpagetrue}%
8266         }%
8267         {\@msd@options@fullpagetrue}%
8268         }%
8269         {}%
8270         \fi%
8271         \fi%
8272         \listxadd{\@msdata@list}{%
8273         \@msd@options@iffullpage%
8274         \ifluatex%
8275         \csxdef{footnote@luatextextdir}{\the\textdir}%
8276         \csxdef{footnote@luatexpardir}{\the\pardir}%
8277         \fi%
8278         \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
8279         \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
8280         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}}%
8281         \reset@msd@options@iffullpage%
8282         }%
8283         }%
8284         {}%
8285         \else%
8286         \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
8287         \ifcsdef{@msdata@\add@msd@cR @dataR}{%
8288         \letcs{\@data}{@msdata@\add@msd@cR @dataR}%
8289         \edef\l@d@nums{%

```

```

8290      000| % Start page = we don't print it
8291      \csuse{@msdata@add@msd@cR @linenumberR}| % Start line number
8292      000| % Start subline number, for now, not used
8293      000| % End page number, we don't print it
8294      \ifnumless{\csuse{@msdata@add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}%
8295      {\csuse{@msdata@add@msd@cR @stoplinenumberR}}%End line number
if in the same page
8296      {\csuse{@lastline@forpageR@the\page@numR}}%Otherwiser, last
number of the page
8297      |%
8298      000| % End sub line number, for now, not used
8299      \edfont@info%Font
8300      }%
8301      \@msd@options@fullpagefalse%
8302      \if@firstlineofpageR%
8303      \unless\if@msdata@insertedfrompreviouspage%
8304      \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
csuse{@msdata@add@msd@c @stopabslinenumberR}+\@one}%
8305      {%
8306      \numdef{\@tmp}{\add@msd@cR+\@one}%
8307      \ifcsdef{@msdata@ \@tmp @abslinenumberR}%
8308      {\ifnumequal{\csuse{@msdata@ \@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
8309      {}%
8310      {\@msd@options@fullpagetrue}%
8311      }%
8312      {\@msd@options@fullpagetrue}%
8313      }%
8314      {}%
8315      \fi%
8316      \fi%
8317      \listxadd{\@msdata@list}{%
8318      \@msd@options@iffullpage%
8319      \ifluatex%
8320      \csxdef{footnote@luatextextdir}{\the\textdir}%
8321      \csxdef{footnote@luatexpardir}{\the\pardir}%
8322      \fi%
8323      \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
8324      \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
8325      \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
8326      \reset@msd@options@iffullpage%
8327      }%
8328      }%
8329      {}%
8330      \fi%
8331      \egroup%
8332      }%
8333      %

```


`\insertedfrompreviouspage` The `\ifmsdata@insertedfrompreviouspage` boolean is set to TRUE if `reledmac` automatically inserts data from previous page in the first line of a page.

```
8334 \newif\ifmsdata@insertedfrompreviouspage%
8335 %
```

`\addmsdata@firstlineofpage` `\addmsdata@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.

```
8336 \newcommand{\addmsdata@firstlineofpage}{%
8337   \bgroup%
8338   \normalfont%
8339   \unless\ifledRcol{%
8340     \ifcsdef{msdata@addmsd@c @data}{%
8341       \ifnumless{\the\absline@num-\@ne}{\csuse{msdata@addmsd@c
8342         @stopabslinenumber}}}%
8343       {%
8344         \global\msdata@insertedfrompreviouspagetrue%
8345         \letcs{\@data}{msdata@addmsd@c @data}%
8346         \edef\l@d@nums{%
8347           000|% Start page = we don't print it
8348           \numexpr\the\line@num+\@ne\relax|% Start line number = first line
8349           of the page. As \addmsdata@firstlineofpage is called before line number
8350           has been incremented, we increment it for printing
8351           000|% Start subline number, for now, not used
8352           000|% End page number, we don't print it
8353           \ifnumless{\csuse{msdata@addmsd@c @stopabslinenumber}}{\csuse{
8354             @lastabsline@forpage@the\page@num}}{\csuse{
8355               \csuse{msdata@addmsd@c @stopabslinenumber}}}%End line number if
8356               in the same page
8357               {\csuse{lastline@forpage@the\page@num}}}%Otherwise, last
8358               number of the page
8359               |%
8360               000|% End sub line number, for now, not used
8361               \edeffont@info{Font
8362               }%
8363               \msd@options@fullpagefalse%
8364               \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse{
8365                 msdata@addmsd@c @stopabslinenumber}+\@ne}%We will test if the ms data is
8366                 for the full page
8367                 {%
8368                   \numdef{\@tmp}{\addmsd@c+\@ne}%
8369                   \ifcsdef{msdata@\@tmp @abslinenumber}%
8370                   {\ifnumequal{\csuse{msdata@\@tmp @abslinenumber}}{\csuse{
8371                     @lastabsline@forpage@the\page@num}}}%
8372                   {}%
8373                   {\msd@options@fullpagetrue}%
8374                   }%
8375                   {\msd@options@fullpagetrue}%
8376                   }%
8377                 }
```

```

8368 {}%
8369 \listxadd{\@msdata@list}{%
8370 \@msd@options@iffullpage%
8371 \ifluatex%
8372 \csxdef{footnote@luatextextdir}{\the\textdir}%
8373 \csxdef{footnote@luatexpardir}{\the\pardir}%
8374 \fi%
8375 \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
8376 \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
8377 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
8378 \reset@msd@options@iffullpage%
8379 }%
8380 }%
8381 {\global\@msdata@insertedfrompreviouspagefalse}%
8382 }{}%
8383 \else%
8384 \ifcsdef{@msdata@\add@msd@cR @dataR}{%
8385 \ifnumless{\the\absline@numR-\@one}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}%
8386 {%
8387 \global\@msdata@insertedfrompreviouspagetrue%
8388 \letcs{\@data}{\@msdata@\add@msd@cR @dataR}%
8389 \edef\l@d@nums{%
8390 000}% Start page = we don't print it
8391 \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
8392 000}% Start subline number, for now, not used
8393 000}% End page number, we don't print it
8394 \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\@
csuse{@lastline@forpageR@\the\page@numR}}%
8395 {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
8396 {\csuse{@lastline@forpageR@\the\page@numR}}}%Otherwise, last
number of the page
8397 |%
8398 000}% End sub line number, for now, not used
8399 \edfont@info%Font
8400 }%
8401 \@msd@options@fullpagefalse%
8402 \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@one}%
8403 {%
8404 \numdef{\@tmp}{\add@msd@cR+\@one}%
8405 \ifcsdef{@msdata@\@tmp @abslinenumberR}%
8406 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}%
8407 {}%
8408 {\@msd@options@fullpagetrue}%

```

```

8409     }%
8410     {\msd@options@iffullpage}%
8411   }%
8412 {}%
8413 \listxadd{\msdata@list}{%
8414   \msd@options@iffullpage%
8415   \ifluatex%
8416     \csxdef{footnote@luatextextdir}{\the\textdir}%
8417     \csxdef{footnote@luatexpardir}{\the\pardir}%
8418   \fi%
8419   \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
8420   \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
8421   \noexpand\csuse{v\msdata@series footnote}{\msdata@series}{\{
expandonce\l@d@nums}{\msdata@label}{\expandonce\@data}}}%
8422   \reset@msd@options@iffullpage%
8423 }%
8424 }%
8425 {\global\msdata@insertedfrompreviouspagefalse}%
8426 }{}%
8427 \fi%
8428 \egroup%
8429 }%
8430 %

```

XXXI.6 Inserting footnote

Just before inserting standard insert (familiar and critical footnotes, sidenotes), we call `\insert@msdata` to insert manuscripts data's footnotes.

```

\insert@msdata31 \newcommand{\insert@msdata}{%
8432   \def\do##1{##1}%
8433   \dolistloop{\msdata@list}%
8434   \global\let\msdata@list\relax%
8435 }%
8436 %

```

XXXI.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.

```

8437 \newif\if@msd@options@fullpage%
8438 \newcommand{\msd@options@iffullpage}[0]{%
8439   \if@msd@options@fullpage%
8440     \noexpand\toggletrue{nonum@}%
8441     \ifdefvoid{\msdata@label}%
8442       {\noexpand\toggletrue{nosep@}}}%

```

```

8443     {}%
8444     \fi%
8445 }%
8446 \newcommand{\reset@msd@options@iffullpage}[0]{%
8447     \noexpand\togglefalse{nonum@}%
8448     \noexpand\togglefalse{nosep@}%
8449 }%
8450 %

```

XXXII Arrays and tables

XXXII.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.

```

8451 \newtoks\@emptytoks
8452
8453 %

```

The rest is from `amsmath`.

`\l@denbody` A token register to contain the body.

```

8454 \newtoks\l@denbody
8455
8456 %

```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```

8457 \newcommand{\addtol@denbody}[1]{%
8458     \global\l@denbody\expandafter{\the\l@denbody#1}}
8459
8460 %

```

`\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`.

```

8461 \newcommand{\l@dcollect@body}[1]{%
8462   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}}%
8463   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}%
8464   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
8465   \begingroup
8466     \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
8467     \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
8468     \processl@denbody%
8469   }%
8470
8471 %

```

\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.

```

8472 \def\l@dpush@begins#1\begin#2{%
8473   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
8474
8475 %

```

\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.

```

8476 \def\l@dcollect@@body#1\end#2{%
8477   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
8478     \expandafter\@gobble\l@dbegin@stack}%
8479   \ifx\@empty\l@dbegin@stack
8480     \endgroup
8481     \@checkend{#2}%
8482     \addtol@denbody{#1}%
8483   \else
8484     \addtol@denbody{#1\end{#2}}%
8485   \fi
8486   \processl@denbody % A little tricky! Note the grouping
8487 }
8488
8489 %

```

There was a question on CTT about how to use \collect@body for a macro taking an argument. The following is part of that thread.

From: Heiko Oberdiek <oberdiek@uni-freiburg.de>
 Newsgroups: comp.text.tex
 Subject: Re: Using \collect@body with commands that take >1 argument

Date: Fri, 08 Aug 2003 09:03:20 +0200

eed132@psu.edu (Evan) wrote:

```
> I'm trying to make a new Latex environment that acts like the>
> \colorbox command that is part of the color package. I looked through
> the FAQ and ran across this bit about using the \collect@body command
> that is part of AMSLaTeX:
> http://www.tex.ac.uk/cgi-bin/texfaq2html?label=cmdasenv
>
> It almost works. If I do something like the following:
> \newcommand{\redbox}[1]{\colorbox{red}{#1}}
>
> \makeatletter
> \newenvironment{redbox}{\collect@body \redbox}{}
```

You will get an error message: Command \redbox already defined.
Thus you must rename either the command \redbox or the environment name.

```
> \begin{coloredbox}{blue}
> Yadda yadda yadda... this is on a blue background...
> \end{coloredbox}
> and can't figure out how to make the \collect@body take this.

> \collect@body \colorbox{red}
> \collect@body {\colorbox{red}}
```

The argument of \collect@body has to be one token exactly.

```
\documentclass{article}
\usepackage{color}
\usepackage{amsmath}

\newcommand{\redbox}[1]{\colorbox{red}{#1}}
\makeatletter
\newenvironment{coloredbox}[1]{%
  \def\next@{\colorbox{#1}}%
  \collect@body\next@
}{%

% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}[1]{%
  \def\next@{\mycoloredbox{#1}}%
  \collect@body\next@
}{%
\newcommand{\mycoloredbox}[2]{%
  \colorbox{#1}{\ignorespaces#2\unskip}%
}

% support of optional color model argument
```

```

\newcommand\coloredboxIII\endcsname{}
\def\coloredboxIII#1#{%
  \@coloredboxIII{#1}%
}
\def\@coloredboxIII#1#2{%
  \def\next@{\mycoloredboxIII{#1}{#2}}%
  \collect@body\next@
}
\newcommand{\mycoloredboxIII}[3]{%
  \colorbox#1{#2}{\ignorespaces#3\unskip}%
}

\makeatother

\begin{document}
  Black text before
  \begin{coloredbox}{blue}
    Hello World
  \end{coloredbox}
  Black text after

  Black text before
  \begin{coloredboxII}{blue}
    Hello World
  \end{coloredboxII}
  Black text after

  Black text before
  \begin{coloredboxIII}[rgb]{0,0,1}
    Hello World
  \end{coloredboxIII}
  Black text after

\end{document}

Yours sincerely
  Heiko <oberdiek@uni-freiburg.de>

```

XXXII.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`.

XXXII.2.1 Disabling and restoring commands

`\l@dtabnoexpands` More no expansion for critical and familiar footnotes in tabular environment.

```

8490 \newcommand*\l@dtabnoexpands}{%
8491   \let\rtab=0%
8492   \let\ctab=0%
8493   \let\ltab=0%
8494   \let\rtabtext=0%
8495   \let\ltabtext=0%
8496   \let\ctabtext=0%
8497   \let\edbeforetab=0%
8498   \let\edaftertab=0%
8499   \let\edatleft=0%
8500   \let\edatright=0%
8501   \let\edvertline=0%
8502   \let\edvertdots=0%
8503   \let\edrowfill=0%
8504 }
8505
8506 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in edtabularx and edarrayx environments.

`\restore@familiarnotes`

```

8507 \newcommand{\disable@familiarnotes}{%
8508   \unless\ifnofamiliar%
8509     \def\do##1{%
8510       \csletcs{footnote@@##1}{footnote##1}%
8511       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
8512         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
8513         \csuse{@footnotemark##1}%
8514       }%
8515     }%
8516     \dolistloop{\@series}%
8517   \fi%
8518 }%
8519 \newcommand{\restore@familiarnotes}{%
8520   \unless\ifnofamiliar%
8521     \def\do##1{%
8522       \csletcs{footnote##1}{footnote@@##1}%
8523     }%
8524     \dolistloop{\@series}%
8525   \fi%
8526 }%
8527
8528 %

```

`\disable@sidenotes` The same for sidenotes.

`\restore@sidenotes`


```

8529 \newcommand{\disable@sidenotes}{%
8530   \let\@@ledrightnote\ledrightnote%
8531   \let\@@ledleftnote\ledleftnote%
8532   \let\@@ledsidenote\ledsidenote%
8533   \let\ledrightnote@gobble%
8534   \let\ledleftnote@gobble%
8535   \let\ledsidenote@gobble%
8536 }%
8537 \newcommand{\restore@sidenotes}{%
8538   \let\ledrightnote\@@ledrightnote%
8539   \let\ledleftnote\@@ledleftnote%
8540   \let\ledsidenote\@@ledsidenote%
8541 }%
8542 %

```

`\disable@endnotes` The same for endnotes.

```

\restore@endnotes
8543 \newcommand{\disable@endnotes}{%
8544   \unless\ifnoend{%
8545     \def\do##1{%
8546       \csletcs{##1@endnote}{##1endnote}%
8547       \expandafter\renewcommand \csname ##1endnote\endcsname[1]{}%
8548     }%
8549     \dolistloop{\@series}%
8550   \fi%
8551 }%
8552 \newcommand{\restore@endnotes}{%
8553   \unless\ifnofamiliar{%
8554     \def\do##1{%
8555       \csletcs{##1endnote}{##1@endnote}%
8556     }%
8557     \dolistloop{\@series}%
8558   \fi%
8559 }%
8560 %

```

`\disable@notes` Disable/restore side, familiar and end notes.

```

\restore@notes
8561 \newcommand{\disable@notes}{%
8562   \disable@sidenotes%
8563   \disable@familiarnotes%
8564   \disable@endnotes%
8565 }%
8566 \newcommand{\restore@notes}{%
8567   \restore@sidenotes%
8568   \restore@familiarnotes%
8569   \restore@endnotes%
8570 }%
8571 %

```

\EDTEXT We need to be able to modify the `\edtext` macros and also restore their original definitions.
\xedtext

```
8572 \let\EDTEXT=\edtext
8573 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
8574 %
```

\EDLABEL We need to be able to modify and restore the `\edlabel` macro.

\xedlabel

```
8575 \let\EDLABEL=\edlabel
8576 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}
8577 %
```

\xedindex Macros supporting modification and restoration of `\edindex`.

\nulledindex

```
8578 \AtBeginDocument{\let\xedindex\edindex}%
8579 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
8580
8581 %
```

\@line@num Macro supporting restoration of `\linenum`.

```
8582 \let\@line@num=\linenum
8583 %
```

\l@dgobblearg `\l@dgobbleoptarg[\langle arg \rangle]{\langle arg \rangle}` replaces these two arguments (first is optional) by `\relax`.

```
8584 \newcommand*{\l@dgobbleoptarg}[2][\relax]%
8585
8586 %
```

\Relax `\let\Relax=\relax`

\NEXT `\let\NEXT=\next`

```
8589
8590 %
```

\l@dmodforedtext Modify and restore various macros for when `\edtext` is used.

\l@drestoreforedtext

```
8591 \newcommand{\l@dmodforedtext}{%
8592   \let\edtext\relax
8593   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
8594   \dolistloop{\@series}%
8595   \let\edindex\nulledindex
8596   \let\linenum@gobble}
8597 \newcommand{\l@drestoreforedtext}{%
8598   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
8599   \dolistloop{\@series}%
8600   \let\edindex\xedindex}
8601 %
```

`\l@dnnullfills` Nullify and restore some column fillers, etc.

```
\l@drestorefills
8602 \newcommand{\l@dnnullfills}{%
8603   \def\edlabel##1{%
8604     \def\edrowfill##1##2##3{%
8605     }
8606   \newcommand{\l@drestorefills}{%
8607     \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
8608   }
8609
8610 %
```

`\letsforverteilen` Gathers some lets and other code that is common to the **verteilen** macros.

```
8611 \newcommand{\letsforverteilen}{%
8612   \let\edtext\xedtext
8613   \let\edindex\xedindex
8614   \def\do##1{\global\csletcs{##1footnote}{##1@@footnote}}
8615   \dolistloop{\@series}%
8616   \let\linenum@line@num
8617   \hilfsskip=\l@dcwidth%
8618   \advance\hilfsskip by -\wd\hilfsbox
8619   \def\edlabel##1{\xedlabel{##1}}
8620
8621 %
```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for
`\enablel@dtabfeet` `\edtext`.

```
8622 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
8623 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
8624 %
```

XXXII.2.2 Counters, boxes and lengths

`\l@dampcount` `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter
`\l@dcolcount` for the columns.

```
8625 \newcount\l@dampcount
8626 \l@dampcount=1\relax
8627 \newcount\l@dcolcount
8628 \l@dcolcount=0\relax
8629
8630 %
```

`\hilfsbox` Some (temporary) helper items.

```
\hilfsskip
8631 \newbox\hilfsbox
\Hilfsbox
8632 \newskip\hilfsskip
\hilfscount
8633 \newbox\Hilfsbox
```

```

8634 \newcount\hilfscount
8635
8636 %

```

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).

```

8637 \newdimen\dcoli
8638 \newdimen\dcolii
8639 \newdimen\dcoliii
8640 \newdimen\dcoliv
8641 \newdimen\dcolv
8642 \newdimen\dcolvi
8643 \newdimen\dcolvii
8644 \newdimen\dcolviii
8645 \newdimen\dcolix
8646 \newdimen\dcolx
8647 \newdimen\dcolxi
8648 \newdimen\dcolxii
8649 \newdimen\dcolxiii
8650 \newdimen\dcolxiv
8651 \newdimen\dcolxv
8652 \newdimen\dcolxvi
8653 \newdimen\dcolxvii
8654 \newdimen\dcolxviii
8655 \newdimen\dcolxix
8656 \newdimen\dcolxx
8657 \newdimen\dcolxxi
8658 \newdimen\dcolxxii
8659 \newdimen\dcolxxiii
8660 \newdimen\dcolxxiv
8661 \newdimen\dcolxxv
8662 \newdimen\dcolxxvi
8663 \newdimen\dcolxxvii
8664 \newdimen\dcolxxviii
8665 \newdimen\dcolxxix
8666 \newdimen\dcolxxx
8667 \newdimen\dcolerr % added for error handling
8668
8669 %

```

\l@dcolwidth This is a cunning way of storing the columnwidths indexed by the column number \l@dcolcount, like an array. (was \Dimenzuordnung)

```

8670 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???
8671 \or \dcoli \or \dcolii \or \dcoliii
8672 \or \dcoliv \or \dcolv \or \dcolvi
8673 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
8674 \or \dcolxi \or \dcolxii \or \dcolxiii
8675 \or \dcolxiv \or \dcolxv \or \dcolxvi

```

```

8676 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
8677 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
8678 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
8679 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
8680 \else \dcolerr \fi}
8681
8682 %

```

\stepl@dcolcount This increments the column counter, and issues an error message if it is too large.

```

8683 \newcommand*\stepl@dcolcount{\advance\l@dcolcount\@ne
8684 \ifnum\l@dcolcount>30\relax
8685 \led@err@TooManyColumns
8686 \fi}
8687
8688 %

```

\l@dsetmaxcolwidth Sets the column width to the maximum value seen so far.

```

8689 \newcommand{\l@dsetmaxcolwidth}{%
8690 \ifdim\l@dcolwidth < \wd\hilfsbox
8691 \l@dcolwidth = \wd\hilfsbox
8692 \else \relax \fi}
8693
8694 %

```

\measuremcell Measure (recursively) the width required for a math cell.

```

8695 \def\measuremcell #1{%
8696 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8697 \else\l@dcheckcols%
8698 \l@dcolcount=0%
8699 \let\NEXT\measuremcell%
8700 \fi%
8701 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8702 \stepl@dcolcount%
8703 \l@dsetmaxcolwidth%
8704 \let\NEXT\measuremcell%
8705 \fi\NEXT}
8706
8707 %

```

\measuretcell Measure (recursively) the width required for a text cell.

```

8708 \def\measuretcell #1{%
8709 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8710 \else\l@dcheckcols%
8711 \l@dcolcount=0%
8712 \let\NEXT\measuretcell%
8713 \fi%

```

```

8714 \else\setbox\hilfsbox=\hbox{#1}%
8715 \step1@dcolcount%
8716 \l@setmaxcolwidth%
8717 \let\NEXT\measuretcell%
8718 \fi\NEXT}
8719
8720 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

8721 \def\measuremrow #1\{\%
8722 \ifx #1&\let\NEXT\relax%
8723 \else\measuremcell #1&\&\&\&%
8724 \let\NEXT\measuremrow%
8725 \fi\NEXT}
8726 %

```

\measuretrow Measure (recursively) the width required for a text row.

```

8727 \def\measuretrow #1\{\%
8728 \ifx #1&\let\NEXT\relax%
8729 \else\measuretcell #1&\&\&\&%
8730 \let\NEXT\measuretrow%
8731 \fi\NEXT}
8732
8733 %

```

\edtabcolsep The length `\edtabcolsep` controls the distance between columns.

```

8734 \newskip\edtabcolsep
8735 \global\edtabcolsep=10pt
8736
8737 %

```

\variab₃₈ `\newcommand{\variab}{\relax}`

```

8739
8740 %

```

\l@dcheckcols Check that the number of columns is consistent.

```

8741 \newcommand*\l@dcheckcols{\%
8742 \ifnum\l@dcolcount=1\relax
8743 \else
8744 \ifnum\l@dampcount=1\relax
8745 \else
8746 \ifnum\l@dcolcount=\l@dampcount\relax
8747 \else
8748 \l@d@err@UnequalColumns

```

```

8749     \fi
8750     \fi
8751     \l@dampcount=\l@dcolcount
8752 \fi}
8753
8754 %

```

`\edfilldimen` A length.

```

8755 \newdimen\edfilldimen
8756 \edfilldimen=0pt
8757
8758 %

```

`\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`

```

8759 \newcounter{addcolcount}
8760 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
8761 %

```

XXXII.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.

```

8762 \def\setmcellright #1&{\def\edlabel##1{}%
8763     \let\edindex\nulledindex
8764     \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
8765         \let\Next\relax%
8766     \else\l@dcolcount=0%
8767         \let\Next=\setmcellright%
8768     \fi%
8769 \else%
8770     \disablel@dtabfeet%
8771     \stepl@dcolcount%
8772     \disable@notes%
8773     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8774     \restore@notes%
8775     \letsforverteilen%
8776     \hskip\hilfsskip$\displaystyle{#1}$%
8777     \hskip\edtabcolsep%
8778     \let\Next=\setmcellright%
8779 \fi\Next}
8780
8781 %

```

`\settcclright` Typeset (recursively) cells of text right justified.

```

8782 \def\settcellright #1{\def\edlabel##1{}%
8783   \let\edindex\nulledindex
8784   \ifx #1\\ \ifnum\l@dc@count=0%\removelastskip
8785     \let\Next\relax%
8786   \else\l@dc@count=0%
8787     \let\Next=\settcellright%
8788   \fi%
8789   \else%
8790     \disablel@dtabfeet%
8791     \step1@dc@count%
8792     \disable@notes%
8793     \setbox\hilfsbox=\hbox{#1}%
8794     \restore@notes%
8795     \letsforverteilen%
8796     \hskip\hilfsskip#1%
8797     \hskip\edtabcolsep%
8798     \let\Next=\settcellright%
8799   \fi\Next}
8800 %

```

\setmcellleft Typeset (recursively) cells of display math left justified.

```

8801 \def\setmcellleft #1{\def\edlabel##1{}%
8802   \let\edindex\nulledindex
8803   \ifx #1\\ \ifnum\l@dc@count=0 \let\Next\relax%
8804     \else\l@dc@count=0%
8805       \let\Next=\setmcellleft%
8806     \fi%
8807   \else \disablel@dtabfeet%
8808     \step1@dc@count%
8809     \disable@notes%
8810     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8811     \restore@notes%
8812     \letsforverteilen%
8813     $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
8814     \let\Next=\setmcellleft%
8815   \fi\Next}
8816
8817 %

```

\settcellleft Typeset (recursively) cells of text left justified.

```

8818 \def\settcellleft #1{\def\edlabel##1{}%
8819   \let\edindex\nulledindex
8820   \ifx #1\\ \ifnum\l@dc@count=0 \let\Next\relax%
8821     \else\l@dc@count=0%
8822       \let\Next=\settcellleft%
8823     \fi%
8824   \else \disablel@dtabfeet%
8825     \step1@dc@count%

```



```

8826         \disable@notes%
8827         \setbox\hilfsbox=\hbox{#1}%
8828         \restore@notes%
8829         \letsforverteilen%
8830         #1\hskip\hilfsskip\hskip\edtabcolsep%
8831         \let\Next=\settcclleft%
8832     \fi\Next}
8833 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

8834 \def\setmcellcenter #1&{\def\edlabel##1}%
8835     \let\edindex\nulledindex
8836     \ifx #1\ \ifnum\l@dc@count=0 \let\Next\relax%
8837         \else\l@dc@count=0%
8838             \let\Next=\setmcellcenter%
8839         \fi%
8840     \else \disablel@dt@feet%
8841         \stepl@dc@count%
8842         \disable@notes%
8843         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8844         \restore@notes%
8845         \letsforverteilen%
8846         \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
8847         \hskip\edtabcolsep%
8848         \let\Next=\setmcellcenter%
8849     \fi\Next}
8850 %
8851 %

```

\settcclcenter Typeset (recursively) cells of text centered.

```

8852 \def\settcclcenter #1&{\def\edlabel##1}%
8853     \let\edindex\nulledindex
8854     \ifx #1\ \ifnum\l@dc@count=0 \let\Next\relax%
8855         \else\l@dc@count=0%
8856             \let\Next=\settcclcenter%
8857         \fi%
8858     \else \disablel@dt@feet%
8859         \stepl@dc@count%
8860         \disable@notes%
8861         \setbox\hilfsbox=\hbox{#1}%
8862         \restore@notes%
8863         \letsforverteilen%
8864         \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
8865         \hskip\edtabcolsep%
8866         \let\Next=\settcclcenter%
8867     \fi\Next}
8868 %
8869 %

```

```
\NEXT 8870 \let\NEXT=\relax
```

```
8871
```

```
8872 %
```

\setmrowright Typeset (recursively) rows of right justified math.

```
8873 \def\setmrowright #1\{\%
8874   \ifx #1& \let\NEXT\relax
8875   \else \centerline{\setmcellright #1&\\&\\&}
8876     \let\NEXT=\setmrowright
8877   \fi\NEXT}
8878 %
```

\settroright Typeset (recursively) rows of right justified text.

```
8879 \def\settroright #1\{\%
8880   \ifx #1& \let\NEXT\relax
8881   \else \centerline{\settcclright #1&\\&\\&}
8882     \let\NEXT=\settroright
8883   \fi\NEXT}
8884 %
8885 %
```

\setmrowleft Typeset (recursively) rows of left justified math.

```
8886 \def\setmrowleft #1\{\%
8887   \ifx #1& \let\NEXT\relax
8888   \else \centerline{\setmcellleft #1&\\&\\&}
8889     \let\NEXT=\setmrowleft
8890   \fi\NEXT}
8891 %
```

\settrorleft Typeset (recursively) rows of left justified text.

```
8892 \def\settrorleft #1\{\%
8893   \ifx #1& \let\NEXT\relax
8894   \else \centerline{\settcclleft #1&\\&\\&}
8895     \let\NEXT=\settrorleft
8896   \fi\NEXT}
8897 %
8898 %
```

\setmrowcenter Typeset (recursively) rows of centered math.

```
8899 \def\setmrowcenter #1\{\%
8900   \ifx #1& \let\NEXT\relax%
8901   \else \centerline{\setmcellcenter #1&\\&\\&}
8902     \let\NEXT=\setmrowcenter
8903   \fi\NEXT}
8904 %
```

\settrowcenter Typeset (recursively) rows of centered text.

```

8905 \def\settrowcenter #1\{\%
8906   \ifx #1& \let\NEXT\relax
8907   \else \centerline{\settcellcenter #1&\&\&}
8908     \let\NEXT=\settrowcenter
8909   \fi\NEXT}
8910
8911 %

```

```

\newcommand{\nullsetzen}{\%
8913   \step1@dcolcount%
8914   \l@dcolwidth=0pt%
8915   \ifnum\l@dcolcount=30\let\NEXT\relax%
8916     \l@dcolcount=0\relax
8917   \else\let\NEXT\nullsetzen%
8918   \fi\NEXT}
8919
8920 %

```

\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.

```

8921 \newcommand{\edatleft}[3][\@empty]{\%
8922   \ifx#1\@empty
8923     \vbox to 10pt{\vss\hbox{\$ \left#2\vrule width0pt height #3
8924       depth 0pt \right. \$\hss}\vfil}
8925   \else
8926     \vbox to 4pt{\vss\hbox{\$#1\left#2\vrule width0pt height #3
8927       depth 0pt \right. \$}\vfil}
8928   \fi}
8929 %

```

\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.

```

8930 \newcommand{\edatright}[3][\@empty]{\%
8931   \ifx#1\@empty
8932     \vbox to 10pt{\vss\hbox{\$ \left.\vrule width0pt height #3
8933       depth 0pt \right#2 \$\hss}\vfil}
8934   \else
8935     \vbox to 4pt{\vss\hbox{\$ \left.\vrule width0pt height #3
8936       depth 0pt \right#2 #1 \$}\vfil}
8937   \fi}
8938
8939 %

```

\edvertline $\edvertline{\langle len \rangle}$ vertical line $\langle len \rangle$ high.

```

8940 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
8941
8942 %

```

\edvertdots `\edvertdots{<len>}` vertical dotted line <len> high.

```

8943 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
8944   {\cleaders\hbox{$\mathhbox{.}\vbox to 0.5em{ }$}\vfil}}}
8945
8946 %

```

\l@dtabaddcols `\l@dtabaddcols{<startcol>}{<endcol>}` adds the widths of the columns <startcol> through <endcol> to `\edfilldimen`. It is a \TeX style reimplementation of the original `\@add@`.

```

8947 \newcommand{\l@dtabaddcols}[2]{%
8948   \l@dcheckstartend{#1}{#2}%
8949   \ifl@dstartendok
8950     \setcounter{addcolcount}{#1}%
8951     \@whilenum \value{addcolcount}<#2\relax \do
8952       {\advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
8953        \advance\edfilldimen by \edtabcolsep
8954        \stepcounter{addcolcount}}%
8955     \advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
8956     \fi
8957 }
8958
8959 %

```

\ifl@dstartendok `\l@dcheckstartend{<startcol>}{<endcol>}` checks that the values of <startcol> and <endcol> are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```

8960 \newif\ifl@dstartendok
8961 \newcommand{\l@dcheckstartend}[2]{%
8962   \l@dstartendoktrue
8963   \ifnum #1<\@ne
8964     \l@dstartendokfalse
8965     \led@err@LowStartColumn
8966   \fi
8967   \ifnum #2>30\relax
8968     \l@dstartendokfalse
8969     \led@err@HighEndColumn
8970   \fi
8971   \ifnum #1>#2\relax
8972     \l@dstartendokfalse
8973     \led@err@ReverseColumns
8974   \fi
8975 }
8976
8977 %

```

`\edrowfill` `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a \LaTeX style reimplementation and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktel` macros.

```

8978 \newcommand*{\edrowfill}[3]{%
8979   \l@dtabaddcols{#1}{#2}%
8980   \hb@xt@ \the\l@dcowidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
8981 \let\@edrowfill=\edrowfill
8982 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
8983
8984 %

```

`\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`.

```

8985 \newcommand{\leftltab}[1]{%
8986   \hb@xt@ \z@{\vbox{\edtabindent%
8987     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8988
8989 %

```

`\leftrtab` `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

8990 \newcommand{\leftrtab}[2]{%
8991   #2\hb@xt@ \z@{\vbox{\edtabindent%
8992     \advance\Hilfsskip by\dcoli%
8993     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8994
8995 %

```

`\leftctab` `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

8996 \newcommand{\leftctab}[2]{%
8997   \hb@xt@ \z@{\vbox{\edtabindent\l@dcowidth=\l@dampcount%
8998     \advance\Hilfsskip by 0.5\dcoli%
8999     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9000       \disablel@dtabfeet$\displaystyle{#2}$}%
9001       \advance\Hilfsskip by -0.5\wd\hilfsbox%
9002       \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
9003     #2}
9004
9005 %

```

`\rightctab` `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

9006 \newcommand{\rightctab}[2]{%
9007     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9008         \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9009         #1\hb@xt@\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9010         \advance\Hilfsskip by 0.5\l@dcolwidth%
9011         \advance\Hilfsskip by -\wd\hilfsbox%
9012         \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9013             \disablel@dtabfeet$\displaystyle{#1}$}%
9014             \advance\Hilfsskip by -0.5\wd\hilfsbox%
9015             \advance\Hilfsskip by \edtabcolsep%
9016             \moveright\Hilfsskip\hbox{ #2}}\hss}%
9017         }
9018     }
9019 %

```

\rightltab `\rightltab{<math>{<math>}{<text>}` for `\edaftertab` in `\ltab`.

```

9020 \newcommand{\rightltab}[2]{%
9021     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9022         \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
9023         #1\hb@xt@\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
9024         \advance\Hilfsskip by\l@dcolwidth%
9025         \advance\Hilfsskip by-\wd\hilfsbox%
9026         \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9027             \disablel@dtabfeet$\displaystyle{#1}$}%
9028             \advance\Hilfsskip by-\wd\hilfsbox%
9029             \advance\Hilfsskip by\edtabcolsep%
9030             \moveright\Hilfsskip\hbox{ #2}}\hss}%
9031         }
9032     }
9033 %

```

\rightrtab `\rightrtab{<math>{<math>}{<text>}` for `\edaftertab` in `\rtab`.

```

9034 \newcommand{\rightrtab}[2]{%
9035     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
9036         \disablel@dtabfeet#2}%
9037         #1\hb@xt@\z@\vbox{\edtabindent%
9038         \advance\Hilfsskip by-\wd\hilfsbox%
9039         \advance\Hilfsskip by\edtabcolsep%
9040         \moveright\Hilfsskip\hbox{ #2}}\hss}%
9041     }
9042 }
9043 %

```

\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.

```

9044 \newcommand{\rtab}[1]{%
9045   \l@dnnullfills
9046   \def\edbeforetab##1##2{\lefttrtab{##1}{##2}}%
9047   \def\edaftertab##1##2{\righttrtab{##1}{##2}}%
9048   \measurebody{#1}%
9049   \l@drestorefills
9050   \variab
9051   \setmrowright #1\&\&\&%
9052   \enablel@dtabfeet}
9053
9054 %

```

\measurebody `\measurebody{<body>}` measures the array `<body>`.

```

9055 \newcommand{\measurebody}[1]{%
9056   \disablel@dtabfeet%
9057   \l@dcolcount=0%
9058   \nullsetzen%
9059   \l@dcolcount=0
9060   \measuremrow #1\&\&\&%
9061   \global\l@dampcount=1}
9062
9063 %

```

\rtabtext `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```

9064 \newcommand{\rtabtext}[1]{%
9065   \l@dnnullfills
9066   \measuretbody{#1}%
9067   \l@drestorefills
9068   \variab
9069   \settroright #1\&\&\&%
9070   \enablel@dtabfeet}
9071
9072 %

```

\measuretbody `\measuretbody{<body>}` measures the tabular `<body>`.

```

9073 \newcommand{\measuretbody}[1]{%
9074   \disable@notes%
9075   \disablel@dtabfeet%
9076   \l@dcolcount=0%
9077   \nullsetzen%
9078   \l@dcolcount=0
9079   \measuretror #1\&\&\&%
9080   \restore@notes%
9081   \global\l@dampcount=1}
9082
9083 %

```

```

\ltab Array with entries left justified.
\edbeforetab
\edaftertab
9084 \newcommand{\ltab}[1]{%
9085 \l@dnnullfills
9086 \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
9087 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
9088 \measurebody{#1}%
9089 \l@drestorefills
9090 \variab
9091 \setmrowleft #1\\&\\%
9092 \enablel@dtabfeet}
9093
9094 %

```

\ltabtext Tabular with entries left justified.

```

9095 \newcommand{\ltabtext}[1]{%
9096 \l@dnnullfills
9097 \measuretbody{#1}%
9098 \l@drestorefills
9099 \variab
9100 \settrrowleft #1\\&\\%
9101 \enablel@dtabfeet}
9102
9103 %

```

```

\ctab Array with centered entries.
\edbeforetab
\edaftertab
9104 \newcommand{\ctab}[1]{%
9105 \l@dnnullfills
9106 \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
9107 \def\edaftertab##1##2{\rightctab{##1}{##2}}%
9108 \measurebody{#1}%
9109 \l@drestorefills
9110 \variab
9111 \setmrowcenter #1\\&\\%
9112 \enablel@dtabfeet}
9113
9114 %

```

\ctabtext Tabular with entries centered.

```

9115 \newcommand{\ctabtext}[1]{%
9116 \l@dnnullfills
9117 \measuretbody{#1}%
9118 \l@drestorefills
9119 \variab
9120 \settrrowcenter #1\\&\\%
9121 \enablel@dtabfeet}
9122
9123 %

```



```
\spreadtext24 \newcommand{\spreadtext}[1]{% \l@dc@lcount=\l@dampcount%
9125 \hb@xt@ \the\l@dc@lwidth{\hbox{#1}\hss}}
9126 %
```

```
\spreadmath27 \newcommand{\spreadmath}[1]{%
9128 \hb@xt@ \the\l@dc@lwidth{\hbox{$\displaystyle{#1}$}\hss}}
9129
9130 %
```

\HILFSskip More helpers.

```
\Hilfsskip
9131 \newskip\HILFSskip
9132 \newskip\Hilfsskip
9133
9134 %
```

```
\EDTABINDENT35 \newcommand{\EDTABINDENT}{%
9136 \ifnum\l@dc@lcount=30\let\NEXT\relax\l@dc@lcount=0%
9137 \else\step\l@dc@lcount%
9138 \advance\Hilfsskip by\l@dc@lwidth%
9139 \ifdim\l@dc@lwidth=0pt\advance\hilfscout\@ne
9140 \else\advance\Hilfsskip by \the\hilfscout\edtabcolsep%
9141 \hilfscout=1\fi%
9142 \let\NEXT=\EDTABINDENT%
9143 \fi\NEXT}%
9144 %
```

\edtabindent (was \tabindent)

```
9145 \newcommand{\edtabindent}{%
9146 \l@dc@lcount=0\relax
9147 \Hilfsskip=0pt%
9148 \hilfscout=1\relax
9149 \EDTABINDENT%
9150 \hilfsskip=\hsize%
9151 \advance\hilfsskip -\Hilfsskip%
9152 \Hilfsskip=0.5\hilfsskip%
9153 }%
9154
9155 %
```

\EDTAB (was \TAB)

```
9156 \def\EDTAB #1|#2|{%
9157 \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
9158 \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
9159 \advance\tabelskip -\wd\tabhilfbox%
```

```

9160 \advance\tabelskip -\wd\tabHilfbox%
9161 \unhbox\tabhilfbox\hskip\tabelskip%
9162 \unhbox\tabHilfbox}%
9163
9164 %

```

`\EDTABtext` (was `\TABtext`)

```

9165 \def\EDTABtext #1|#2|{%
9166 \setbox\tabhilfbox=\hbox{#1}%
9167 \setbox\tabHilfbox=\hbox{#2}%
9168 \advance\tabelskip -\wd\tabhilfbox%
9169 \advance\tabelskip -\wd\tabHilfbox%
9170 \unhbox\tabhilfbox\hskip\tabelskip%
9171 \unhbox\tabHilfbox}%
9172 %

```

`\tabhilfbox` Further helpers.

```

\begin{code}
\tabHilfbox
9173 \newbox\tabhilfbox
9174 \newbox\tabHilfbox
9175
9176 %
\end{code}

```

XXXII.2.4 Environments

`edarrayl edarrayc edarrayr` The ‘environment’ forms for `\ltab`, `\ctab` and `\rtab`.

```

9177 \newenvironment{edarrayl}{\l@collect@body\ltab}{\l@collect@body\ltab}%
9178 \newenvironment{edarrayc}{\l@collect@body\ctab}{\l@collect@body\ctab}%
9179 \newenvironment{edarrayr}{\l@collect@body\rtab}{\l@collect@body\rtab}%
9180
9181 %

```

`edtabularl edtabularc edtabularr` The ‘environment’ forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

```

9182 \newenvironment{edtabularl}{\l@collect@body\ltabtext}{\l@collect@body\ltabtext}%
9183 \newenvironment{edtabularc}{\l@collect@body\ctabtext}{\l@collect@body\ctabtext}%
9184 \newenvironment{edtabularr}{\l@collect@body\rtabtext}{\l@collect@body\rtabtext}%
9185
9186 %

```

XXXIII 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
\newcommand{\initnumbering@quote}{
  \ifnoquotation@else
  \renewcommand{\quotation}{\par\leavevmode%
    \parindent=1.5em%
    \skipnumbering%
    \ifautopar%
      \vskip-\parskip%
    \else%
      \vskip\topsep%
    \fi%
    \global\leftskip=\leftmargin%
    \global\rightskip=\leftmargin%
  }
  \renewcommand{\endquotation}{\par%
    \global\leftskip=0pt%
    \global\rightskip=0pt%
    \leavevmode%
    \skipnumbering%
    \ifautopar%
      \vskip-\parskip%
    \else%
      \vskip\topsep%
    \fi%
  }
  \renewcommand{\quote}{\par\leavevmode%
    \parindent=0pt%
    \skipnumbering%
    \ifautopar%
      \vskip-\parskip%
    \else%
      \vskip\topsep%
    \fi%
    \global\leftskip=\leftmargin%
    \global\rightskip=\leftmargin%
  }
  \renewcommand{\endquote}{\par%
    \global\leftskip=0pt%
    \global\rightskip=0pt%
    \leavevmode%
    \skipnumbering%
    \ifautopar%
      \vskip-\parskip%
    \else%
      \vskip\topsep%
    \fi%
  }
}
\fi
}

```

```
9235 %
```

XXXIV Section's title commands

XXXIV.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```
9236 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
9237 %
```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```
9238 \newcommand{\ledsectnomark}{%
9239   \let\chaptermark\@gobble%
9240   \let\sectionmark\@gobble%
9241   \let\subsectionmark\@gobble%
9242 }
9243 %
```

XXXIV.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 `reledpar` 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.

XXXIV.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).

```
9244 \catcode`\#=12
9245 \notbool{@noeled@sec}{%
9246 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
9247 \ifl@dmemoir
9248 \newcommand\beforeeledchapter{%
9249 \clearforchapter%
9250 }
9251 \else
9252 \newcommand\beforeeledchapter{%
9253 \if@openright%
9254 \cleardoublepage%
9255 \else%
9256 \clearpage%
9257 \fi%
9258 }
9259 \fi
9260 %
```

XXXIV.4 Auxiliary commands

`\print@leftmargin@eledsection` `\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.

```
9261 \def\print@rightmargin@eledsection{%
9262 \if@eled@sectioning%
9263 \begingroup%
9264 \if@RTL%
9265 \let\llap\rlap%
9266 \let\leftlinenum\rightlinenum%
9267 \let\leftlinenumR\rightlinenumR%
9268 \let\l@drd@ta\l@dld@ta%
9269 \let\l@drsn@te\l@dlsn@te%
9270 \fi%
9271 \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
9272 \endgroup%
9273 \fi%
```

```

9274 }%
9275
9276 \def\print@leftmargin@eledsection{%
9277   \if@eled@sectioning%
9278     \leavevmode%
9279     \begingroup%
9280     \if@RTL%
9281       \let\rlap\llap%
9282       \let\rightlinenum\leftlinenum%
9283       \let\rightlinenumR\leftlinenumR%
9284       \let\l@dld@ta\l@dld@ta%
9285       \let\l@dlsn@te\l@dlsn@te%
9286     \fi%
9287     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
9288     \endgroup%
9289   \fi%
9290 }%
9291
9292 %

```

XXXIV.5 Patching standard commands

We have to patch \LaTeX , book and memoir sectioning commands in order to:

- Disable `\edtext` inside.
- Disable page breaking (for `\chapter`).
- Add line numbers and sidenotes.

Unfortunately, Maïeul Rouquette was not able to try if memoir is loaded. That is why `eledmac` tries to define for both standard class and memoir class.

```

9293 \AtBeginDocument{%
9294
9295
9296 \pretocmd{\M@sect}
9297   {\let\old@edtext=\edtext%
9298    \let\edtext=\dummy@edtext@showlemma%
9299   }
9300 {}
9301 {}
9302
9303 \apptocmd{\M@sect}
9304   {\let\edtext=\old@edtext}
9305 {}
9306 {}
9307
9308 \patchcmd{\M@sect}
9309   { #9}

```

```

9310 { #9%
9311 \print@rightmargin@eledsection%
9312 }
9313 {}
9314 {}
9315
9316 \patchcmd{\M@sect}
9317 {\hskip #3\relax}
9318 {\hskip #3\relax%
9319 \print@leftmargin@eledsection%
9320 }
9321 {}
9322 {}
9323
9324 \patchcmd{\@mem@old@ssect}
9325 {#5}
9326 {#5%
9327 \print@leftmargin@eledsection%
9328 }
9329 {}
9330 {}
9331
9332 \patchcmd{\@mem@old@ssect}
9333 {\hskip #1}
9334 {\hskip #1%
9335 \print@rightmargin@eledsection%
9336 }
9337 {}
9338 {}
9339
9340
9341
9342 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
9343 \if@eled@sectioning\else%
9344 \ifl@dprintingpages\else%
9345 \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a
9346 \fi}%
9347 \fi%
9348 }
9349 {}
9350 {}
9351
9352 \patchcmd{\@makechapterhead}
9353 {#1}
9354 {\print@leftmargin@eledsection%
9355 #1%
9356 \print@rightmargin@eledsection%

```

```

9357 }
9358 {}
9359 {}
9360
9361 \patchcmd{\@makechapterhead}% For BIDI
9362 {\if@RTL\raggedleft\else\raggedright\fi}%
9363 {\if@eled@sectioning\else%
9364   \if@RTL\raggedleft\else\raggedright\fi%
9365   \fi%
9366 }%
9367 {}%
9368 {}%
9369
9370 \patchcmd{\@makeschapterhead}
9371 {#1}
9372 {\print@leftmargin@eledsection%
9373   #1%
9374   \print@rightmargin@eledsection%
9375 }
9376 {}
9377 {}
9378
9379 \pretocmd{\@sect}
9380 {\let\old@edtext=\edtext
9381   \let\edtext=\dummy@edtext@showlemma%
9382 }
9383 {}
9384 {}
9385
9386 \apptocmd{\@sect}
9387 {\let\edtext=\old@edtext}
9388 {}
9389 {}
9390
9391 \pretocmd{\@ssect}
9392 {\let\old@edtext=\edtext%
9393   \let\edtext=\dummy@edtext@showlemma%
9394 }
9395 {}
9396 {}
9397
9398 \apptocmd{\@ssect}
9399 {\let\edtext=\old@edtext}
9400 {}
9401 {}
9402
9403 %

```

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.


```

9404 \@ifpackageloaded{nameref}{
9405
9406   \patchcmd{\NR@sect}
9407     {#8}
9408     {#8%
9409     \print@rightmargin@eledsection%
9410     }
9411     {}
9412     {}
9413
9414   \patchcmd{\NR@sect}
9415     {\hskip #3\relax}
9416     {\hskip #3\relax%
9417     \print@leftmargin@eledsection%
9418     }
9419     {}
9420     {}
9421
9422   \patchcmd{\NR@ssect}
9423     {#5}
9424     {#5%
9425     \print@rightmargin@eledsection%
9426     }
9427     {}
9428     {}
9429
9430   \patchcmd{\NR@ssect}
9431     {\hskip #1}
9432     {\hskip #1%
9433     \print@leftmargin@eledsection%
9434     }
9435     {}
9436     {}
9437   }%
9438   {
9439     \patchcmd{\@sect}
9440       {#8}
9441       {#8%
9442       \print@rightmargin@eledsection%
9443       }
9444       {}
9445       {}
9446
9447     \patchcmd{\@sect}
9448       {\hskip #3\relax}
9449       {\hskip #3\relax%
9450       \print@leftmargin@eledsection%
9451       }
9452       {}
9453       {}

```

```

9454 \patchcmd{\@ssect}
9455   {\#5}
9456   {\#5%
9457   \print@rightmargin@eledsection%
9458   }
9459   {}
9460   {}
9461   {}
9462
9463 \patchcmd{\@ssect}
9464   {\hskip #1}
9465   {\hskip #1%
9466   \print@leftmargin@eledsection%
9467   }
9468   {}
9469   {}
9470   }%
9471 }%
9472 %

```

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 `#`.

```

9473 {}}%
9474 \protect\catcode`\#=6 %Space NEEDS by \catcode
9475 %

```

\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.

```

9476 \AtBeginDocument{%
9477 \patchcmd{\chapter}{\clearforchapter}{%
9478   \if@eled@sectioning\else%
9479   \ifl@dprintingpages\else%
9480   \clearforchapter%
9481   \fi%
9482 \fi%
9483 }%
9484 {}%
9485 {}%
9486
9487 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
9488   \if@eled@sectioning\else%
9489   \ifl@dprintingpages%
9490   \endgraf%
9491   \else%
9492   \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
9493   \fi%

```

```

9494 \fi%
9495 }%
9496 {}%
9497 {}%
9498 }%
9499 %

```

\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.

```

9500 \newif\if@eled@sectioning%
9501 %

```

We reopen a new \notbool{@noeled@sec} statement, as we will define the \elesection commands.

```

9502 \notbool{@noeled@sec}{%
9503 %

```

XXXIV.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.

```

9504 \newwrite\eled@sectioning@out
9505 %

```

\eledchapter And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsection
\eledsubsection
\eledsubsubsection
\eledchapter*
\eledsection*
\eledsubsection*
\eledsubsubsection*
9506 \newcommand{\eledchapter}[2] [] {%
9507   \disable@familiarnotes%
9508   #2%
9509   \restore@familiarnotes%
9510   \ifledRcol%
9511     \immediate\write\eled@sectioningR@out{%
9512       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
9513     }%
9514   \else%
9515     \immediate\write\eled@sectioning@out{%
9516       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}%
9517     }%
9518   \fi%
9519 }
9520
9521 \newcommand{\eledsection}[2] [] {%
9522   \disable@familiarnotes%
9523   #2%

```

```

9524 \restore@familiarnotes%
9525 \ifledRcol%
9526 \immediate\write\eled@sectioningR@out{%
9527 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R}
9528 }%
9529 \else%
9530 \immediate\write\eled@sectioning@out{%
9531 \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}%
9532 }%
9533 \fi%
9534 }
9535
9536 \newcommand{\eledsubsection}[2][]{%
9537 \disable@familiarnotes%
9538 #2%
9539 \restore@familiarnotes%
9540 \ifledRcol%
9541 \immediate\write\eled@sectioningR@out{%
9542 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R}
9543 }%
9544 \else%
9545 \immediate\write\eled@sectioning@out{%
9546 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{R}%
9547 }%
9548 \fi%
9549 }
9550 \newcommand{\eledsubsubsection}[2][]{%
9551 \disable@familiarnotes%
9552 #2%
9553 \restore@familiarnotes%
9554 \ifledRcol%
9555 \immediate\write\eled@sectioningR@out{%
9556 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}
9557 }{R}%
9558 }%
9559 \else%
9560 \immediate\write\eled@sectioning@out{%
9561 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L}
9562 }{R}%
9563 }%
9564 \fi%
9565 }
9566 \WithSuffix\newcommand{\eledchapter}[2][]{%
9567 \disable@familiarnotes%
9568 #2%
9569 \restore@familiarnotes%

```

```

9570 \ifledRcol%
9571 \immediate\write\eled@sectioningR@out{%
9572 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumstartsR}{*}{R}
9573 }%
9574 \else%
9575 \immediate\write\eled@sectioning@out{%
9576 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}%
9577 }%
9578 \fi%
9579 }
9580
9581 \WithSuffix\newcommand\eledsection*[2] [] {%
9582 \disable@familiarnotes%
9583 #2%
9584 \restore@familiarnotes%
9585 \ifledRcol%
9586 \immediate\write\eled@sectioningR@out{%
9587 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumstartsR}{*}{R}
9588 }%
9589 \else%
9590 \immediate\write\eled@sectioning@out{%
9591 \string\eled@section{#1}{\unexpanded{#2}}{\the\pstarts@read@L}{*}{}%
9592 }%
9593 \fi%
9594 }
9595
9596 \WithSuffix\newcommand\eledsubsection*[2] [] {%
9597 \disable@familiarnotes%
9598 #2%
9599 \restore@familiarnotes%
9600 \ifledRcol%
9601 \immediate\write\eled@sectioningR@out{%
9602 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumstartsR}{*}{
R}
9603 }%
9604 \else%
9605 \immediate\write\eled@sectioning@out{%
9606 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
}{*}{}%
9607 }%
9608 \fi%
9609 }
9610
9611 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
9612 \disable@familiarnotes%
9613 #2%
9614 \restore@familiarnotes%
9615 \ifledRcol%
9616 \immediate\write\eled@sectioningR@out{%
9617 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumstartsR

```

```

}{{*}}{R}
    }%
9618   \else%
9619     \immediate\write\eled@sectioning@out{%
9620       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\pstarts@read@L
9621     }{{*}}}%
9622     }%
9623   \fi%
9624 }
9625 %

```

XXXIV.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.

```

9626 \def\eled@chapter#1#2#3#4#5{%
9627   \ifstrempy{#4}%
9628   {%
9629     \ifstrempy{#1}%
9630     {%
9631       \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}%
9632       \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%
9633       }%Need for \pairs, because of using parbox.
9634       {%
9635         \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
9636         \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}}%Need for \pairs, because of using parbox.
9637       }%
9638     }%
9639     {%
9640       \ifstrempy{#1}%
9641       {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
9642       {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#1}{#2}}}%Bug in LaTeX!
9643     }%
9644     \listcsgadd{eled@sections#5@@}{#3}%

```

```

9645 }
9646 \def\eled@section#1#2#3#4#5{%
9647   \ifstrempy{#4}%
9648     {\ifstrempy{#1}%
9649       {%
9650         \csgdef{eled@sectioning@#3#5}{\section{#2}}%
9651         \csgdef{eled@sectmark@#3#5}{\let\edtext=dummy@edtext{}\sectionmark
9652 {#2}}}%Need for \pairs, because of using parbox.
9653       }%
9654       {\csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
9655        \csgdef{eled@sectmark@#3#5}{\let\edtext=dummy@edtext{}\sectionmark
9656 {#1}}}%Need for \pairs, because of using parbox.
9657       }%
9658     }{\ifstrempy{#1}%
9659       {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
9660       {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
9661     }
9662   \listcsgadd{eled@sections#5@@}{#3}%
9663 }
9664 \def\eled@subsection#1#2#3#4#5{%
9665   \ifstrempy{#4}%
9666     {\ifstrempy{#1}%
9667       {%
9668         \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
9669         \csgdef{eled@sectmark@#3#5}{\let\edtext=dummy@edtext{}\csuse{
9670 subsectionmark}{#2}}}%Need for \pairs, because of using parbox. \csuse in
9671 case of \subsectionmark is not defined (book)
9672       }%
9673       {\csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
9674        \csgdef{eled@sectmark@#3#5}{\let\edtext=dummy@edtext{}\csuse{
9675 subsectionmark}{#1}}}%Need for \pairs, because of using parbox. \csuse in
9676 case of \subsectionmark is not defined (book)
9677       }%
9678     }{\ifstrempy{#1}%
9679       {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
9680       {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
9681     }
9682   \listcsgadd{eled@sections#5@@}{#3}%
9683 }
9684 \def\eled@subsubsection#1#2#3#4#5{%
9685   \ifstrempy{#4}%
9686     {\ifstrempy{#1}%
9687       {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
9688       {\csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
9689     }%
9690     {\ifstrempy{#1}%

```

```

9689      {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
9690      {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in
LaTeX!
9691      }
9692      \listcsgadd{eled@sections#5@#}{#3}%
9693      }
9694
9695 %

```

End of the conditional test about noeledsec option.

```

9696 }{}
9697 %

```

XXXV 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.

```

9698 \def\normal@page@break{}
9699 %

```

\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.

```

9700 \def\l@prev@pb{}
9701 \def\l@prev@nopb{}
9702 %

```

\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.

```

9703 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
9704 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
9705 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
9706 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
9707 %

```


`\led@pb` The `\led@pb` adds the absolute line number in the `\prev@pb` list. The `\led@pbnum` adds the argument in the `\prev@pb` list. The `\led@nopb` adds the absolute line number in the `\prev@nopb` list. The `\led@nopbnum` adds the argument in the `\prev@nopb` list.

```

9708 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
9709 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{\the\absline@num}{#1}}
9710 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
9711 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{\the\absline@num}{#1}}
9712 %

```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

```

9713 \def\led@pb@setting{before}
9714 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
9715 %

```

`\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`.

```

9716 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
9717 \newcommand{\led@check@nopb}{%
9718   \IfStrEq{\led@pb@setting}{before}{%
9719     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
9720       {\numdef{\abs@prevline}{\the\absline@num-1}}%
9721       \xifinlist{\abs@prevline}{\normal@page@break}{%
9722         {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9723       }%
9724     }%
9725   }%
9726 }%
9727 \IfStrEq{\led@pb@setting}{after}{%
9728   \xifinlist{\the\absline@num}{\l@prev@nopb}{%
9729     \xifinlist{\the\absline@num}{\normal@page@break}{%
9730       {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9731     }%
9732   }%
9733 }%
9734 }%
9735 }%
9736 }
9737 %

```

XXXVI 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.

```

9738 \newcommand{\check@pb@in@verse}{%
9739   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
9740   \ifnum\page@num=\last@page@num\else%If we have change page
9741     \IfStrEq{\led@pb@setting}{before}{%
9742       \numgdef{\abs@line@verse}{\the\absline@num-1}%
9743       \ledpbnum{\abs@line@verse}%
9744     }{}%
9745     \IfStrEq{\led@pb@setting}{after}{%
9746       \numgdef{\abs@line@verse}{\the\absline@num-1}%
9747       \lednopbnum{\abs@line@verse}%
9748     }{}%
9749   \fi%
9750 \fi\fi\fi%
9751 }
9752 %

```

XXXVII 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.³⁸

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.³⁹

```

9753 \def\Hy@raisedlink@left#1{%
9754   \ifvmode
9755     #1%
9756   \else

```

³⁸<http://tex.stackexchange.com/a/17138/7712>.

³⁹The code is inspired by an answer given by @unbonpetit. Thanks to him. <http://texnique.fr:80/osqa/questions/781/hyraisedlink-perturbe-la-maniere-dont-se-fait-la-coupeure-de-ligne/801>.

```

9757 \Hy@SaveSpaceFactor
9758 \llap{\smash{%
9759 \begingroup
9760 \let\HyperRaiseLinkLength\@tempdima
9761 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
9762 \HyperRaiseLinkHook
9763 \expandafter\endgroup
9764 \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
9765 \Hy@RestoreSpaceFactor
9766 #1%
9767 \Hy@SaveSpaceFactor
9768 }%
9769 }}%
9770 \Hy@RestoreSpaceFactor
9771 \penalty\@M\hskip\z@\relax
9772 \fi
9773 }
9774 %

```

XXXVIII Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

9775 \ifeledmaccompat{%
9776
9777 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
9778 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
9779 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
9780 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
9781
9782 \unless\ifnocritical@
9783 \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
9784 \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
9785 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
9786 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
9787 \let\hsizetwocol\Xhsizetwocol
9788 \let\hsizethreecol\Xhsizethreecol
9789 \let\bhookXnote\Xbhooknote
9790 \let\boxsymlinenum\Xboxsymlinenum
9791 \let\symlinenum\Xsymlinenum
9792 \let\beforenumberinfootnote\Xbeforenumber
9793 \let\afternumberinfootnote\Xafternumber
9794 \let\beforeXsymlinenum\Xbeforesymlinenum
9795 \let\afterXsymlinenum\Xaftersymlinenum
9796 \let\inplaceofnumber\Xinplaceofnumber
9797 \let\Xlemmaseparator\lemmaseparator
9798 \let\afterlemmaseparator\Xafterlemmaseparator
9799 \let\beforelemmaseparator\Xbeforelemmaseparator
9800 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator

```

```

9801 \let\txtbeforeXnotes\Xtxtbeforenotes
9802 \let\afterXrule\Xafterrule
9803 \let\numberonlyfirstinline\Xnumberonlyfirstinline
9804 \let\numberonlyfirstintwlines\Xnumberonlyfirstintwlines
9805 \let\nonumberinfootnote\Xnonumberinfootnote
9806 \let\pstartinfootnote\Xpstart
9807 \let\pstartinfootnoteeverytime\Xpstarteverytime
9808 \let\onlyXpstart\Xonlypstart
9809 \let\Xnonumberinfootnote\Xnonumber
9810 \let\Xnonbreakableafternumber\Xnonbreakableafternumber
9811 \let\maxhXnotes\Xmaxhnotes
9812 \let\beforeXnotes\Xbeforenotes
9813 \let\boxlinenum\Xboxlinenum
9814 \let\boxlinenumalign\Xboxlinenumalign
9815 \let\boxstartlinenum\Xboxstartlinenum
9816 \let\boxendlinenum\Xboxendlinenum
9817 \let\twlines\Xtwlines
9818 \let\morethantwlines\Xmorethantwlines
9819 \let\twlinesbutnotmore\Xtwlinesbutnotmore
9820 \let\twlinesonlyinsamepage\Xtwlinesonlyinsamepage
9821 \fi
9822
9823 \unless\ifnofamiliar@
9824 \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
9825 \fi
9826 \newcommandx{\parafootsep}[2][1,usedefault]{%
9827 \Xparafootsep[#1]{#2}%
9828 \parafootsepX[#1]{#2}
9829 }%
9830
9831 \newcommandx{\afternote}[2][1,usedefault]{%
9832 \Xafternote[#1]{#2}%
9833 \afternoteX[#1]{#2}%
9834 }%
9835
9836 \unless\ifnoend@
9837 \let\XendXtwlines\Xendtwlines
9838 \let\XendXmorethantwlines\Xendmorethantwlines
9839 \let\XhookXendnote\Xendhooknote
9840 \let\XboxXendlinenum\Xendboxlinenum%
9841 \let\XboxXendlinenumalign\Xendboxlinenumalign%
9842 \let\XboxXendstartlinenum\Xendboxstartlinenum%
9843 \let\XboxXendendlinenum\Xendboxendlinenum%
9844 \let\XendXlemmaseparator\Xendlemmaseparator
9845 \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
9846 \let\XendXafterlemmaseparator\Xendafterlemmaseparator
9847 \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
9848 \fi
9849
9850 \AtBeginDocument{%

```

```
9851 \ifdef\lineref{}\let\lineref\edlineref}%  
9852 }%  
9853  
9854  
9855 \fi%  
9856 %
```

</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 changed⁴⁰ 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
```

⁴⁰A name like `\text` is likely to be defined by other \LaTeX packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.

```
\let<=\critext
```

Then you might say `<{Smith}\variant{Jones}/`. This of course destroys the ability to use `<` in any new macro definitions, so long as it remains in effect; hence it should be used with care.

Changing the character at the end of the command requires more work:

```
\catcode`\<=\active
\def\xtext#1#2>{\critext{#1}{#2}/}
\let<=\xtext
```

This allows you to say `<{Smith}\Afootnote{Jones}>`.

Aliases for `\critext` of the first kind shown here also can't be nested—that is, you can't use the alias in the text that forms the first argument to `\critext`. (See VI p. 133 to find out why.) Aliases of the second kind may be nested without any problem.

If you really have to use `\critext` in any of the tabular or array environments, then `\edtext` must not be used in the same environment. If you use `\critext` in one of these environments then you have to issue the declaration `\usingcritext` beforehand. The declaration `\usingedtext` must be issued to revert to the default assumption that `\edtext` will be used.

Appendix A.2 Migration from ledmac to eledmac

In `eledmac`, some changes were made in the code to allow easy customization. This may cause problems for people who have already made their own. The next sections explain how to handle this.

If you have created your own series using `\addfootins` and `\addfootinsX`, you must use instead the `\newseries` command (see 6.7.1 p. 36), 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. 38) or `\Xfootnote` options (6.2.2 p. 25). Otherwise please add a new ticket on Github to request a new function for doing this.⁴¹

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]
```

⁴¹<https://github.com/maieul/ledmac/issues>

If you do not make that, you will get a spurious [X], where X is series letter.

If you used a `\protect` command inside a `\footnote` command inside a numbered section, you must change the `\protect` to `\noexpand`. Otherwise the command after the `\protect` will be discarded.

Appendix A.3 Migration to eledmac 1.5.1

The version 1.5.1 corrects a bug in `stanzaindentsrepetition` (cf. 9.3 p. 52). This bug had two consequences:

1. `stanzaindentsrepetition` did not work when its value was greater than 2.
2. `stanzaindentsrepetition` worked wrong when its value was equal to 2.

So, if you used `stanzaindentsrepetition` with a value equal to 2, you had to change your `\setstanzaindents`. Explanation:

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

This code, in versions prior to 1.5.1, made the first line have an indentation of 0, the second line of 1, the third verse of 0, the fourth verse of 1 and so forth.

But this code should have instead achieved quite the contrary: the first line would have an indentation of 1, the second line of 0, the third line of 1, the fourth line of 0 and so forth.

So version 1.5.1 corrected this bug. If you want to keep the former presentation, you must change:

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

to:

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

Appendix A.4 Migration to eledmac 1.12.0

The migration to eledmac 1.12.0 is easy:

- You must first delete all the auxiliary files, then compile your document three times as usual.
- If you have modified `\l@reg`, which is not advisable, you must rename it to `\@nl@reg`.

There is an additional problem. If you have put text into brackets just after `\pstart` or `\pend`, this text will be considered to be an optional argument of `\pstart` or `\pend` (see 5.2.3 p. 18). If so, add a `\relax` between `\pstart`/`\pend` and the first bracket.

The version 1.12.0 also introduce a better way to handle sectional divisions inside numbered text. Please read 16.2 p. 72.

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. 74 in order to know how to limit it.

Appendix A.9.2 Multiple indices with memoir

Eledmac and ledmac used the specific indexing tools of the memoir class designed to produce multiple indices. However, eledmac could also use imakeidx or indextools tools independently of the memoir class. This system forced to maintain redundant code. Since reledmac, we use only the imakeidx or indextools tools.

Consequently: Users of memoir are invited to use indextool or imakeidx to produce multiple indices.

Appendix A.9.3 Deprecated commands and options

The table of deprecated commands and their alternatives follows. Note that the way some commands must be used may have changed. Please read the handbook.

<i>Deprecated command</i>	<i>Replaced with</i>
<code>\addfootins</code>	<code>\newseries</code>
<code>\addfootinsX</code>	<code>\newseries</code>
<code>\critext</code>	<code>\edtext</code>
<code>\falseverse</code>	<code>\newverse</code>
<code>\interparanoteglue</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\ledchapter</code>	<code>\eledchapter</code>
<code>\ledsection</code>	<code>\eledsection</code>
<code>\ledsetnormalparstuff</code>	<code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code>
<code>\ledsubsection</code>	<code>\eledsubsection</code>
<code>\ledsubsubsection</code>	<code>\eledsubsubsection</code>
<code>\noeledsec</code>	Package option <code>noeledsec</code>
<code>\noendnotes</code>	Package option <code>noendnotes</code>
<code>\pageparbreak</code>	<code>\ledpb</code>

The `ledsecnolinenumber` option has been removed, because it was related to deprecated commands.

The `oldprintnpnumspace` option has been removed too, because it was related to a historical bug. The `\usingedtext` and `\usingcritext` commands are also deprecated.

Appendix A.9.4 \renewcommand replaced by command

Many uses of \renewcommand have been replaced with uses of specific commands. Please read handbook about specific commands.

<i>Deprecated \renewcommand</i>	<i>Replaced with</i>
\@led@extranofeet	\newseries
\apprefprefixmore	\setapprefprefixmore
\apprefprefixsingle	\setapprefprefixsingle
\endstanzaextra	Optional argument of \&
\hangingsymbol	\sethangingsymbol
\ledfootinsdim	\Xmaxhnotes and \maxhnotesX
\parafootftmsep	\Xparafootsep and \parafootsepX
\notenumfont	\Xnotenumfont, \Xendnotenumfont and \notenumfontX
\notefontsetup	\Xnotefontsize, \Xendnotefontsize and \notefontsizeX
\sidenoteseq	\setsidenotsep
\startstanzahook	Optional argument of \stanza
\symplinenum	\Xsymplinenum

Appendix A.9.5 Commands the names of which have been changed

In order to help the migration from eledmac to reledmac, you may load reledmac with eledmac-compat option. However, it is advised not to, and to change the command names themselves instead. In many cases, you use only a few of them, except the \footparagraph command.

<i>Old command</i>	<i>New command</i>
\footparagraph	\Xarrangement
\footnormal	\Xarrangement
\foottwocol	\Xarrangement
\footthreecol	\Xarrangement
\footparagraphX	\arrangementX
\footnormalX	\arrangementX
\foottwocolX	\arrangementX
\footthreecolX	\arrangementX
\afterlemmaseparator	\Xafterlemmaseparator
\afternote	\Xafternote and \afternoteX
\afternumberinfootnote	\Xafternumber
\afterXrule	\Xafterrule
\afterXsymplinenum	\Xaftersymplinenum
\beforelemmaseparator	\Xbeforelemmaseparator
\beforenumberinfootnote	\Xbeforenumber
\beforeXnotes	\Xbeforenotes
\beforeXsymplinenum	\Xbeforesymplinenum

<i>Old command</i>	<i>New command</i>
<code>\bhookXnote</code>	<code>\Xbhookendnote</code>
<code>\bhookXnote</code>	<code>\Xbhooknote</code>
<code>\boxendlinenum</code>	<code>\Xboxendlinenum</code>
<code>\boxlinenum</code>	<code>\Xboxlinenum</code>
<code>\boxlinenumalign</code>	<code>\Xboxlinenumalign</code>
<code>\boxstartlinenum</code>	<code>\Xboxstartlinenum</code>
<code>\boxsymlinenum</code>	<code>\Xboxsymlinenum</code>
<code>\boxXendlinenum</code>	<code>\Xendboxlinenum</code>
<code>\boxXendlinenumalign</code>	<code>\Xendboxlinenumalign</code>
<code>\boxXendstartlinenum</code>	<code>\boxXendstartlinenum</code>
<code>\letboxXendendlinenum</code>	<code>\Xendletboxendlinenum</code>
<code>\hsizetwocol</code>	<code>\Xhsizetwocol</code>
<code>\hsizethreecol</code>	<code>\Xhsizethreecol</code>
<code>\inplaceoflemmaseparator</code>	<code>\Xinplaceoflemmaseparator</code>
<code>\inplaceofnumber</code>	<code>\Xinplaceofnumber</code>
<code>\lemmaseparator</code>	<code>\Xlemmaseparator</code>
<code>\maxhXnotes</code>	<code>\Xmaxhnotes</code>
<code>\morethantwolines</code>	<code>\Xmorethantwolines</code>
<code>\nonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\notesXwidthliketwocolumns</code>	<code>\noteswidthliketwocolumnsX</code>
<code>\noXlemmaseparator</code>	<code>\Xnolemmaseparator</code>
<code>\numberonlyfirstinline</code>	<code>\Xnumberonlyfirstinline</code>
<code>\numberonlyfirstintwolines</code>	<code>\Xnumberonlyfirstintwolines</code>
<code>\nonbreakableafternumber</code>	<code>\Xnonbreakableafternumber</code>
<code>\onlyXpstart</code>	<code>\Xonlypstart</code>
<code>\parafootsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\pstartinfootnote</code>	<code>\Xpstart</code>
<code>\pstartinfootnoteeverytime</code>	<code>\Xpstarteverytime</code>
<code>\symlinenum</code>	<code>\Xsymlinenum</code>
<code>\twolines</code>	<code>\Xtwolines</code>
<code>\twolinesbutnotmore</code>	<code>\Xtwolinesbutnotmore</code>
<code>\twolinesonlyinsamepage</code>	<code>\Xtwolinesonlyinsamepage</code>
<code>\txtbeforeXnotes</code>	<code>\Xtxtbeforenotes</code>
<code>\XendXafterlemmaseparator</code>	<code>\Xendafterlemmaseparator</code>
<code>\XendXbeforelemmaseparator</code>	<code>\Xendbeforelemmaseparator</code>
<code>\XendXinplaceoflemmaseparator</code>	<code>\Xendinplaceoflemmaseparator</code>
<code>\XendXlemmaseparator</code>	<code>\Xendlemmaseparator</code>
<code>\XendXmorethantwolines</code>	<code>\Xendmorethantwolines</code>
<code>\XendXtwolines</code>	<code>\Xendtwolines</code>
<code>\Xnonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\lineref</code>	<code>\edlineref</code>

Appendix A.9.6 Endnotes

With *reledmac*, there is now one auxiliary file for every endnotes set (*.Aend*, *.Bend*, *.Cend* etc.). If you have overridden `\doendnotes` (which you would not have done) you must adapt your code.

Appendix A.9.7 Z Series

The ‘Z’ series of notes has been removed. Only five series are provided now by default: A, B, C, D, E.

Appendix A.9.8 Internal commands

Users who have overridden internal commands, which is wrong, must adapt according to the following. Or better, they should not override any of such commands and use *reledmac* options instead.

- If you have modified `\Xfootfmt`, note that the fourth argument is now mandatory.
- `\unvxh` has been replaced with `\Xunvxh` and `\unvxhX` with two mandatory arguments.

Appendix A.10 Migration to *reledmac* 2.1.0

Reledmac 2.1.0 fix some bugs when using `\Xbhooknote` and `\bhooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }.`
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` in the argument of `\Xbhooknote/\bhooknoteX`.

Appendix A.11 Migration to *reledmac* 2.1.3

Reledmac 2.1.3 fix an historical bug, (style in *ledmac* 0.7!) which doubled the space before the rules of paragraphed familiar footnotes. Consequently, if you use paragraphed familiar footnotes, you should maybe adapt it, playing with `\beforenotesX`.

Appendix A.12 Migration to *reledmac* 2.3.0

Before *reledmac* 2.3.0, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use optional argument of `\stanza` or `\endverse`.

Appendix A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{\langle...}\rangle}` to the use of `\Xlinerangeseparator` or `/` and `\Xendlinerangeseparator` (7.2.4 p. 40).

Appendix A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 7.2.3 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.

Appendix A.19 Migration to reledmac 2.18.0

After updating reledmac, and before any new compilation, you need to clean your `.aux` files, if you use `\edlabel` or related.

Appendix A.20 Migration to reledmac 2.21.0

Previously, there was a bug, which meant that the description in the handbook was incorrect. If you wrote

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

“elephant” was indexed in the main text and in the critical footnotes. With the new version of reledmac, it is indexed only in main text. If you also want to index it in critical footnotes, do

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

Appendix A.21 Migration to reledmac 2.24.0

When using `\labelpstarttrue`, a spurious space^a was introduced after the `pstart` number (only for normal typesetting, not while typesetting in parallel). The new version of the package has deleted this spurious space. If you consider that it was NOT a spurious space, you should add it manually in your definition of `\thepstart`.

Appendix A.22 Migration to reledmac 2.26.0

You must delete your `.aux` file after having upgraded to this new version of reledmac.

References

- [Bre96] Herbert Breger. `tabmac`. October 1996. (Available from CTAN in `macros/plain/contrib/tabmac`)
- [Bur01] John Burt. ‘Typesetting critical editions of poetry’. *TUGboat*, **22**, 4, pp 353–361, December 2001. (Code available from CTAN in `macros/latex/contrib/poemscol`)
- [Eck03] Matthias Eckermann. *The Parallel-Package*. April 2003. (Available from CTAN in `macros/latex/contrib/parallel`)
- [Fai03] Robin Fairbairns. *footmisc—a portmanteau package for customising footnotes in L^AT_EX*. February 2003. (Available from CTAN in `macros/latex/contrib/footmisc`)
- [LW90] John Lavagnino and Dominik Wujastyk. ‘An overview of edmac: a PLAIN TeX format for critical editions’. *TUGboat*, **11**, 4, pp. 623–643, November 1990. (Code available from CTAN in `macros/plain/contrib/edmac`)
- [Lüc03] Uwe Lück. ‘ednotes—critical edition typesetting with LaTeX’. *TUGboat*, **24**, 2, pp. 224–236, 2003. (Code available from CTAN in `macros/latex/contrib/ednotes`)
- [Sul92] Wayne G. Sullivan. *The file edstanza.doc*. June 1992. (Available from CTAN in `macros/plain/contrib/edmac`)
- [Wil02] Peter Wilson. *The memoir class for configurable typesetting*. November 2002. (Available from CTAN in `macros/latex/contrib/memoir`)
- [Wil04] Peter Wilson and Maïeul Rouquette. *Parallel typesetting for critical editions: the eledpar package*. December 2004. (Available from CTAN in `macros/latex/contrib/ledmmac`)

Index

Symbols

<code>\&</code>	51
<code>\@EDROWFILL@</code>	1
<code>\@adv</code>	1
<code>\@advancestanzanumber</code>	1
<code>\@annot</code>	1
<code>\@annot@end@print</code>	1
<code>\@annot@start@print</code>	1
<code>\@beforeinsertofthisedtext</code>	1
<code>\@docclearpage</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
<code>\@emptytoks</code>	1
<code>\@fnpos</code>	1
<code>\@footnotemark</code>	1
<code>\@footnotetext</code>	1
<code>\@getfirstseries</code>	1
<code>\@gobblefour</code>	1
<code>\@gobbleseven</code>	1
<code>\@gobblethree</code>	1
<code>\@h</code>	1
<code>\@changingsymbol</code>	1
<code>\@iiiminipage</code>	1
<code>\@insertstanzaNumber</code>	1
<code>\@k</code>	1
<code>\@l@dtmPCnta</code>	1
<code>\@l@dtmPCntb</code>	1
<code>\@lab</code>	1
<code>\@led@testifnoFoot</code>	1
<code>\@lemma</code>	1
<code>\@line@Cnum</code>	1
<code>\@lock</code>	1
<code>\@lopL</code>	1
<code>\@lopR</code>	1
<code>\@makechapterhead</code>	1
<code>\@makeschapterhead</code>	1
<code>\@mem@extranOfFeet</code>	1
<code>\@mem@old@ssect</code>	1
<code>\@mpfnpos</code>	1
<code>\@msd</code>	1
<code>\@msd@c</code>	1
<code>\@msd@options@ifFullPage</code>	1
<code>\@msdata@list</code>	1
<code>\@nl</code>	1
<code>\@nl@reg</code>	1
<code>\@opXfeet</code>	1
<code>\@outputpage</code>	1
<code>\@page</code>	1
<code>\@pend</code>	1
<code>\@pendR</code>	1
<code>\@ref</code>	1
<code>\@ref@reg</code>	1
<code>\@ref@reg@parse</code>	1
<code>\@sect</code>	1
<code>\@series</code>	1
<code>\@set</code>	1
<code>\@sidenotesep</code>	1
<code>\@ssect</code>	1
<code>\@startstanza</code>	1

\@stopmsd	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1
\@xloop	1
\@xympar	1
CLASSarticle	74
CLASSbook	74, 366
CLASSmemoir	215, 216, 278–281, 316, 366, 386, 447, 451
CLASSscrbook	451
COMMAND*footnote	76
COMMAND\...@footnotemark...	219
COMMAND\...d@ta	161
COMMAND\<hook	
@<series	266
COMMAND\<hookname	
<pseudoserries	268, 269
COMMAND\<type	
footfmt	204
COMMAND\@@line	195
COMMAND\@MM	182, 448
COMMAND\@Rlineflag	318, 448
COMMAND\@Serefprefix	295
COMMAND\@Serefprefixmore	295
COMMAND\@add@	356
COMMAND\@adv	116
COMMAND\@annot	167
COMMAND\@annot@end@print	207
COMMAND\@annot@start@print	207
COMMAND\@apprefprefixmore	295
COMMAND\@apprefprefixsingle	295
COMMAND\@beforeinsertofthisedtext	139, 140
COMMAND\@bsphack	283
COMMAND\@doclearpage	279, 280, 441, 451
COMMAND\@doreinfeetX	451
COMMAND\@dprintingcolumns	448
COMMAND\@edindex@hyperref	318, 319
COMMAND\@edtext@level	136
COMMAND\@esphack	283
COMMAND\@firstofone	33
COMMAND\@fnpos	237, 275
COMMAND\@footnotemark	216, 441, 451
COMMAND\@footnotetext	216, 217, 441
COMMAND\@gobble	32, 58, 134, 135, 167, 261
COMMAND\@gobblefive	450
COMMAND\@gobblefour	447
COMMAND\@gobbleseven	263
COMMAND\@gobblethree	440
COMMAND\@h	199

COMMAND\@hangingsymbol	323
COMMAND\@iiiminipage	307, 309, 440, 451
COMMAND\@iiiminpage	307
COMMAND\@l	445
COMMAND\@l@tempcnta	164, 168, 176
COMMAND\@l@tempcntb	168
COMMAND\@l@reg	445
COMMAND\@lab	112, 283, 286, 291, 440
COMMAND\@ldunboxmpfoot	310
COMMAND\@led@extranofeet	387
COMMAND\@ledinnote@command	314, 315
COMMAND\@lemma	139, 141
COMMAND\@lock	106, 324
COMMAND\@lopL	441
COMMAND\@lopR	441
COMMAND\@makecol	275, 276, 278, 451
COMMAND\@mempnum	281
COMMAND\@mpfnpos	237
COMMAND\@msd	333
COMMAND\@msd@c	333
COMMAND\@msd@options@iffullpage	339
COMMAND\@msdata@list	333, 334
COMMAND\@nl	112, 114, 115, 117, 127, 167, 286, 440, 441
COMMAND\@nl@reg	112, 384, 441, 445
COMMAND\@opXfeet	441
COMMAND\@opfeetX	451
COMMAND\@opxtrafeeti	451
COMMAND\@outputpage	281, 282
COMMAND\@page	112, 114, 286
COMMAND\@pend	441
COMMAND\@pendR	441
COMMAND\@ref	112, 122–124, 128, 129, 134
COMMAND\@ref@later	123, 124, 130
COMMAND\@ref@reg	122, 167, 441
COMMAND\@ref@reg@parsearg	123, 124
COMMAND\@reinserts	275–278, 451
COMMAND\@secondoftwo	76
COMMAND\@sect	368
COMMAND\@series	265
COMMAND\@set	117
COMMAND\@sidenotesep	306
COMMAND\@stopmsd	334
COMMAND\@sw	124, 143, 147, 148
COMMAND\@tag	136, 137, 141
COMMAND\@tempcnta	90
COMMAND\@tempcntb	90
COMMAND\@toksa	97
COMMAND\@toksb	97
COMMAND\@xloop	177
COMMAND\@xympar	299, 451

COMMAND\Aendnote	16, 25
COMMAND\Afootfmt	181
COMMAND\Afootgroup	181
COMMAND\Afootnote	8, 16, 24, 25, 28, 138, 190, 215, 238, 255, 450
COMMAND\Afootstart	181
COMMAND\AtBeginDocument	278
COMMAND\AtEndEveryPend	19, 457
COMMAND\AtEveryPend	18, 19, 54, 154, 447, 449, 451, 457
COMMAND\AtEveryPend*	19
COMMAND\AtEveryPstart	18, 19, 54, 447, 449, 451, 454, 457
COMMAND\AtEveryPstart*	19
COMMAND\AtEveryStanza	54, 455, 457
COMMAND\AtEveryStopStanza	55, 455, 457
COMMAND\AtStartEveryPstart	19, 457
COMMAND\AtStartEveryStanza	54, 55, 457
COMMAND\BeforeEveryStopStanza	457
COMMAND\Bendnote	16, 24
COMMAND\Bfootnote	8, 16, 215, 238, 255
COMMAND\Centering	47
COMMAND\Cfootnote	215
COMMAND\Columns	90, 150, 187
COMMAND\Dfootnote	215
COMMAND\Efootnote	215
COMMAND\Gls	68
COMMAND\Hy@raisedlink	378
COMMAND\Hy@raisedlink@left	378
COMMAND\LTR	47
COMMAND\NR	368
COMMAND\Pages	90, 150, 276, 277, 281
COMMAND\ProcessOptionsX	81
COMMAND\RL	46
COMMAND\RaggedLeft	47
COMMAND\RaggedRight	47
COMMAND\RenewExpandableDocumentCommand	33, 146
COMMAND\SEonlypage	293, 453
COMMAND\SEref	61–63, 293, 296, 454, 456
COMMAND\SErefonlypage	61–63, 390, 453
COMMAND\SErefwithpage	61, 63, 293, 296, 390, 453, 455
COMMAND\Stanza	446
COMMAND\Waklam	357
COMMAND\X@doreinfeet	277, 451
COMMAND\XR@prefix	299
COMMAND\XR@test	299
COMMAND\XR@test@mac	299
COMMAND\XR@test@mac@test	299
COMMAND\XXXXXXfmt	383
COMMAND\XXXXXXfmt	383
COMMAND\Xafterrenumber	42
COMMAND\Xafterlemmaseparator	44, 387
COMMAND\Xafternote	48, 386, 387

COMMAND\Xafternumber	42, 387
COMMAND\Xafterrule	49, 238, 387, 446, 449
COMMAND\Xaftersymlinenum	42, 43, 387
COMMAND\Xarrangement	38, 48, 76, 182, 183, 267, 387
COMMAND\Xarrangement@footparagraph	188
COMMAND\Xarrangement@normal	183
COMMAND\Xarrangement@paragraph	188
COMMAND\Xbeforeinserting	47
COMMAND\Xbeforelemmaseparator	44, 387
COMMAND\Xbeforenotes	49, 238, 387, 446, 449
COMMAND\Xbeforenumber	40, 42, 387
COMMAND\Xbeforesymlinenum	42, 43, 387
COMMAND\Xbhookendnote	388
COMMAND\Xbhookgroup	49, 453, 454
COMMAND\Xbhooknote	47, 388, 389, 451, 452
COMMAND\Xboxendlinenum	43, 44, 388, 450
COMMAND\Xboxlinenum	43, 44, 388
COMMAND\Xboxlinenumalign	43, 44, 388, 450
COMMAND\Xboxstartlinenum	43, 44, 388, 450
COMMAND\Xboxsymlinenum	43, 388
COMMAND\Xcolalign	47, 449
COMMAND\Xdo@feet	451, 456
COMMAND\Xend	263
COMMAND\XendXafterlemmaseparator	388
COMMAND\XendXbeforelemmaseparator	388
COMMAND\XendXinplaceoflemmaseparator	388
COMMAND\XendXlemmaseparator	388
COMMAND\XendXmorethantwolines	388
COMMAND\XendXtwolines	388
COMMAND\Xendafterenumber	42, 452
COMMAND\Xendafterlemmaseparator	45, 388
COMMAND\Xendafternote	51, 454
COMMAND\Xendafternumber	44
COMMAND\Xendafterpagenumbe	455
COMMAND\Xendafterpagenumber	40, 63
COMMAND\XendafterpagenumberSErefonlypage	390
COMMAND\Xendaftersymlinenum	42, 44, 452
COMMAND\Xendahookinplaceofnumber	44, 452
COMMAND\Xendahooklinenum	44, 452
COMMAND\Xendbeforelemmaseparator	45, 388
COMMAND\Xendbeforelinenum	44
COMMAND\Xendbeforenumber	42, 452
COMMAND\Xendbeforepagenumber	40, 62, 63
COMMAND\XendbeforepagenumberSErefonlypage	62
COMMAND\Xendbeforesymlinenum	42, 44, 452
COMMAND\Xendbhookinplaceofnumber	44, 452
COMMAND\Xendbhooklinenum	44, 452
COMMAND\Xendbhooknote	47
COMMAND\Xendboxendlinenum	44, 450
COMMAND\Xendboxlinenum	44, 388, 448

COMMAND\Xendboxlinenumalign	44, 388, 450
COMMAND\Xendboxstartlinenum	44, 450
COMMAND\Xendboxsymlinenum	43, 452
COMMAND\Xendhangindent	46, 452, 454
COMMAND\Xendinplaceoflemmaseparator	26, 45, 388
COMMAND\Xendinplaceofnumber	43, 451
COMMAND\Xendinplaceofpagenumber	40, 456
COMMAND\Xendinsertsep@	245
COMMAND\Xendlemmadisablefontselection	45
COMMAND\Xendlemmafont	45, 453
COMMAND\Xendlemmaseparator	26, 44, 388
COMMAND\Xendletboxendlinenum	388
COMMAND\Xendlineflag	63
COMMAND\Xendlinenumannotationposition	57, 63
COMMAND\Xendlineprefixmore	40, 63
COMMAND\Xendlineprefixsingle	40, 63
COMMAND\Xendlinerangeseparator	40, 63, 202, 390, 452
COMMAND\Xendmorethantwolines	26, 41, 63, 388, 449, 450
COMMAND\Xendnonumber	41, 451
COMMAND\Xendnote	241, 261–263, 449
COMMAND\Xendnotefontsize	46, 387
COMMAND\Xendnotenumberfont	44, 45, 387
COMMAND\Xendnotes	244
COMMAND\Xendnumberonlyfirstinline	39, 452
COMMAND\Xendnumberonlyfirstintwolines	39, 452
COMMAND\Xendpagenumberonlyfirst	39, 456
COMMAND\Xendpagenumberonlyfirsttfsingle	39, 456
COMMAND\Xendpagenumberonlyfirstinttwo	40, 456
COMMAND\Xendparagraph	51, 446
COMMAND\Xendsep	51
COMMAND\Xendsublinesep	42, 63, 202
COMMAND\Xendsymlinenum	39, 452
COMMAND\Xendsympagenum	40, 456
COMMAND\Xendtwolines	26, 41, 63, 388, 449, 450
COMMAND\Xendtwolinesbutnotmore	41, 63, 449, 450
COMMAND\Xendtwolinesonlyinsamepage	41, 63, 449, 450
COMMAND\Xendtxtbeforenotes	49, 458
COMMAND\Xendwrapcontent	46, 455
COMMAND\Xendwraplemma	46, 455
COMMAND\Xendwraplinenumannotation	58, 63
COMMAND\Xfootfmt	389
COMMAND\Xfootgroup	187
COMMAND\Xfootins	186
COMMAND\Xfootnote	60, 66, 136, 383, 443, 447–449, 453, 455
COMMAND\Xfootstarts	187
COMMAND\Xgroupbyline	48, 174, 214, 458
COMMAND\Xgroupbylines	456
COMMAND\Xgroupbylineseparetwolines	48
COMMAND\Xhangindent	46, 452
COMMAND\Xsize	390, 453, 454

COMMAND\Xhsizethreecol	47, 50, 388
COMMAND\Xhsizetwocol	47, 50, 268, 388
COMMAND\Xinplaceoflemmaseparator	25, 44, 388
COMMAND\Xinplaceofnumber	43, 388, 448, 450
COMMAND\Xinsertparafootsep	193, 194
COMMAND\Xledsetnormalparstuff	385, 386, 449
COMMAND\Xlemmadisablefontselection	45
COMMAND\Xlemmafont	45, 453
COMMAND\Xlemmaseparator	44, 202, 269, 272, 274, 388
COMMAND\Xlineflag	62, 390, 391, 455
COMMAND\Xlinenumannotationposition	57, 63, 166
COMMAND\Xlinenumannotationposition@side	166
COMMAND\Xlinenumannotationpositionside	57
COMMAND\Xlinangeseparator	40, 63, 202, 390, 452
COMMAND\Xmaxhnotes	50, 74, 76, 387, 388, 446, 448
COMMAND\Xmorethantwolines	25, 40, 41, 63, 388, 448
COMMAND\Xnoindent	452
COMMAND\Xnolemmaseparator	44, 274, 388
COMMAND\Xnonbreakableafternumber	42, 388, 444
COMMAND\Xnonumber	41, 388
COMMAND\Xnonumberinfootnote	388
COMMAND\Xnotefontsize	45, 387
COMMAND\Xnotefontsize@<s>	193, 197, 198
COMMAND\Xnotenumfont	45, 387
COMMAND\Xnoteswidthliketwocolumns	50, 447
COMMAND\Xnumberonlyfirstinline	39, 40, 48, 109, 204, 268, 269, 272, 388, 443, 448, 457
COMMAND\Xnumberonlyfirstintwolines	39, 48, 388, 443, 457
COMMAND\Xonlypstart	41, 388, 443, 448
COMMAND\Xpagelinesep	42, 456
COMMAND\Xparafootsep	48, 109, 387, 388, 455, 457, 458
COMMAND\Xparafootsep@series	193
COMMAND\Xparinden	390
COMMAND\Xparindent	46, 449, 452, 454
COMMAND\Xprenotes	49, 238, 455
COMMAND\Xprenotes@	186, 238, 443
COMMAND\Xpstart	41, 385, 388, 443, 448
COMMAND\Xpstarteverytime	41, 385, 388, 448
COMMAND\Xragged	48
COMMAND\Xrigidbalance	195, 390, 453
COMMAND\Xstanza	42, 55
COMMAND\Xstanzaseparator	42
COMMAND\Xstorelineinfo	204
COMMAND\Xsublinesep	22, 42, 63, 202, 390
COMMAND\Xsublinesepside	22, 42
COMMAND\Xsymlinenum	39, 48, 387, 388, 450, 457
COMMAND\Xtextbeforenotes	177
COMMAND\Xtoendnotes	27, 262
COMMAND\Xtwolines	25, 40, 41, 63, 210, 211, 268, 388, 448
COMMAND\Xtwolinesappref	268
COMMAND\Xtwolinesbutnotmore	41, 63, 388, 449

COMMAND\Xtwolinesbutnotmoreappref	269
COMMAND\Xtwolinesonlyinsamepage	41, 63, 388, 449
COMMAND\Xtxtbeforenotes	49, 388, 455, 456, 458
COMMAND\Xtxtbeforenotesonlyonce	49, 458
COMMAND\Xunvxh	191, 389
COMMAND\Xwidth	50, 390, 454
COMMAND\Xwrapcontent	46, 455
COMMAND\Xwraplemma	46, 47, 455
COMMAND\Xwraplinenumannotation	57, 58, 63, 167
COMMAND\Xwraplinenumannotation@ref	167
COMMAND\Xwraplinenumannotation@side	167
COMMAND\Xwraplinenumannotationref	58
COMMAND\Xwraplinenumannotationside	57
COMMAND\&	387
COMMAND\absline@num	105, 162
COMMAND\accent	135
COMMAND\actionlines@list	106, 164
COMMAND\actions@list	106
COMMAND\add@Xgroupbyline	174
COMMAND\add@inserts	106, 174
COMMAND\add@inserts@next	174
COMMAND\add@msd@	333
COMMAND\add@msdata	333, 334
COMMAND\add@msdata@firstlineofpage	337
COMMAND\add@msddata	333
COMMAND\add@penalties	162, 175
COMMAND\addcontentsline	364
COMMAND\addfootins	383, 386
COMMAND\addfootinsX	383, 386
COMMAND\addtoendnotes	262
COMMAND\advancelabel@refs	285
COMMAND\advanceline	23, 108, 116, 131, 451
COMMAND\affixlin@num	306
COMMAND\affixline@num	168, 171, 172, 441
COMMAND\affixpstart@num	172
COMMAND\afterXrule	387
COMMAND\afterXsymlinenum	387
COMMAND\aftergroup	134, 138
COMMAND\afterlemmaseparator	387
COMMAND\afternote	387
COMMAND\afternoteX	48, 386, 387
COMMAND\afternumberinfootnote	387
COMMAND\afterruleX	49, 446, 449
COMMAND\annot@list	108, 167
COMMAND\annotationref	57, 58
COMMAND\annotref	59
COMMAND\applabel	62, 287, 288, 449, 455
COMMAND\appref	60, 62, 63, 293, 296, 453, 454
COMMAND\apprefprefixmore	62, 387
COMMAND\apprefprefixsingle	62, 387

COMMAND\apprefwithpage	62, 63, 293, 296, 450, 453
COMMAND\arrangementX	38, 76, 219, 267, 387
COMMAND\arrangementX@normal	225
COMMAND\article	15
COMMAND\at@every@pend	154
COMMAND\autopar	18, 151, 155, 156, 235, 442, 444, 445, 449, 458
COMMAND\ballast	75
COMMAND\ballast@count	162, 176
COMMAND\baselineskip	38, 189, 193
COMMAND\beforeXnotes	387
COMMAND\beforeXsymlinenum	387
COMMAND\beforeedchapter	10, 72, 365
COMMAND\beforeinsertingX	47
COMMAND\beforelemmaseparator	387
COMMAND\beforenotesX	49, 389, 445, 446, 449
COMMAND\beforenumberinfotnote	387
COMMAND\begin	341
COMMAND\beginnumbering	17–20, 91, 92, 95, 96, 105, 109, 110, 125, 126, 155, 241, 332, 443, 446, 450, 451, 456, 457
COMMAND\beginnumberingR	91
COMMAND\bf	443
COMMAND\bfseries	45, 443
COMMAND\bhookXnote	388
COMMAND\bhookgroupX	49, 453
COMMAND\bhooknoteX	47, 389, 451, 452
COMMAND\body	324
COMMAND\bodyfootmarkA	35
COMMAND\book	15
COMMAND\boxXendlinenum	388
COMMAND\boxXendlinenumalign	388
COMMAND\boxXendstartlinenum	388
COMMAND\boxendlinenum	388
COMMAND\boxlinefootnote	206
COMMAND\boxlinenum	388
COMMAND\boxlinenumalign	388
COMMAND\boxstartlinenum	388
COMMAND\boxsymlinenum	388
COMMAND\break	39, 191
COMMAND\brokenpenalty	176
COMMAND\c@page	127
COMMAND\c@page@page	127
COMMAND\centering	47
COMMAND\ch@ck@l@ck	441
COMMAND\ch@cksub@l@ck	171, 441
COMMAND\chapter	71, 366, 370, 446, 449, 451, 455
COMMAND\chaptermark	364
COMMAND\check@pb@in@verse	378
COMMAND\colalignX	47, 449
COMMAND\collect@body	341
COMMAND\color	455

COMMAND\colorbox	76
COMMAND\columns	50
COMMAND\columnwidth	189, 447
COMMAND\command names	268
COMMAND\continuousnumberingwithcolumns@sync@linenumber@singletext	165
COMMAND\copyright	135
COMMAND\correct@Xfootins@box	448
COMMAND\correct@footinsX@box	448
COMMAND\count	196, 197
COMMAND\critex	442
COMMAND\critext	141, 382, 383, 386
COMMAND\csname	81, 145
COMMAND\csquotes	259
COMMAND\ctab	357, 362
COMMAND\ctabtext	362
COMMAND\current@annot	167
COMMAND\dcoll	351
COMMAND\def	78, 94
COMMAND\detokenize	145
COMMAND\dimen	197
COMMAND\dimexpr	50
COMMAND\discretionary	190
COMMAND\displaywidowpenalty	175
COMMAND\do@Xfeet	276, 441, 451, 456
COMMAND\do@actions	162–164, 441
COMMAND\do@actions@fixedcode	441
COMMAND\do@actions@next	163, 164
COMMAND\do@ballast	162, 176
COMMAND\do@feet@custom@order	275
COMMAND\do@insidelinehook	444
COMMAND\do@line	106, 133, 153, 157, 160, 161, 174, 175, 324, 441, 442, 444, 446
COMMAND\do@linehook	441
COMMAND\do@lockoff	107
COMMAND\do@lockon	107
COMMAND\dodoreintrafeet	440
COMMAND\doendnotes	26, 245, 389, 450, 457
COMMAND\doendnotesbysection	26, 245, 263, 450, 457
COMMAND\doennotes	457
COMMAND\doinsidelinehook	24, 446
COMMAND\dolinehook	24, 446
COMMAND\doreintrafeeti	451
COMMAND\doreintrafeetii	451
COMMAND\doxtrafeet	275, 440
COMMAND\doxtrafeeti	451
COMMAND\doxtrafeetii	451
COMMAND\dummy@ref	134
COMMAND\edaftertab	70, 357, 358
COMMAND\edannotationref	290
COMMAND\edatleft	70, 355
COMMAND\edatright	70, 71, 355

COMMAND\edbeforetab	70, 357
COMMAND\edfilldimen	356
COMMAND\edfont@info	141
COMMAND\edgls	68, 313
COMMAND\edglsadd	457
COMMAND\edgls...	454
COMMAND\edindex	33, 66–68, 313, 314, 317, 319, 346, 444, 446, 448, 451, 452, 456–458
COMMAND\edindexlab	67
COMMAND\edlabel	59– 62, 64, 135, 283, 285, 286, 289, 290, 298, 313, 346, 378, 391, 440, 443–445, 448, 453, 458
COMMAND\edlabelE	61, 288
COMMAND\edlabelS	61, 288
COMMAND\edlabelSE	61
COMMAND\edlineref	59, 283, 388, 448, 450, 453, 457
COMMAND\edmakelabel	61, 298
COMMAND\edpageref	59, 283, 289, 298
COMMAND\edrowfill	357
COMMAND\edsublineref	59
COMMAND\edtabcolsep	350
COMMAND\edtext	7, 24, 25, 27, 28, 30–32, 51, 56, 59, 60, 62, 65, 66, 68, 76, 94, 106, 108, 122, 124, 128, 129, 133–141, 144, 145, 147–149, 166, 167, 287, 288, 292, 346, 347, 366, 378, 382, 383, 386, 440, 442, 444, 446–450, 456–459
COMMAND\edtext@level	450
COMMAND\edtextlater	124
COMMAND\edvertdots	71, 356
COMMAND\edvertline	71, 355
COMMAND\elechapter	72
COMMAND\eled@sectioning@out	371
COMMAND\eledchapter	72, 386, 446, 451
COMMAND\eledchapter*	72
COMMAND\eledmac@error	440
COMMAND\eledsection	7, 16, 72, 134, 159, 365, 386, 448, 457, 459
COMMAND\eledsection*	72
COMMAND\eledsubsection	72, 386
COMMAND\eledsubsection*	72
COMMAND\eledsubsubsection	72, 386
COMMAND\eledsubsubsection*	72
COMMAND\eledxxx	10, 72, 371, 446
COMMAND\eledxxxx	364
COMMAND\elesection	371
COMMAND\else	312, 365
COMMAND\emph	33
COMMAND\empty	90, 169, 283
COMMAND\end	340, 341
COMMAND\end@lemmas	134
COMMAND\endashchar	202
COMMAND\endgraf	153, 193, 235
COMMAND\endlock	22, 107, 132, 328
COMMAND\endminipage	307, 310, 440, 451
COMMAND\endmsdata	35

COMMAND\endnotes	449, 453
COMMAND\endnumbering	17, 18, 20, 91, 92, 94, 95, 126, 441, 450, 456, 457
COMMAND\endnumberingR	91
COMMAND\endprint	241, 244, 263, 385
COMMAND\endstanzaextra	387
COMMAND\endsub	22, 107, 130
COMMAND\endverse	389
COMMAND\everypar	156
COMMAND\extensionchars	74, 91
COMMAND\externaldocument	64, 299
COMMAND\f@x@l@cks	441
COMMAND>falseverse	386, 444, 446
COMMAND\fi	365
COMMAND\firstlinenum	21, 168, 442
COMMAND\firstsublinenum	21, 442
COMMAND\flag@end	128, 129, 140, 446
COMMAND\flag@end@RTL	129
COMMAND\flag@end@later	130
COMMAND\flag@start	128, 129, 140, 446, 447
COMMAND\flag@start@RTL	129
COMMAND\flag@start@later	130
COMMAND\flagstanza	58
COMMAND\floatingpenalty	182, 448
COMMAND\flush@notes	176, 177
COMMAND\fnpos	37, 237, 445, 456
COMMAND\footfmt	181, 184
COMMAND\footfmt...	220
COMMAND\footfootmarkA	35
COMMAND\footfudgefactor	191
COMMAND\footfudgefiddle	75, 189, 440
COMMAND\footgroup	181
COMMAND\footins	186
COMMAND\footnormal	267, 387, 440
COMMAND\footnormalX	387
COMMAND\footnote	35, 75, 215–217, 384, 441
COMMAND\footnote@lang	202
COMMAND\footnoteA	16, 35
COMMAND\footnoteB	16
COMMAND\footnoteC	24
COMMAND\footnoteE	35
COMMAND\footnoteX	9, 36, 259, 261, 456
COMMAND\footnoteX@reading	260
COMMAND\footnoteXmark	261, 457
COMMAND\footnoteXmk	274
COMMAND\footnoteXtext	261, 457
COMMAND\footnote⟨X⟩	135
COMMAND\footnote⟨X⟩mark	36
COMMAND\footnote⟨X⟩mk	36
COMMAND\footnote⟨X⟩nomk	36
COMMAND\footnote⟨X⟩text	36

COMMAND\footnotelang@lua	180
COMMAND\footnotelang@poly	180
COMMAND\footnotemark	36, 261
COMMAND\footnoteoption@	179, 452
COMMAND\footnoterule	196
COMMAND\footnotesize	45, 46
COMMAND\footnotetext	36, 261
COMMAND\footparagraph	189, 267, 387, 446
COMMAND\footparagraphX	230, 387, 446
COMMAND\footsplitskips	441, 448
COMMAND\footstart	181, 186, 196
COMMAND\footstrut	193
COMMAND\footthreecol	387
COMMAND\footthreecolX	387, 449
COMMAND\foottwocol	387
COMMAND\foottwocolX	387, 449
COMMAND\foreignlanguage	46
COMMAND\fullstop	390
COMMAND\get@edindex@hyperref	318
COMMAND\get@edindex@ledinnote@command	314
COMMAND\get@fnmark	217
COMMAND\get@index@command	445
COMMAND\get@linelistfile	441
COMMAND\get@thisfootnote	224
COMMAND\getline@num	162, 164
COMMAND\gl@p	98
COMMAND\global	112
COMMAND\globaldefs	112
COMMAND\gls	68, 321
COMMAND\hangindentX	46, 449, 452
COMMAND\hangingsymbol	387, 442
COMMAND\hbox	190
COMMAND\hfill	444
COMMAND\hidenumbering	23, 121, 449
COMMAND\hidenumberingonleftpage	23, 121, 455
COMMAND\hidenumberingonrightpage	23, 121, 455
COMMAND\hline	68
COMMAND\hrulefill	357
COMMAND\hspace	39, 186, 189, 191, 197, 200, 236, 441, 447
COMMAND\hspaceX	390, 453, 454
COMMAND\hsizethreecol	388
COMMAND\hsizethreecolX	48, 50
COMMAND\hsizetwocol	388
COMMAND\hsizetwocolX	48, 50
COMMAND\hyperlinkR	318
COMMAND\hyperlinkformat	318
COMMAND\hyperlinkformatR	318
COMMAND@if@RTL	82, 138
COMMAND@if@edtext@	447, 450
COMMAND@if@eled@sectioning	371

COMMAND\if@firstlineofpage	83
COMMAND\if@firstlineofpageR	83
COMMAND\if@msd@options@fullpage	339
COMMAND\if@msdata@insertedfrompreviouspage	337
COMMAND\if@nobreak	154
COMMAND\if@noneed@Footnote	128
COMMAND\ifXnote@	91
COMMAND\ifafterendnumberingR	91
COMMAND\ifautopar	156
COMMAND\ifby@autopar	156
COMMAND\ifbypage@	98
COMMAND\ifbypage@R	98
COMMAND\ifbypstart@	98
COMMAND\ifbypstart@R	98
COMMAND\iffirst@linenum@out@	125, 126
COMMAND\ifndtl@innote	91
COMMAND\ifndtl@notenumber	91
COMMAND\ifinserthangingsymbol	324
COMMAND\ifinstanza	323
COMMAND\ifstwofollowinglines	211
COMMAND\ifl@d@Xmorethantwolines	207, 448
COMMAND\ifl@d@Xtwolines	207
COMMAND\ifl@d@dash	207
COMMAND\ifl@d@elin	207
COMMAND\ifl@d@esl	207
COMMAND\ifl@d@pnum	207
COMMAND\ifl@d@ssub	207
COMMAND\ifl@dend@X	261
COMMAND\ifl@dmemoir	440
COMMAND\ifl@dpaging	447
COMMAND\ifl@dpairing	90, 442
COMMAND\ifl@dprintingpages	448
COMMAND\ifl@dskipnumber	168
COMMAND\ifl@dstartendok	356
COMMAND\ifl@imakeidx	82
COMMAND\ifledRcol	90, 442
COMMAND\ifledRcol@	90, 446
COMMAND\iflemmacommand@	447
COMMAND\ifnoend@	246
COMMAND\ifnoledgroup@	312
COMMAND\ifnoteschanged@	108
COMMAND\ifnumberedpar@	150
COMMAND\ifnumbering	92, 95
COMMAND\ifnumberingR	91, 442
COMMAND\ifnumberline	140, 168
COMMAND\ifpst@rted	442
COMMAND\ifpst@rtedL	92
COMMAND\ifresumenummering@start	95
COMMAND\ifseriesbefore	265
COMMAND\ifstopmsdata@inserted@	332

COMMAND\ifsublines@	105, 118
COMMAND\iftrue	450
COMMAND\ifvmode	285
COMMAND\ifxxx	365
COMMAND\ignorespaces	138
COMMAND\imki@wrindexentry	82
COMMAND\immediate	125, 126, 240
COMMAND\indent	18, 156, 389
COMMAND\index	320, 321, 456
COMMAND\indtl@wrindexentry	82
COMMAND\initnumbering@quote	363, 451
COMMAND\initnumbering@reg	441
COMMAND\initnumbering@sectcmd	451
COMMAND\inplaceoflemmaseparator	388
COMMAND\inplaceofnumber	388
COMMAND\insert	140, 173, 181, 184, 198, 214, 220
COMMAND\insert@Xtxtbeforenotes	178, 198
COMMAND\insert@count	122, 128, 137
COMMAND\insert@countR	137
COMMAND\insert@msdata	333, 339
COMMAND\insert@txtbeforenotesX	178
COMMAND\inserthangingsymbol	444
COMMAND\insertlines@list	106, 122
COMMAND\insertparafootsepX	234
COMMAND\inserts@list	133, 151, 173, 174, 190
COMMAND\interAfootnotelinepenalty	442
COMMAND\interfootnotelinepenalty	442
COMMAND\interlinepenalty	181
COMMAND\interparanoteglue	386
COMMAND\justifying	47
COMMAND\l@advance@parledegroup@beforenormalnotes	451
COMMAND\l@d@@@wrindexhyp	447
COMMAND\l@d@add	142
COMMAND\l@d@end	241, 261
COMMAND\l@d@nums	137, 140, 142, 207, 208
COMMAND\l@d@section	241
COMMAND\l@d@set	117, 131
COMMAND\l@dampcount	347
COMMAND\l@dbfnote	217, 441
COMMAND\l@dcheckstartend	356
COMMAND\l@dchset@num	117
COMMAND\l@dcolcount	347, 348
COMMAND\l@dcollect@@body	341
COMMAND\l@dcollect@body	340
COMMAND\l@dcsnote	446
COMMAND\l@dcsnotetext	161, 303, 304
COMMAND\l@dcsnotetext@l	161, 303, 304
COMMAND\l@dcsnotetext@r	161, 303, 304
COMMAND\l@ddodoreinextrafeet	276, 440
COMMAND\l@ddoxtrafeet	276, 440

COMMAND\l@demptyd@ta	441
COMMAND\l@dend@close	240
COMMAND\l@dend@open	240
COMMAND\l@dend@stuff	241
COMMAND\l@denbody	341
COMMAND\l@dfeetbeginmini	440
COMMAND\l@dfeetendmini	440
COMMAND\l@dgetline@margin	442
COMMAND\l@dgetlock@disp	442
COMMAND\l@dgetref@num	291
COMMAND\l@dgetsidenote@margin	300, 442
COMMAND\l@dgobbeloptarg	447
COMMAND\l@dgobblearg	447
COMMAND\l@dgobbleoptarg	346
COMMAND\l@dlabel@parse	291
COMMAND\l@dld@ta	168, 170
COMMAND\l@dlp@rbox	305
COMMAND\l@dlsn@te	442
COMMAND\l@dlsnote	446
COMMAND\l@dmake@labels	285, 286, 299
COMMAND\l@dmake@labelsR	299
COMMAND\l@dnumpsstartL	92, 442
COMMAND\l@dp@rsefootspec	208
COMMAND\l@dp@rsefootspec	208
COMMAND\l@dpush@begins	341
COMMAND\l@drd@ta	168, 170
COMMAND\l@dref@undefined	291
COMMAND\l@drsn@te	442
COMMAND\l@drsnote	446
COMMAND\l@dtabaddcols	356
COMMAND\l@dtabnoexpands	440
COMMAND\l@dumboxmpfoot	451
COMMAND\l@dunboxmpfoot	441
COMMAND\l@dunhbox@line	157
COMMAND\l@dzeropenalties	442, 447
COMMAND\l@pb	377
COMMAND\l@prev@nopb	376
COMMAND\l@prev@pb	376
COMMAND\l@reg	384
COMMAND\l@wrap@ifnotemptybox	89
COMMAND\label	19, 61, 64, 67, 283, 284, 292
COMMAND\label@refs	283
COMMAND\labelstarttrue	19, 391, 443, 458
COMMAND\labelref@list	283, 286
COMMAND\language	190
COMMAND\lastbox	156
COMMAND\lastskip	130
COMMAND\latex@makecol	278
COMMAND\leavevmode	18, 156
COMMAND\led@check@nopb	377

COMMAND\led@check@pb	377
COMMAND\led@nopb	376–378
COMMAND\led@nopbnum	376, 377
COMMAND\led@pb	376–378
COMMAND\led@pb@macro	377
COMMAND\led@pbnum	376, 377
COMMAND\led@reinit@index@fornote	321
COMMAND\led@set@index@fornote	320
COMMAND\ledRflag	318
COMMAND\ledchapter	386, 443
COMMAND\ledfootinsdim	387
COMMAND\ledinnernote	64, 301, 446, 455
COMMAND\ledinnote	315, 385, 450
COMMAND\ledinnotemark	66, 385, 449
COMMAND\ledleftnote	64, 301
COMMAND\ledlinenum	103, 442
COMMAND\ledllfill	161
COMMAND\ledsnotefontsetup	455
COMMAND\ledsnotesep	65
COMMAND\ledsnotewidth	64
COMMAND\lednopb	73, 376
COMMAND\lednopbinverse	378
COMMAND\lednopbinversetrue	54, 73
COMMAND\lednopbnum	376
COMMAND\ledouternote	64, 301, 446, 455
COMMAND\ledpb	73, 376, 386
COMMAND\ledpbnum	376
COMMAND\ledpbsetting	73, 377, 452
COMMAND\ledrightnote	64, 301
COMMAND\ledrsnotefontsetup	455
COMMAND\ledrsnotesep	65
COMMAND\ledrsnotewidth	64
COMMAND\ledsection	386
COMMAND\ledsectnomark	364
COMMAND\ledsectnotoc	364
COMMAND\ledsetnormalparstuff	385, 386, 449
COMMAND\ledsetnormalparstuff@common	235
COMMAND\ledsetnormalparstuffX	385, 386, 449
COMMAND\ledsidenote	64, 301, 303, 304
COMMAND\ledsubsection	386
COMMAND\ledsubsubsection	386
COMMAND\ledxxx	446
COMMAND\left	70
COMMAND\leftctab	357
COMMAND\leftheadline	103
COMMAND\leftlinenum	22, 103, 440, 442
COMMAND\leftltab	357
COMMAND\leftnoteupfalse	65
COMMAND\leftpstartnum	172
COMMAND\leftftab	357

COMMAND\leftsidenote	304
COMMAND\leftskip	186, 189, 191
COMMAND\lemma	3, 25, 27, 28, 30–32, 133, 137, 138, 141, 144, 382, 442, 443, 450, 451, 453
COMMAND\lemmaseparator	388
COMMAND\let	28, 51, 135, 140, 261, 278, 328, 440
COMMAND\letboxXendendlinenum	388
COMMAND\line	195, 199
COMMAND\line@list	106, 108, 123, 140
COMMAND\line@list@stuff	92, 109, 110, 125, 440, 442
COMMAND\line@list@version	112
COMMAND\line@margin	99, 170, 300
COMMAND\line@num	105, 107, 168, 440
COMMAND\line@set	142
COMMAND\lineannot	57, 60, 142
COMMAND\lineannot@set	142
COMMAND\lineation	21, 98, 458, 459
COMMAND\linebreak	39
COMMAND\lineneation	454
COMMAND\linenum	25, 27, 28, 57, 60, 62, 133, 142, 289, 292, 299, 382
COMMAND\linenum@out	125, 283, 286
COMMAND\linenumannotation	56, 166
COMMAND\linenumberannotation	458
COMMAND\linenumberlist	21, 90, 169, 440
COMMAND\linenumberstyle	23, 103, 440, 457
COMMAND\linenumincrement	21, 442
COMMAND\linenummargin	21, 99, 300
COMMAND\linenumr@p	103, 440, 442
COMMAND\linenumrep	103, 442
COMMAND\linenumsep	22, 65, 103, 301
COMMAND\linerangesep@	274
COMMAND\lineref	283, 289, 298, 388, 448
COMMAND\list@clear	97
COMMAND\list@clearing@reg	442
COMMAND\list@create	97
COMMAND\lock@disp	102
COMMAND\lock@off	120
COMMAND\lock@on	119
COMMAND\lockdisp	22, 102
COMMAND\loop	177, 324
COMMAND\ltab	357, 358, 362
COMMAND\ltabtext	362
COMMAND\m@mmf@prepare	216
COMMAND\makeatletter	160
COMMAND\makehboxoffhboxes	192, 193
COMMAND\makeindex	66, 317
COMMAND\makelabel	298
COMMAND\managestanza@modulo	325
COMMAND\marginpar	64, 75, 299, 300, 441
COMMAND\marginparwidth	64, 301
COMMAND\markboth	160

COMMAND\mathchardef	325
COMMAND\maxhXnotes	388
COMMAND\maxhnotesX	50, 76, 387, 445, 446, 448, 449
COMMAND\maxlinesinpar@list	110
COMMAND\measurebody	359
COMMAND\measuretbody	359
COMMAND\memorybreak	20
COMMAND\metaXXXvfootnote	214
COMMAND\microtype	157
COMMAND\morenoexpands	76, 133, 135
COMMAND\morethantwolines	388
COMMAND\mpfnpos	37, 237, 445, 456
COMMAND\mpnormalfootgroup	440
COMMAND\mpnormalvfootnote	441
COMMAND\msdata	34, 35, 331–333, 457
COMMAND\msdataposition	35
COMMAND\multfootsep	36, 216
COMMAND\multiplefootnotemarker	216
COMMAND\musixtex	445
COMMAND\n@num	442, 449
COMMAND\n@num@ref	449
COMMAND\new@line	127, 441
COMMAND\new@series	135
COMMAND\newcommand	28, 78, 94, 216, 286
COMMAND\newcommandx	28, 29
COMMAND\newhookarg@specific	274
COMMAND\newhookcommand@series	268, 269, 449
COMMAND\newhookcommand@series@reload	269
COMMAND\newhookcommand@toggle@reload	269, 447
COMMAND\newhooktoggle@series	268, 269, 449
COMMAND\newhooktoggle@specific	274
COMMAND\newif	449
COMMAND\newline	39
COMMAND\newlinechar	262
COMMAND\newseries	36, 383, 386, 387
COMMAND\newseries@	253, 266
COMMAND\newverse	54, 58, 386, 446, 457
COMMAND\next	324
COMMAND\next@action	110
COMMAND\next@actionline	110
COMMAND\next@insert	174
COMMAND\next@line@list@stuff	126, 458
COMMAND\nl@regR	112
COMMAND\no@expands	76, 135, 141, 261, 440
COMMAND\noXlemmaseparator	388
COMMAND\nobreak	207
COMMAND\nocritical	253
COMMAND\noeledsec	72, 386
COMMAND\noendnotes	386
COMMAND\noexpand	384

COMMAND\nofamiliar	271
COMMAND\noindent	18, 19, 54, 55, 156, 452
COMMAND\noindentX	452
COMMAND\nomk@	274
COMMAND\nonbreakableafternumber	388
COMMAND\nonnumberinfootnote	388
COMMAND\norelax	52
COMMAND\normal@footnotemarkX	220
COMMAND\normal@page@break	376
COMMAND\normal@pars	235
COMMAND\normalbfnoteX	442
COMMAND\normalbodyfootmarkX	220
COMMAND\normalfont	456
COMMAND\normalfootfmt	51, 186, 193, 202, 241, 455
COMMAND\normalfootfmtX	221
COMMAND\normalfootfootmarkX	221
COMMAND\normalfootgroup	187
COMMAND\normalfootgroupX	222
COMMAND\normalfootnoterule	182
COMMAND\normalfootstart	186, 189
COMMAND\normalfootstartX	221
COMMAND\normalvfootnote	184, 185
COMMAND\normalvfootnote@inserted	184, 185
COMMAND\normalvfootnoteX	220
COMMAND\notbool	365
COMMAND\notefontsetup	387
COMMAND\notefontsizeX	46, 387
COMMAND\notenumfont	387
COMMAND\notenumfontX	45, 387
COMMAND\notesXwidthliketwocolumns	388
COMMAND\noteswidthliketwocolumnsX	50, 388, 447, 449
COMMAND\num@lines	150, 176
COMMAND\numberlinefalse	20
COMMAND\numberlinetrue	20
COMMAND\numberonlyfirstinline	266, 388
COMMAND\numberonlyfirstintwolines	388
COMMAND\numberpstartfalse	19
COMMAND\numberpstarttrue	19, 41, 385, 442, 451, 457, 458
COMMAND\numberstanza	42
COMMAND\numberstanzafalse	55
COMMAND\numberstanzatrue	55
COMMAND\numlabfont	22, 51, 103
COMMAND\none@line	150
COMMAND\nonehalfspacing	452
COMMAND\nonlyXpstart	388
COMMAND\nonlysideX	260
COMMAND\npage@action	107, 117
COMMAND\npage@start	107, 442
COMMAND\npagecontents	107
COMMAND\npagenumbering	281

COMMAND\pageparbreak	386
COMMAND\pageref	61, 289
COMMAND\par	27, 39, 155, 156, 235
COMMAND\par@line	150, 176
COMMAND\par@patch@pagenumbering	281
COMMAND\par@patch@thepage	127, 281
COMMAND\para@footgroup	189
COMMAND\para@footgroupX	233
COMMAND\para@footsetup	189, 440
COMMAND\para@footsetupX	230, 440, 447
COMMAND\para@vfootnoteX	231
COMMAND\parafootfmt	192, 193, 455
COMMAND\parafootfmtX	232
COMMAND\parafootftm	194
COMMAND\parafootftmX	234
COMMAND\parafootftmsep	387
COMMAND\parafootsep	388, 445, 450
COMMAND\parafootsepX	48, 109, 387, 388, 455
COMMAND\parafootstart	189
COMMAND\parafootstartX	231
COMMAND\paravfootnote	190, 193
COMMAND\parfillskip	192
COMMAND\parindent	452
COMMAND\parindentX	46, 390, 452, 454
COMMAND\parse@annot	167
COMMAND\parshape	75
COMMAND\parskip	19, 54, 55, 156
COMMAND\pause@numbering	95
COMMAND\pausenumbering	20, 95, 110, 111, 157, 445, 447, 454
COMMAND\pausenumbering@page@num	95
COMMAND\penalty	192
COMMAND\pend	2, 7, 17–21, 23, 72, 73, 131, 133, 136, 143, 150, 151, 153–157, 172, 173, 384, 445, 446, 455–457
COMMAND\preXnotes	449, 455
COMMAND\prenotesX	49, 239, 449
COMMAND\prepare@Xgroupbyline	214
COMMAND\prepare@Xprenotes	238
COMMAND\prev@nopb	377
COMMAND\prev@pb	377
COMMAND\prevlineX	109
COMMAND\prevpageX@num	109
COMMAND\print@Xfootnoterule	449
COMMAND\print@Xnotes	276, 277
COMMAND\print@Xnotes@forpages	448
COMMAND\print@eledsection	159
COMMAND\print@footnoteXrule	449
COMMAND\print@leftmargin@eledsection	365
COMMAND\print@lemma	203
COMMAND\print@line	158
COMMAND\print@notesX@forpages	448

COMMAND\print@rightmargin@eledsection	365
COMMAND\printendlines	246, 252, 296, 390, 440, 442
COMMAND\printlinefootnote	204, 206, 448
COMMAND\printlinefootnotearea	206, 207, 448
COMMAND\printlinefootnotenumbers	204
COMMAND\printlines	108, 186, 202, 207, 208, 246, 296, 390, 440, 442, 448, 453
COMMAND\printnpnum	385, 390
COMMAND\printpstart	203
COMMAND\protect	135, 384
COMMAND\providecommand	215, 440
COMMAND\pstarL	151
COMMAND\pstart	2, 7, 17–21, 23, 71–73, 117, 131, 136, 143, 150, 151, 153–156, 159, 173, 370, 384, 442–444, 446, 447, 449–451, 454–458
COMMAND\pstartinfootnote	388
COMMAND\pstartinfootnoteeverytime	388
COMMAND\pstartnum	172
COMMAND\pstartref	59, 283, 290, 445, 457
COMMAND\pstarts	442, 443
COMMAND\pstarts@readL	150
COMMAND\pstarts@typeset@L	150
COMMAND\raggedX	48
COMMAND\raggedleft	47
COMMAND\raggedright	47
COMMAND\raw@text	150, 151
COMMAND\rbracket	44
COMMAND\read@linelist	109, 111
COMMAND\ref	61, 64, 67
COMMAND\refformated@	296
COMMAND\refformatedwithpage	296
COMMAND\relax	19, 117, 140, 163, 174, 328, 346, 384
COMMAND\renewcommand	75, 387, 390
COMMAND\reset@current@annot	167
COMMAND\reset@msd@options@iffullpage	339
COMMAND\resetprevline@	109
COMMAND\resetprevpage@	109
COMMAND\resumenumbering	20, 91, 95, 96, 110, 111, 157, 442, 446, 447, 454
COMMAND\right	70
COMMAND\rightctab	357
COMMAND\rightlinenum	22, 103, 440, 442
COMMAND\rightltab	358
COMMAND\rightnoteupfalse	65
COMMAND\rightrtab	358
COMMAND\rightsidenote	303
COMMAND\rightrightskip	186, 189, 191, 192
COMMAND\rightstartnum	172
COMMAND\rigidbalance	195, 197, 199, 390, 453
COMMAND\rigidbalanceX	195, 390, 453
COMMAND\robustify	39
COMMAND\roman	351, 453
COMMAND\rtab	357, 358, 362

COMMAND\rtabtext	359, 362
COMMAND\sameparallelpagenunder	127
COMMAND\sameword	16, 29–32, 124, 143–146, 148, 448, 450, 452, 455, 458
COMMAND\sameword@inedtext	144
COMMAND\saweword	144
COMMAND\scriptsize	103
COMMAND\section	71, 442, 456
COMMAND\section@num	91
COMMAND\sectionmark	364
COMMAND\select@lemmafont	51, 179
COMMAND\series	253
COMMAND\series@	253
COMMAND\seriesatbegin	37, 265, 449
COMMAND\seriesatend	37, 265, 449
COMMAND\set@Xtxtbeforenotes	177
COMMAND\set@continuousnumberingforl	96
COMMAND\set@line	140
COMMAND\set@line@action	107, 118
COMMAND\set@txtbeforenotesX	177
COMMAND\setSErefonlypageprefixmore	62, 296, 454, 457
COMMAND\setSErefonlypageprefixsingle	62, 296, 454, 457
COMMAND\setSErefprefixmore	62
COMMAND\setSErefprefixsingle	62
COMMAND\setapprefprefixmore	62, 387
COMMAND\setapprefprefixsingle	62, 387, 453
COMMAND\setcommand@series	267
COMMAND\sethangingsymbol	53, 323, 387, 452
COMMAND\sethangingsymbol	52
COMMAND\setistwofollowinglines	211
COMMAND\setl@dlprbox	305
COMMAND\setline	23, 108, 112, 117, 131, 135, 153, 451
COMMAND\setlinenum	23, 112, 117, 131, 440, 458
COMMAND\setmsdatalabel	35
COMMAND\setmsdataposition	457
COMMAND\setmsdataseries	35
COMMAND\setprintendlines	246, 249, 442
COMMAND\setprintlines	208, 211, 246, 442
COMMAND\setsidenoteseq	65
COMMAND\setsidenoteseq	387
COMMAND\setstanzaindent	326
COMMAND\setstanzaindents	53, 325, 384
COMMAND\setstanzapenalties	325
COMMAND\setstanzavalues	325
COMMAND\settoggle@series	266, 443, 447
COMMAND\showlemma	134, 440, 441
COMMAND\showwordrank	32, 145
COMMAND\sidenote@margin	441
COMMAND\sidenotemargin	64, 441, 446
COMMAND\sidenoteseq	387
COMMAND\sidepstartnumtrue	19

COMMAND\skip	186
COMMAND\skipnumbering	23, 120, 132, 442, 449, 450
COMMAND\skipnumbering@reg	449
COMMAND\small	45, 46
COMMAND\somemacro	33
COMMAND\special	13
COMMAND\splitmaxdepth	182, 197
COMMAND\splitoff	195
COMMAND\splittopskip	182, 197, 199
COMMAND\stanza	22, 23, 54, 55, 58, 328, 387, 389, 452, 457
COMMAND\stanza@hang	328
COMMAND\stanza@line	328
COMMAND\stanzaindent	53, 326, 448
COMMAND\stanzaindent*	53
COMMAND\stanzaindentbase	324, 325
COMMAND\stanzanumwrapper	55
COMMAND\startlock	22, 107, 132, 328
COMMAND\startstanzahook	387
COMMAND\startsub	22, 107, 130
COMMAND\stopmsd	334
COMMAND\stopmsdata	34, 332
COMMAND\store@annot@to@absline	167
COMMAND\strip@pt	189
COMMAND\strutbox	197
COMMAND\sub@action	107, 118
COMMAND\sub@lock	106
COMMAND\sub@off	115, 286
COMMAND\sub@on	115, 286
COMMAND\subline@num	105, 107
COMMAND\sublinenum@rep	440
COMMAND\sublinenumberstyle	23, 103, 440
COMMAND\sublinenumincrement	21
COMMAND\sublinenumr@p	103, 440, 442
COMMAND\sublinenumrep	103, 442
COMMAND\sublineref	59, 283, 289
COMMAND\subsectionmark	364
COMMAND\sw@inthisedtext	137
COMMAND\sw@list@inedtext	141, 149
COMMAND\sw@txt	145
COMMAND\swnoexpands	32, 458
COMMAND\symlinenum	388
COMMAND\symplinenumber	387
COMMAND\sza@penalty	328
COMMAND>tag	448
COMMAND\text	382
COMMAND\text<language>	46
COMMAND\textbeforenotesX	177
COMMAND\textcolor	76
COMMAND\textheight	76
COMMAND\textsuperscript	57, 89, 167

COMMAND\the	127, 440
COMMAND\thefootnoteA	35
COMMAND\thefootnoteX	444
COMMAND\thelabidx	319
COMMAND\thepage	112, 127, 281
COMMAND\thepstart	19, 391
COMMAND\thepstartL	443
COMMAND\thepstartR	443
COMMAND\thestanza	55
COMMAND\this@line@list@version	125
COMMAND\this@section@next@page@num	95
COMMAND\this@section@page@num	108
COMMAND\thisfootnote	224
COMMAND\threecol@begin@insert	198
COMMAND\threecolfootfmt	198, 455
COMMAND\threecolfootfmtX	229
COMMAND\threecolfootgroup	197
COMMAND\threecolfootgroupX	229
COMMAND\threecolfootsetup	196
COMMAND\threecolfootsetupX	228
COMMAND\threecolvfootnote	197
COMMAND\threecolvfootnote@inserted	197
COMMAND\threecolvfootnoteX	228
COMMAND\toendnotes	26, 244, 455
COMMAND\twocolfootfmt	455
COMMAND\twocolfootfmtX	226
COMMAND\twocolfootgroupX	227
COMMAND\twocolfootsetupX	226
COMMAND\twocolvfootnoteX	226
COMMAND\twolines	266, 388
COMMAND\twolines@A	266
COMMAND\twolines@B	266
COMMAND\twolines@C	266
COMMAND\twolinesbutnotmore	388
COMMAND\twolinesonlyinsamepage	388
COMMAND\txtbeforXnotes	388
COMMAND\txtbeforenonesX	49, 458
COMMAND\txtbeforenonesonlyonceX	49, 458
COMMAND\uline	36
COMMAND\unhbox	157, 190
COMMAND\unpenalty	192, 193
COMMAND\unskip	192
COMMAND\unvxh	192, 389
COMMAND\unvxhX	389
COMMAND\upbracefill	357
COMMAND\usingcritext	383, 386
COMMAND\usingedtext	383, 386
COMMAND\vAfootnote	181
COMMAND\vadjust	129
COMMAND\variant	28

COMMAND\ vbox	153, 155, 190, 191, 195, 237
COMMAND\ vfootnote	181, 186, 190, 197
COMMAND\ vl@dbfnote	217, 441
COMMAND\ vl@disnote	304
COMMAND\ vl@dlsnote	304
COMMAND\ vl@dosnote	304
COMMAND\ vl@drsnote	304
COMMAND\ vnumfootnoteX	442
COMMAND\ vsize	50, 76
COMMAND\ vsplit	175
COMMAND\ wklam	357
COMMAND\ wklamec	357
COMMAND\ wapunktel	357
COMMAND\ wastricht	357
COMMAND\ widthX	50, 390, 454
COMMAND\ wrap@edcrossref	289, 447
COMMAND\ wrapcontentX	46, 455
COMMAND\ wrapped@bodyfootmarkX	234
COMMAND\ wrapped@footfootmarkX	234
COMMAND\ x...	60
COMMAND\ xabslineref	290
COMMAND\ xannotationref	60
COMMAND\ xdef	97, 328
COMMAND\ xflagref	60, 290, 390, 453
COMMAND\ xleft@appenditem	97, 134
COMMAND\ xlineref	60, 390, 453
COMMAND\ xpageref	60
COMMAND\ xparse	33
COMMAND\ xpstartref	60, 445, 458
COMMAND\ xr	64
COMMAND\ xright@appenditem	97
COMMAND\ xsublineref	60
COMMAND\ xxref	60, 292, 299, 445, 448, 455
COMMAND\ xxxmatter	281
ENVIRONMENTastanza	453
ENVIRONMENTedarrayc	362
ENVIRONMENTedarrayl	362
ENVIRONMENTedarrayr	362
ENVIRONMENTedtabularc	362
ENVIRONMENTedtabularl	362
ENVIRONMENTedtabularr	362
ENVIRONMENTledgroup	82, 311, 390, 453
ENVIRONMENTledgroupsized	311
PACKAGE(r)(e)ledmac	37
PACKAGEEledmac	78, 107, 316, 385, 386, 448, 449
PACKAGEEledpar	449
PACKAGEEtoolbox	81
PACKAGEParallel	392
PACKAGEReledmac	389, 390
PACKAGEamsgen	340

PACKAGEamsmath	340
PACKAGEbabel	46, 77, 351, 453
PACKAGEbiblatex	74
PACKAGEbidi	46, 47, 82, 452
PACKAGEcaption	90
PACKAGEcolor	76
PACKAGEcsquotes	457
PACKAGEedmac	1, 6, 11–14, 78, 207, 215, 283, 325, 382, 392, 440
PACKAGEedstanza	1, 14, 323
PACKAGEeledmac	1, 10, 11, 14, 16, 67, 215, 312, 316, 343, 366, 379, 383, 385–387, 444, 446, 448
PACKAGEeledpar	182, 392, 442, 446–448
PACKAGEetex	452
PACKAGEetoolbox	97, 143, 253, 266, 275, 303, 365, 376
PACKAGEfancyhdr	278, 456
PACKAGEfloatrow	74, 390
PACKAGEfootmisc	36, 77, 82, 215, 392, 456
PACKAGEgeometry	15
PACKAGEglossaries	67, 68, 321, 453
PACKAGEhandout	447
PACKAGEhyperlink	260
PACKAGEhyperref	60, 136, 234, 284, 318, 319, 368, 378, 445–447, 454, 457
PACKAGEifluatex	81
PACKAGEifxetex	81
PACKAGEimakeidx	65, 66, 75, 82, 312, 316, 386, 444–446, 448
PACKAGEindextools	320
PACKAGEindextool	386
PACKAGEindextools	65, 66, 75, 82, 91, 312, 316, 320, 386, 448, 453, 456
PACKAGEinputenc	145
PACKAGEledarab	77
PACKAGEledmac	1, 11, 14, 77, 97, 316, 382, 383, 386, 389
PACKAGEledpar	77
PACKAGEMemoir	82, 316, 386, 392, 446
PACKAGEMorewrites	75
PACKAGEMusixtex	446
PACKAGEperpage	453
PACKAGEpolyglossia	44, 77, 180, 202, 453
PACKAGERagged2e	47, 81
PACKAGEReledmac	1, 2, 6, 11, 12, 14–17, 20, 21, 24, 25, 28–30, 33, 36, 39, 40, 43, 46, 47, 49, 51, 53, 55, 58, 60–62, 64–68, 72–79, 98, 100, 106, 108, 111, 112, 115, 125–127, 135, 138, 140, 142, 143, 173, 183, 186, 191, 202, 215, 241, 253, 258, 266, 275, 281, 289, 290, 293, 299, 316, 337, 343, 364, 365, 378, 386, 387, 389–391, 451, 454, 456
PACKAGEReledpar	1, 4, 6, 8, 16, 19, 36, 50, 60, 62, 63, 73, 74, 77, 79, 90, 98, 110, 115, 126, 127, 137, 140, 150, 151, 183, 187, 236, 237, 253, 259, 274, 276, 277, 281, 312, 323, 364, 452, 453, 455, 456, 458, 459
PACKAGESuffix	81
PACKAGETabmac	1, 14, 392
PACKAGEulem	36
PACKAGEuninormalize	29
PACKAGExargs	29, 81
PACKAGExkeyval	79, 274

PACKAGEparse	81, 458
PACKAGEX	5, 64, 299, 454
PACKAGEXref	299
PACKAGEXstring	81, 318

A

\absline@num	1
Abu Kamil Shuja' b. Aslam	13
\actionlines@list	1
\actions@list	1
\add@inserts	1
\add@inserts@next	1
\add@msd@	1
\add@msdata	1
\add@msdata@firstlineofpage	1
\add@penalties	1
\add@Xgroupbyline	1
\addtol@denbody	1
Adelard II	13
\advancelabel@refs	1
\advanceline	1, 22
\Aendnote	25
\affixline@num	1
\affixpstart@num	1
\affixside@note	1
\Afootnote	25
\afternoteX	48
\afterruleX	49
\ampersand	1, 58
\annot@list	1
\annotationref	1
\annotref	59
\append@notesX	1
\append@Xnotes	1
\applabel	1, 62
\appref	1, 62
\apprefwithpage	1, 62
\arrangementX	1, 38
\arrangementX@normal	1
\arrangementX@threecol	1
\arrangementX@twocol	1
\at@every@pend	1
\AtEndEveryPend	1, 19
\AtEveryPend	1, 19
\AtEveryPstart	1, 19
\AtEveryStanza	1, 54
\AtEveryStopStanza	1, 55
\AtStartEveryPstart	1, 19
\AtStartEveryStanza	1, 54, 55
\autopar	1, 18

B

\ballast	75
\ballast@count	1
Beeton, Barbara Ann Neuhaus Friend	18
\beforeedchapter	1
\BeforeEveryStopStanza	1
\beforeinsertingX	47
\beforeinsertion@X	1
\beforenotesX	49
\beginnumbering	1, 17
\Bendnote	25
\Bfootnote	25
\bhookgroupX	49
\bhooknoteX	47
\bodyfootmarkA	35
\boxfootnotenumbers	1
Bredon, Simon	13
Breger, Herbert	13, 14, 343
Brey, Gerhard	13
Busard, Hubert L. L.	13
\bypage@false	1
\bypage@true	1
\bypstart@false	1
\bypstart@true	1

C

\c@addcolcount	1
\c@ballast	1
\c@firstlinenum	1
\c@firstsublinenum	1
\c@labidx	1
\c@linenumincrement	1
\c@sublinenumincrement	1
\Cendnote	25
\Cfootnote	25
\ch@ck@l@ck	1
\ch@cksub@l@ck	1
\chapter	1
\check@pb@in@verse	1
Chester, Robert of	13
Claassens, Geert H. M.	13
\colalignX	47
\continuousnumberingwithcolumns@sync@linenumber@singletext	1
Copernicus, Nicolaus	13
\critext	382
\ctab	1
\ctabtext	1
\current@annot	1

D

Dekker, Dirk-Jan	76
\Dendnote	25
\Dfootnote	25
\disable@endnotes	1
\disable@familiarnotes	1
\disable@notes	1
\disable@sidenotes	1
\disablel@dtabfeet	1
\do@actions	1
\do@actions@fixedcode	1
\do@actions@next	1
\do@ballast	1
\do@feet@custom@order	1
\do@feetX	1
\do@insidelinehook	1
\do@line	1
\do@linehook	1
\do@lockoff	1
\do@lockoffL	1
\do@lockon	1
\do@lockonL	1
\do@Xfeet	1
\doedindexlabel	1
\doendnotes	1, 26
\doendnotesbysection	1, 26
\doinsidelinehook	1, 24
\dolinehook	1, 24
\dosplits	1
Downes, Michael	75, 190, 192
\doextrafeet	1
\dummy@edindex	1
\dummy@edtext	1
\dummy@edtext@showlemma	1
\dummy@ref	1

E

\edaftertab	1, 70, 357
edarrayc (environment)	68
edarrayl (environment)	68
edarrayr (environment)	68
\edatleft	1, 70
\edatright	1, 70
\edbeforetab	1, 70, 357
\edfilldimen	1
\edfont@info	1
\edindex	1, 65
\edindexlab	1, 67
\EDLABEL	1
\edlabel	1, 59

<code>\edlabelE</code>	1, 61
<code>\edlabelS</code>	1, 61
<code>\edlabelSE</code>	1, 61
<code>\edlineref</code>	1, 59
<code>\edmakelabel</code>	1, 61
<code>\edpageref</code>	1, 59
<code>\edrowfill</code>	1, 69
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 69
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc</code> (environment)	68
<code>edtabularl</code> (environment)	68
<code>edtabularr</code> (environment)	68
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 24
<code>\edvertdots</code>	1, 71
<code>\edvertline</code>	1, 71
<code>\Eendnote</code>	25
<code>\Efootnote</code>	25
<code>\eled@chapter</code>	1
<code>\eled@section</code>	1
<code>\eled@sectioning@out</code>	1
<code>\eled@subsection</code>	1
<code>\eled@subsubsection</code>	1
<code>\eledchapter</code>	1
<code>\eledchapter*</code>	1
<code>\eledsection</code>	1
<code>\eledsection*</code>	1
<code>\eledsubsection</code>	1
<code>\eledsubsection*</code>	1
<code>\eledsubsubsection</code>	1
<code>\eledsubsubsection*</code>	1
<code>\enablel@dtabfeet</code>	1
<code>\end@lemmas</code>	1
<code>\endashchar</code>	1
<code>\endline@num</code>	1
<code>\endlock</code>	1, 22
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 17
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 22
<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	68
<code>edarrayl</code>	68

edarrayr	68
edtabularc	68
edtabularl	68
edtabularr	68
ledgroup	58
ledgroupsize	59
minipage	58
Euclid	13
\extensionchars	1, 74

F

\f@x@l@cks	1
Fairbairns, Robin	36
\first@linenum@out@false	1
\first@linenum@out@true	1
\firstlinenum	1, 21
\firstseriesX@	1
\firstsublinenum	1, 21
\firstXseries@	1
\flag@end	1
\flag@end@later	1
\flag@end@RTL	1
\flag@start	1
\flag@start@later	1
\flag@start@RTL	1
\flagstanza	1, 58
\flush@notes	1
\fnpos	1, 37
Folkerts, Menso	13
\footfootmarkA	35
\footfudgefiddle	1, 75
\footnote	1
\footnoteA	35
\footnoteB	35
\footnoteC	35
\footnoteD	35
\footnoteE	35
\footnotelang@lua	1
\footnotelang@poly	1
\footnoteoptions@	1
\footnoteXmark	36
\footnoteXtext	36
\footplitskips	1
\fullstop	1

G

Gädeke, Nora	13
\get@edindex@hyperref	1
\get@edindex@ledinnote@command	1
\get@fnmark	1

\get@fnmarkX	1
\get@index@command	1
\get@linelistfile	1
\get@sw@txt	1
\get@thisfootnote	1
\get@thisfootnoteX	1
\getline@num	1
\gl@p	1

H

\h@num	1
\hangindentX	46
\hidenumbering	1, 23
\hidenumberingonleftpage	1, 23
\hidenumberingonrightpage	1
\Hilfsbox	1
\hilfsbox	1
\hilfscount	1
\HILFSskip	1
\Hilfsskip	1
\hilfsskip	1
\hsizethreecolX	48
\hsizetwocolX	48
\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
\ifafterendnumberingR	1
\ifat@every@pend@star@	1
\ifat@every@pstart@star@	1
\ifautopar@pause	1
\ifbypage@	1
\ifbypage@R	1
\ifbypstart@	1
\ifbypstart@R	1
\ifeledmacompat@	1
\iffirst@linenum@out@	1

<code>\ifindtl@innote</code>	1
<code>\ifindtl@notenumber</code>	1
<code>\ifinserthangingsymbol</code>	1
<code>\ifistanza</code>	1
<code>\ifl@d@dash</code>	1
<code>\ifl@d@elin</code>	1
<code>\ifl@d@esl</code>	1
<code>\ifl@d@pnum</code>	1
<code>\ifl@d@ssub</code>	1
<code>\ifl@d@Xmorethantwolines</code>	1
<code>\ifl@d@Xtwolines</code>	1
<code>\ifl@dend@X</code>	1
<code>\ifl@dhiddenumber</code>	1
<code>\ifl@dmemoir</code>	1
<code>\ifl@dpaging</code>	1
<code>\ifl@dpairing</code>	1
<code>\ifl@dprintingcolumns</code>	1
<code>\ifl@dprintingpages</code>	1
<code>\ifl@dskipnumber</code>	1
<code>\ifl@dskipversenumber</code>	1
<code>\ifl@dstartendok</code>	1
<code>\ifl@footmisc</code>	1
<code>\ifl@imakeidx</code>	1
<code>\ifl@indextools</code>	1
<code>\ifledfinal</code>	1, 74
<code>\ifledgroupnotesL@</code>	1
<code>\ifledgroupnotesR@</code>	1
<code>\iflednopbinverse</code>	1
<code>\ifledRcol</code>	1
<code>\ifledRcol@</code>	1
<code>\ifnocritical@</code>	1
<code>\ifnoend@</code>	1
<code>\ifnofamiliar@</code>	1
<code>\ifnoledgroup@</code>	1
<code>\ifnoquotation@</code>	1
<code>\ifnoteschanged@</code>	1
<code>\ifnumberedpar@</code>	1
<code>\ifnumbering</code>	1
<code>\ifnumberingR</code>	1
<code>\ifnumberline</code>	1
<code>\ifnumberstanza</code>	1
<code>\ifparapparatus@</code>	1
<code>\ifparledgroup</code>	1
<code>\ifprevpgnotnumbered</code>	1
<code>\ifpst@rtedL</code>	1
<code>\ifresumenumbering@start</code>	1
<code>\ifsameparallelpagenumber</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@txtbeforenotesX</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	177
---------------------------	-----

L

<code>\l@advance@parledgroup@beforenormalnotes</code>	1
<code>\l@d@add</code>	1
<code>\l@d@nums</code>	1
<code>\l@d@section</code>	1
<code>\l@d@set</code>	1
<code>\l@d@Xend</code>	1
<code>\l@dampcount</code>	1
<code>\l@dbfnote</code>	1
<code>\l@dcheckcols</code>	1
<code>\l@dcheckstartend</code>	1
<code>\l@dchset@num</code>	1
<code>\l@dcolcount</code>	1
<code>\l@dcollect@@body</code>	1
<code>\l@dcollect@body</code>	1
<code>\l@dcolwidth</code>	1
<code>\l@dcsnote</code>	1
<code>\l@dcsnotetext</code>	1
<code>\l@dcsnotetext@l</code>	1
<code>\l@dcsnotetext@r</code>	1
<code>\l@ddodoreintrafeet</code>	1
<code>\l@dedbeginmini</code>	1
<code>\l@dedendmini</code>	1
<code>\l@emptyd@ta</code>	1
<code>\l@dend@close</code>	1
<code>\l@dend@open</code>	1
<code>\l@dend@stuff</code>	1
<code>\l@dend@Xfalse</code>	1

<code>\l@dend@Xtrue</code>	1
<code>\l@denbody</code>	1
<code>\l@desnote</code>	1
<code>\l@dfambeginmini</code>	1
<code>\l@dfamendmini</code>	1
<code>\l@dfeetbeginmini</code>	1
<code>\l@dfeetendmini</code>	1
<code>\l@dgetline@margin</code>	1
<code>\l@dgetlock@disp</code>	1
<code>\l@dgetref@num</code>	1
<code>\l@dgetsidenote@margin</code>	1
<code>\l@dobblearg</code>	1
<code>\l@disnote</code>	1
<code>\l@dlabel@parse</code>	1
<code>\l@dld@ta</code>	1
<code>\l@dlp@rbox</code>	1
<code>\l@dlsn@te</code>	1
<code>\l@dlsnote</code>	1
<code>\l@dmake@labels</code>	1
<code>\l@dmodforedtext</code>	1
<code>\l@dnullfills</code>	1
<code>\l@dnumstartsl</code>	1
<code>\l@doldold@footnotetext</code>	1
<code>\l@dp@rsefootspec</code>	1
<code>\l@dparsedendline</code>	1
<code>\l@dparsedendpage</code>	1
<code>\l@dparsedendsub</code>	1
<code>\l@dparsedstartline</code>	1
<code>\l@dparsedstartpage</code>	1
<code>\l@dparsedstartsub</code>	1
<code>\l@dparsefootspec</code>	1
<code>\l@dpush@begins</code>	1
<code>\l@drd@ta</code>	1
<code>\l@dref@undefined</code>	1
<code>\l@drestorefills</code>	1
<code>\l@drestoreforedtext</code>	1
<code>\l@drp@rbox</code>	1
<code>\l@drrsn@te</code>	1
<code>\l@drrsnote</code>	1
<code>\l@dsetmaxcolwidth</code>	1
<code>\l@dskipnumberfalse</code>	1
<code>\l@dskipnumbertrue</code>	1
<code>\l@dtabaddcols</code>	1
<code>\l@dtabnoexpands</code>	1
<code>\l@dunboxmpfoot</code>	1
<code>\l@dunhbox@line</code>	1
<code>\l@dzeroopenalties</code>	1
<code>\l@the@c@page</code>	1
<code>\l@wrap@ifnotemptybox</code>	1
<code>\l@wrapcs@ifnotemptybox</code>	1

\label	61
\labelstartfalse	1
\labelstarttrue	1, 19
\labelref@list	1
\labelrefsparseline	1
\labelrefsparsesubline	1
\last@page@num	1
Lavagnino, John	12
\led@check@nopb	1
\led@check@pb	1
\led@err@AutoparNotNumbered	1
\led@err@BadAction	1
\led@err@edtextoutsidepstart	1
\led@err@EdtextWithoutFootnote	1
\led@err@FootnoteNotInSecondArgEdtext	1
\led@err@HighEndColumn	1
\led@err@LineationInNumbered	1
\led@err@LowStartColumn	1
\led@err@ManyLeftnotes	1
\led@err@ManyRightnotes	1
\led@err@ManySidenotes	1
\led@err@NumberingNotStarted	1
\led@err@NumberingShouldHaveStarted	1
\led@err@NumberingStarted	1
\led@err@NumberingWithoutPstart	1
\led@err@PendNoPstart	1
\led@err@PendNotNumbered	1
\led@err@PstartInPstart	1
\led@err@PstartNotNumbered	1
\led@err@ReverseColumns	1
\led@err@toendnotes@outsidenumbering	1
\led@err@TooManyColumns	1
\led@err@UnequalColumns	1
\led@error@fail@patch@@doclearpage	1
\led@error@fail@patch@@iiiminipage	1
\led@error@fail@patch@@makecol	1
\led@error@fail@patch@@reinserts	1
\led@error@fail@patch@endminipage	1
\led@error@PackageAfterEledmac	1
\led@mess@NotesChanged	1
\led@mess@SectionContinued	1
\led@nopb	1
\led@nopbnum	1
\led@pb	1
\led@pb@setting	1
\led@pbnum	1
\led@reinit@index@fornote	1
\led@set@index@fornote	1
\led@toksa	1
\led@toksb	1

<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@NoFile</code>	1
<code>\led@warn@NoIndexFile</code>	1
<code>\led@warn@NoMarginpars</code>	1
<code>\led@warn@RefUndefined</code>	1
<code>\led@warn@SeriesStillExist</code>	1
<code>\led@warning@hsizeX@deprecated</code>	1
<code>\led@warning@msdatawithoutstop</code>	1
<code>\led@warning@preXnotes@deprecated</code>	1
<code>\led@warning@Xhsize@deprecated</code>	1
<code>ledgroup (environment)</code>	58
<code>ledgroupsize (environment)</code>	59
<code>\ledinnernote</code>	1, 64
<code>\ledinnote</code>	1
<code>\ledinnotehyperpage</code>	1
<code>\ledinnotemark</code>	1
<code>\ledleftnote</code>	1, 64
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 65
<code>\ledlsnotesep</code>	1, 65
<code>\ledlsnotewidth</code>	1, 64
<code>\lednopb</code>	1, 73
<code>\lednopbinversetrue</code>	73
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	1, 64
<code>\ledpb</code>	1, 73
<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 73
<code>\ledrightnote</code>	1, 64
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 65
<code>\ledrsnotesep</code>	1, 65
<code>\ledrsnotewidth</code>	1, 64
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1

<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 64
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 22
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	65
<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
Leibniz	13
<code>\lemma</code>	1, 27
<code>\letsforverteilen</code>	1
<code>\line@list</code>	1
<code>\line@list@stuff</code>	1
<code>\line@list@version</code>	1
<code>\line@margin</code>	1
<code>\line@num</code>	1
<code>\line@set</code>	1
<code>\lineannot</code>	1, 57
<code>\lineannot@set</code>	1
<code>\lineation</code>	1, 21
<code>\linenum</code>	1, 27
<code>\linenum@out</code>	1
<code>\linenumannotation</code>	1, 55
<code>\linenumberlist</code>	1, 21
<code>\linenumberstyle</code>	1, 23
<code>\linenumincrement</code>	1, 21
<code>\linenummargin</code>	1, 21
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 22
<code>\linerangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 22
Lorch, Richard	13
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	81

M

<code>\m@mmf@check</code>	1
<code>\m@mmf@prepare</code>	1
<code>\M@sect</code>	1
<code>\makehboxofhboxes</code>	1
<code>\managestanza@modulo</code>	1
<code>\maxhnotesX</code>	50

Mayer, Gyula	13
\measurebody	<u>1</u>
\measurecell	<u>1</u>
\measuremrow	<u>1</u>
\measurtbody	<u>1</u>
\measuretcell	<u>1</u>
\measuretrow	<u>1</u>
Middleton, Thomas	14, 105
minipage (environment)	58
Mittelbach, Frank	13
\morenoexpands	<u>1</u> , 76
\mp@append@notesX	<u>1</u>
\mp@append@Xnotes	<u>1</u>
\mpfnpos	<u>1</u> , 37
\mpnormalfootgroup	<u>1</u>
\mpnormalfootgroupX	<u>1</u>
\mpnormalvfootnote	<u>1</u>
\mpnormalvfootnote@inserted	<u>1</u>
\mpnormalvfootnoteX	<u>1</u>
\mppara@footgroupX	<u>1</u>
\mppara@vfootnoteX	<u>1</u>
\mpparafootgroup	<u>1</u>
\mpparavfootnote	<u>1</u>
\mpthreecolfootgroup	<u>1</u>
\mpthreecolfootgroupX	<u>1</u>
\mpthreecolfootsetup	<u>1</u>
\mpthreecolfootsetupX	<u>1</u>
\mptwocolfootgroup	<u>1</u>
\mptwocolfootgroupX	<u>1</u>
\mptwocolfootsetup	<u>1</u>
\mptwocolfootsetupX	<u>1</u>
\msdata	<u>1</u> , 34
\msdata@c	<u>1</u>
\msdata@cR	<u>1</u>
\multfootsep	<u>1</u> , 36
\multiplefootnotemarker	<u>1</u>

N

\n@num	<u>1</u>
\n@num@stanza	<u>1</u>
\new@line	<u>1</u>
\newhookarg@specific	<u>1</u>
\newhookcommand@series	<u>1</u>
\newhookcommand@series@reload	<u>1</u>
\newhooktoggle@series	<u>1</u>
\newhooktoggle@series@reload	<u>1</u>
\newhooktoggle@specific	<u>1</u>
\newseries@	<u>1</u>
\newverse	<u>1</u>
\NEXT	<u>1</u>

<code>\next@line@list@stuff</code>	1
<code>\no@expands</code>	1
<code>\noeledsec</code>	72
<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>	46
<code>\notenumfontX</code>	45
<code>\noteschanged@false</code>	1
<code>\noteschanged@true</code>	1
<code>\noteswidthliketwocolumnsX</code>	50
<code>\nulledindex</code>	1
<code>\nullsetzen</code>	1
<code>\num@lines</code>	1
<code>\numberedpar@false</code>	1
<code>\numberedpar@true</code>	1
<code>\numberingfalse</code>	1
<code>\numberingtrue</code>	1
<code>\numberlinefalse</code>	20
<code>\numberlinetrue</code>	20
<code>\numberpstartfalse</code>	1, 19
<code>\numberpstarttrue</code>	1, 19
<code>\numberstanzafalse</code>	55
<code>\numberstanzatrue</code>	55
<code>\numlabfont</code>	1, 51

O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>optionafter</code>	57
<code>optionauxdir</code>	15, 455
<code>optionbefore</code>	57
<code>optioncontinuousnumberingwithcolumns</code>	91, 159, 165, 454, 459
<code>optioninnnote</code>	453
<code>optioninnnote</code>	453

optionlinrangesep	274
optionnocritical	453
optionnoeledsec	370, 455
optionnoend	453
optionnopenalties	75
optionnotenumber	453
optionsameparallelpagenumber	459
optionswcaseinsensitive	29, 458

P

\page@action	1
\page@num	1
\pagelinesep	1, 66
\pageref	61
\par@line	1
\par@patch@pagenumbering	1
\par@patch@thepage	1
\para@footgroupX	1
\para@footsetup	1
\para@footsetupX	1
\para@vfootnoteX	1
\parafootfmt	1
\parafootfmtX	1
\parafootgroup	1
\parafootsepX	48
\parafootstart	1
\parafootstartX	1
\paravfootnote	1
\parindentX	46
\parse@annot	1
\pausenumbering	1, 20
\pausenumbering@page@num	1
\pend	1, 17
Plato of Tivoli	13
\postbodyfootmark	1
\prebodyfootmark	1
\prenotesX	49
\prepare@edindex@fornote	1
\prepare@prenotesX	1
\prepare@Xgroupbyline	1
\prepare@Xprenotes	1
\prev@nopb	1
\prev@pb	1
\prevpage@num	1
\print@eledsection	1
\print@footnoteXrule	1
\print@leftmargin@eledsection	1
\print@lemma	1
\print@line	1
\print@notesX	1

<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, 17
<code>\pstarteref</code>	1
<code>\pstartnum</code>	1
<code>\pstartref</code>	59
<code>\pstarts@read@L</code>	1
<code>\pstarts@typeset@L</code>	1

Q

<code>\quotation</code>	1
<code>\quote</code>	1

R

<code>\raggedX</code>	48
<code>\raw@text</code>	1
<code>\rbracket</code>	1
<code>\read@linelist</code>	1
<code>\ref</code>	61
<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@current@annot</code>	1
<code>\reset@msd@options@iffullpage</code>	1
<code>\resetprevline@</code>	1, 109
<code>\resetprevpage@</code>	1
<code>\resetprevpage@num</code>	109
<code>\restore@endnotes</code>	1
<code>\restore@familiarnotes</code>	1
<code>\restore@notes</code>	1
<code>\restore@sidenotes</code>	1
<code>\resumenumbering</code>	1, 20
<code>\rightctab</code>	1
<code>\rightlinenum</code>	1, 22

<code>\rightltab</code>	1
<code>\rightnoteupfalse</code>	65
<code>\rightrtab</code>	1
<code>\rightstartnum</code>	1
<code>\rigidbalance</code>	1
<code>\rigidbalanceX</code>	1
<code>\rtab</code>	1
<code>\rtabtext</code>	1

S

<code>Sacrobosco</code>	13
<code>\sameword</code>	1, 28
<code>\sameword@inedtext</code>	1
<code>Schöpf, Rainer</code>	13
<code>\section@num</code>	1
<code>\select@@lemmafont</code>	1
<code>\select@lemmafont</code>	1, 51
<code>\SEref</code>	1, 61
<code>\SErefonlypage</code>	61
<code>\SErefwithpage</code>	1, 61
<code>\series</code>	1
<code>\seriesatbegin</code>	1, 37
<code>\seriesatend</code>	1, 37
<code>\set@continuousnumberingforL</code>	1
<code>\set@line</code>	1
<code>\set@line@action</code>	1
<code>\set@txtbeforenotesX</code>	1
<code>\set@Xtxtbeforenotes</code>	1
<code>\setapprefprefixmore</code>	62
<code>\setapprefprefixsingle</code>	62
<code>\setcommand@series</code>	1
<code>\sethangingsymbol</code>	1, 53
<code>\setistwofollowinglines</code>	1
<code>\setl@dlp@rbox</code>	1
<code>\setl@drpr@box</code>	1
<code>\setline</code>	1, 22
<code>\setlinenum</code>	1, 23
<code>\setmcellcenter</code>	1
<code>\setmcellleft</code>	1
<code>\setmcellright</code>	1
<code>\setmrowcenter</code>	1
<code>\setmrowleft</code>	1
<code>\setmrowright</code>	1
<code>\setmsdatalabel</code>	1, 35
<code>\setmsdataposition</code>	1, 35
<code>\setmsdataseries</code>	1, 35
<code>\setnoteswidthliketwocolumnsX@</code>	1
<code>\setnotesXpositionliketwocolumns@</code>	1
<code>\setprintendlines</code>	1
<code>\setprintlines</code>	1

<code>\setSErefonlypageprefixmore</code>	62
<code>\setSErefonlypageprefixsingle</code>	62
<code>\setSErefprefixmore</code>	62
<code>\setSErefprefixsingle</code>	62
<code>\setsidenotesep</code>	65
<code>\setstanzaindents</code>	1, 52
<code>\setstanzapenalties</code>	1, 53
<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, 74
<code>\showwordrank</code>	1, 32
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 64
<code>\sidepstartnumtrue</code>	19
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 23
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 69
<code>\spreadtext</code>	1, 69
<code>\stanza</code>	1, 51
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 53
<code>\stanzaindent*</code>	1, 53
<code>\stanzaindentbase</code>	1, 52
<code>\stanzanumwrapper</code>	1, 55
<code>\startlock</code>	1, 22
<code>\startsub</code>	1, 22
<code>\stepl@dcolcount</code>	1
<code>\stopmsdata</code>	1, 34
<code>\store@annot@to@absline</code>	1
<code>\strip@szacnt</code>	1
<code>\sub@action</code>	1
<code>\sub@lock</code>	1
<code>\sub@off</code>	1
<code>\sub@on</code>	1
<code>\subline@num</code>	1
<code>\sublinenumberstyle</code>	1, 23
<code>\sublinenumincrement</code>	1, 21
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1

<code>\sublineref</code>	1, 59
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	13, 14, 75, 90, 95, 190, 191, 283, 323
<code>\sw@noexpand</code>	1
<code>\swnoexpands</code>	32
<code>\sza@penalty</code>	1

T

<code>\tabHilfbox</code>	1
<code>\tabhilfbox</code>	1
<code>\theadcolcount</code>	1
<code>\theadtext</code>	1
<code>\theendpageline</code>	1
<code>\tfootnoteA</code>	35
Theodosius	13
<code>\thepageline</code>	1
<code>\thepar@page</code>	1
<code>\thepstart</code>	1, 19
<code>\thestanza</code>	1, 55
<code>\thestartpageline</code>	1
<code>\this@line@list@version</code>	1
<code>\this@section@last@page@num</code>	1
<code>\this@section@page@num</code>	1
<code>\threecol@begin@insert</code>	1
<code>\threecolfootfmt</code>	1
<code>\threecolfootfmtX</code>	1
<code>\threecolfootgroup</code>	1
<code>\threecolfootgroupX</code>	1
<code>\threecolfootsetup</code>	1
<code>\threecolfootsetupX</code>	1
<code>\threecolvfootnote</code>	1
<code>\threecolvfootnote@inserted</code>	1
<code>\threecolvfootnoteX</code>	1
<code>\toendnotes</code>	1, 26
<code>\toendnotes*</code>	1
<code>\twocolfootfmt</code>	1
<code>\twocolfootfmtX</code>	1
<code>\twocolfootgroup</code>	1
<code>\twocolfootgroupX</code>	1
<code>\twocolfootsetup</code>	1
<code>\twocolfootsetupX</code>	1
<code>\twocolvfootnote</code>	1
<code>\twocolvfootnote@inserted</code>	1
<code>\twocolvfootnoteX</code>	1
<code>\txtbeforenotesonlyonceX</code>	49
<code>\txtbeforenotesX</code>	49

U

<code>\unvzxhX</code>	1
-----------------------------	---

V

Vamana	13
<code>\variab</code>	1
<code>\vbfnoteX</code>	1
<code>\vl@dbfnote</code>	1
<code>\vl@dcnote</code>	1
<code>\vl@disnote</code>	1
<code>\vl@dlsnote</code>	1
<code>\vl@dosnote</code>	1
<code>\vl@drsnote</code>	1
<code>\vnumfootnoteX</code>	1

W

Whitney, Ron	13
<code>\widthX</code>	50
<code>\wrap@edcrossref</code>	1
<code>\wrapcontentX</code>	46
<code>\wrapped@bodyfootmarkX</code>	1
<code>\wrapped@footfootmarkX</code>	1
Wujastyk, Dominik	12

X

<code>\X@atbegininsertion</code>	1
<code>\X@beforeinsertion</code>	1
<code>\X@doreinfeet</code>	1
<code>\xabslineref</code>	1
<code>\Xafterlemmaseparator</code>	44
<code>\Xafternote</code>	48
<code>\Xafternumber</code>	42
<code>\Xafterrule</code>	49
<code>\Xaftersymmlinum</code>	42
<code>\xannotationref</code>	1
<code>\xannotref</code>	60
<code>\Xarrangement</code>	1, 38
<code>\Xarrangement@normal</code>	1
<code>\Xarrangement@paragraph</code>	1
<code>\Xarrangement@threecol</code>	1
<code>\Xarrangement@twocol</code>	1
<code>\Xbeforeinserting</code>	47
<code>\Xbeforelemmaseparator</code>	44
<code>\Xbeforenotes</code>	49
<code>\Xbeforenumber</code>	40, 42
<code>\Xbeforesymmlinum</code>	42
<code>\Xhookgroup</code>	49
<code>\Xhooknote</code>	47
<code>\Xboxlinenum</code>	43
<code>\Xboxlinenumalign</code>	43

<code>\Xboxsymlinenum</code>	43
<code>\Xcolalign</code>	47
<code>\xedindex</code>	<u>1</u>
<code>\xedlabel</code>	<u>1</u>
<code>\xedtext</code>	<u>1</u>
<code>\Xendafterlemmaseparator</code>	45
<code>\Xendafternote</code>	51
<code>\Xendafterpagenumber</code>	40
<code>\Xendaftersymlinenum</code>	42
<code>\Xendaftrenumber</code>	42
<code>\Xendahookinplaceofnumber</code>	44
<code>\Xendahooklinenumber</code>	44
<code>\Xendbeforelemmaseparator</code>	45
<code>\Xendbeforenumber</code>	42
<code>\Xendbeforepagenumber</code>	40
<code>\Xendbeforesymlinenum</code>	42
<code>\Xendbhookinplaceofnumber</code>	44
<code>\Xendbhooklinenumber</code>	44
<code>\Xendbhooknote</code>	47
<code>\Xendboxendlinenumalign</code>	44
<code>\Xendboxlinenum</code>	44
<code>\Xendboxlinenumalign</code>	44
<code>\Xendboxstartlinenumalign</code>	44
<code>\Xendboxsymlinenum</code>	43
<code>\Xendhangindent</code>	46
<code>\Xendinplaceoflemmaseparator</code>	45
<code>\Xendinplaceofnumber</code>	43
<code>\Xendinplaceofpagenumber</code>	40
<code>\Xendlemmadisablefontselection</code>	45
<code>\Xendlemmafont</code>	45
<code>\Xendlemmaseparator</code>	44
<code>\Xendlinenumannotationposition</code>	57
<code>\Xendlineprefixmore</code>	40
<code>\Xendlineprefixsingle</code>	40
<code>\Xendlinerangeseparator</code>	40
<code>\Xendmorethantwolines</code>	41
<code>\Xendnonumber</code>	41
<code>\Xendnotefontsize</code>	46
<code>\Xendnotenumfont</code>	45
<code>\Xendnumberonlyfirstinline</code>	39
<code>\Xendnumberonlyfirstintwolines</code>	39
<code>\Xendpagenumberonlyfirst</code>	39
<code>\Xendparagraph</code>	51
<code>\Xendsep</code>	51
<code>\Xendsublinesep</code>	42
<code>\Xendsymlinenum</code>	39
<code>\Xendsympagenum</code>	40
<code>\Xendtwolines</code>	41
<code>\Xendtwolinesbutnotmore</code>	41
<code>\Xendtxtbeforenotes</code>	49

\Xendwrapcontent	46
\xflagref	1
\Xgroupbyline	48
\Xgroupbylineseparetwolines	48
\Xhangindent	46
\Xhsizethreecol	47
\Xhsizetwocol	47
\Xinplaceoflemmaseparator	44
\Xinplaceofnumber	43
\Xinsertparafootsep	1
\Xledsetnormalparstuff	1
\xleft@appenditem	1
\Xlemmadisablefontselection	45
\Xlemmafont	45
\Xlemmaseparator	44
\Xlinenumannotationposition	57
\Xlinenumannotationposition@side	1
\Xlinerrangeseparator	40
\xlineref	1, 60
\Xmaxhnotes	50
\Xmorethantwolines	40
\Xnolemmaseparator	1, 44
\Xnonbreakableafternumber	42
\Xnonumber	41
\Xnotefontsize	45
\Xnotenumfont	45
\Xnoteswidthliketwocolumns	50
\Xnumberonlyfirstinline	39
\Xnumberonlyfirstintwolines	39
\Xonlypstart	41
\Xpagelinesep	42
\xpageref	1, 60
\Xparafootsep	48
\Xparindent	46
\Xprenotes	1, 49
\Xprenotes@	1
\Xpstart	41
\Xpstarteverytime	41
\xpstartref	1, 60
\XR@test	1
\XR@test@mac	1
\XR@test@mac@test	1
\Xragged	48
\xright@appenditem	1
\Xrigidbalance	1
\Xstanza	42
\Xstanzaseparator	42
\Xstorelineinfo	1
\xsublineref	1, 60
\Xsublinesep	22, 42

<code>\Xsublinesepside</code>	22
<code>\Xsymlinenum</code>	39
<code>\Xtoendnotes</code>	26
<code>\Xtwolines</code>	40
<code>\Xtwolinesonlyinsamepage</code>	41
<code>\Xtxtbeforenotes</code>	49
<code>\Xtxtbeforenotesonlyonce</code>	49
<code>\Xunvxh</code>	<u>1</u>
<code>\Xwidth</code>	50
<code>\Xwrapcontent</code>	46
<code>\Xwrapendlemma</code>	46
<code>\Xwraplemma</code>	46
<code>\Xwraplinenumannotation@ref</code>	<u>1</u>
<code>\Xwraplinenumannotation@side</code>	<u>1</u>
<code>\xxref</code>	<u>1</u> , 60

Z

<code>\zz@@@</code>	<u>1</u>
<code>\zz@@@@</code>	<u>1</u>

Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
\ifl@dmemoir: Added \ifl@dmemoir for memoir class having been used	82
\morenoexpands: Added \l@dtabnoexpands to \no@expands	135
\reledmac@error: Added \eledmac@error and replaced error messages	83
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	286
\doxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	275
\edlabel: Tweaked \edlabel to get correct page numbers	283
\l@ddodoreinxtrafeet: Renamed \dodoreinxtrafeet to \l@ddodoreinxtrafeet	276
\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	135
General: Added text about normal labeling	61
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	278
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	189
\next@line@list@stuff: Added initial write of page number in \line@list@stuff	126
\para@footsetup: Added \footfudgefiddle to \para@footsetup	189
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	230
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	89
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	286
\@n@reg: Added a bunch of code to \@n@ for handling \setlinenum	112
\ledlinenum: Added \linenumr@p and \sublinenumr@p to \leftlinenum and \rightlinenum	103
\linenumberlist: Added \linenumberlist mechanism	90
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	249
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	213
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	103
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	309
\Xarrangement@normal: Added minpage footnote setup to \footnormal	183
\edtext: Added \showlemma to \edtext	136
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	307
\mpnormalfootgroup: Added \mpnormalfootgroup	187

\mpnormalvfootnote: Added \mpnormalvfootnote	185
\showlemma: Added \showlemma	90
General: Added final/draft options	79
Added ledgroup environment	311
Added ledgroupsize environment	311
Added minipage, etc., support	1
v0.4.1.	
\doXfeet: Changed \doXfeet code for easier extensions	276
\edindex: Let eledmac take advantage of memoir's indexing	316
\printXnotes: Added \opXfeet	276
General: Added code for changing \doclearpage	279
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	216
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	217
\@xympar: Eliminated \marginpar disturbance	299
General: Added left and right sidenotes	300
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	168
\l@getsidenote@margin: Added \sidenotemargin and \sidenote@margin	300
General: Added moveable sidenote	300
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	311
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	115
\@nl@reg: Extended \@nl to include the page number	112
\get@thisfootnote: Changed \l@dbfnote and \v1@dbfnote as originals could give incorrect markers in the footnotes	217
\l@the@cc@page: Extended \new@line to output page numbers	127
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	112
\@ref@reg: Added \@ref@reg	122
\affixline@num: Added skipnumering to \affixline@num	168
\do@actions@fixedcode: Added \do@actions@fixedcode	164
\do@actions@next: Added number skipping to \do@actions	163
\do@insidelinehook: Added \do@linehook for use in \do@line	160
\endnumbering: Changed \endnumbering for eledpar	94
\fix@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fix@l@cks	171
\footplitskips: Added \footplitskips for use in many footnote styles	182
\get@linelistfile: Added \get@linelistfile	111
\initnumbering@reg: Added \initnumbering@reg	92
\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	310
\l@dcsnotetext@r: Added \l@emptyd@ta	161

\l@dgetline@margin: Added \l@dgetline@margin	99
\l@dgetlock@disp: Added \l@dgetlock@disp	102
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	300
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	90
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	161
\l@dzeropenalties: Added \l@dzeropenalties	155
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	103
\list@clearing@reg: Added \list@clearing@reg	111
\n@num: Added \n@num	120
\next@line@list@stuff: Deleted \page@start from \line@list@stuff	126
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	223
\pausenumbering@page@num: Changed \resumenumbering for eledpar	95
\setprintendlines: Added \setprintendlines for use by \printendlines	246
\setprintlines: Added \setprintlines for use by \printlines	208
\skipnumbering: Added \skipnumbering and supports	132
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	101
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	103
Using \sublinenumrep instead of \sublinenumr@p	103
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	224
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	83
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	90
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	19
Possibility to number the pstart with the commands \numberpstarttrue.	1
v0.12.1.	
General: Don't number \pstarts 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.	325
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	52
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 <code>memoir</code> 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.	283
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.	168
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	324
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code> .	212
General: Line numbering can be reset at each <code>pstart</code> .	98
Possibility to print <code>\pstart</code> number inside.	19
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	324
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	135
General: <code>\lemma</code> can contain commands.	27
Debug in lineation command	21
New generic commands to customize footnote display.	38, 266
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code> .	25
Options of <code>\Xfootnotes</code> .	179
Possibility to have commands in sidenotes.	64
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.	39
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code>	238
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	266
General: Add <code>\labelpstarttrue</code> .	19
Add <code>\Xnumberonlyfirstintwolines</code>	39
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	41
New hook to add arbitrary code at the beginning of the notes	46
New options for block of notes.	49
New package option: <code>parapparatus</code> .	1
New tools to change order of series	265
Sectioning commands.	71
v1.2.0.	
<code>\Xprenotes</code> : Debug in familiar footnotes (bug introduced by v1.1).	238
<code>\endquote</code> : Compatibility of <code>\ledchapter</code> with the <i>memoir</i> class.	363

v1.3.0.		
\endquote:	<i>Quotation</i> and quote environment inside numbered sections.	363
v1.4.0.		
\edtext:	Compatibility of \edtext with the right-to-left direction (with Polyglossia).	136
\ledsetnormalparstuffX:	Direction of footnotes with polyglossia.	235
\newseries@:	Remembers the language of the lemma, in order to create a correct direction for the footnote separator.	255
\rbracket:	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX).	202
General:	Compatibility with LuaTeX of RTL notes.	1
v1.4.1.		
\affixside@note:	Remove spurious spaces.	306
\endquote:	New option <i>noquotation</i>	363
\get@thisfootnote:	Compatibility of standard footnotes with eledmac when these footnotes contain any commands.	217
\labelrefsparsesubline:	Fix a bug with \edlabel.	285
v1.4.2.		
General:	Debug with some special classes.	1
v1.4.3.		
General:	Add \Xnonbreakableafternumber.	42
Spurious space after familiar footnotes.	1
v1.4.4.		
General:	Label inside familiar footnotes.	1
v1.4.5.		
General:	Bug with komasscript + eledpar + chapter.	1
v1.4.6.		
General:	Bug with memoir class introduced by 1.4.5.	1
v1.4.7.		
\endquote:	Compatibility of sectioning commands with \autopar.	363
v1.4.8.		
General:	Corrects a bug with parallel texts introduced by 1.1.	1
v1.4.9.		
\normalbfnoteX:	Allow to redefine \thefootnoteX with alph when some packages are loaded.	223
v1.5.0.		
\do@insidelinehook:	Added \do@insidelinehook for use in \do@line	160
\edindex:	Compatibility with imakeidx package, and possibility to use multiple index with \edindex.	316
General:	Correct indexing when the call is made in critical notes.	312
v1.5.1.		
\managestanza@modulo:	Correct stanzaindentsrepetition counter	325
\normalvfootnoteX:	Fix a bug with normal familiar footnotes when mixing RTL and LTR text.	220
v1.6.0.		
\newverse:	Add \falseverse macro.	328
v1.6.1.		
\AtStartEveryPstart:	Spurious space in \pstart.	151
\ifinserthangingsymbol:	Hang verse is now not automatically flush right.	324
\ldunhbox@line:	Move the call to \inserthangingsymbol to allow use \hfill inside.	157

\pend: Spurious space in \pend.	153
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.	73
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts.	363
\get@index@command: Debug \get@index@command and compatibility with hyperref package.	315
\newhookcommand@series@reload: Debug \beforenotesX and \maxnotesX which did not work.	269
\prevpage@num: Correct \parafootsep when using with ledgroup.	194
General: Compatibility with parledgroup option of eledpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	312
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	240
v1.8.2.	
General: Debug compatibility problem with hebrew option of babel package.	1
v1.8.3.	
General: Fixes spurious spaces added by v1.7.0.	1
v1.8.5.	
General: Debug indexing in right column, with eledpar.	312
v1.9.0.	
\doextrafeet: Add \fnpos to choice the order of footnotes.	275
\l@dfetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup.	307
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	363
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.	45
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, \@1 becomes \@l. Consequently, \@1@reg becomes \@nl@reg.	112

\AtStartEveryPstart: New optional argument for \pstart, to execute code before it.	151
\edindex: Use correctly default index when imakeidx is loaded.	316
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	363
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar.	92
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	300
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcnote defined only one time, in eledmac, including needs for eledpar case.	301
\l@dnumbstartsL: Add \ifledRcol@ for eledpar	90
\l@dunhbox@line: \do@line is split in more little commands.	157
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	269
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	269
\pend: New optional argument for \pend, to execute code after it.	153
\stanza: &can have an optional argument: content to be printed after.	328
\Stanza can have an optional argument: content to be printed before.	328
Add \newverse macro, \falseverse deprecated.	328
General: Add \ledinnernote and \ledouternote commands.	64
Add \Xendparagraph and related settings.	51
Add hyperlink to crossref (needs hyperref package).	59
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	312
New hooks: \Xafterrule and \afterruleX	49
New options for ragged-paragraph notes	48
New sectioning commands.	71
Optional arguments for \pstart and \pend.	18
v1.12.1.	
\wrap@edcrossref: Fix spurious spaces.	289
v1.12.2.	
\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	157
v1.12.3.	
\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	128
\flag@start send a error message when a \edtext is done without insert (note)	128
\reledmac@error: Replaced error messages	83
General: Add macros for new messages since v0.7	83
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.	64
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	318
\@ssect: Debug \eledchapter in parallel with memoir	366
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	160

\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	94
\l@dgobblearg: \l@dgobblearg becomes \l@dgobbeloptarg	346
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	346
\pausenumbering@page@num: Debug \resumenumbering	95
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	289
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	128
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	269
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	230
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	266
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	50
Added widthliketwocolumns option	79
v1.13.1.	
\ifat@every@pstart@star@: Add \l@dzeropenalties in \pstart	151
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@dpaging for eledpar	90
General: Fix a 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 a bug with index when memoir class is used without hyperref	1
v1.14.0.	
\edindex: Let eledmac take advantage of imakeidx even when memoir class is used	316
General: Debug spurious characters before endnotes.	240
Delete previous override of \l@d@@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	320
Move gobbling command	89
Provide \@gobblefour	89
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref	
package.	368
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	366
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	136
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	230
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	130
\if@lemmacommand@: New boolean \iflemmacommand@.	141
General: Fix a bug with footnotes layout when using some options of the geometry	
package (bug add by v1.13.0).	1
New commands \AtEveryPstart and \AtEveryPend.	19
New tools to prevent ambiguous references in lemma	28

v1.15.1.	
\next@line@list@stuff:	Revert modification of 1.5.2 which makes bug with numbering. Leave vertical mode to solve spurious space before minipage.
	126
v1.16.0.	
General:	\edtext is now defined only in eledmac, not in eledpar. Debug wrong numbering when using \sameword + eledpar + \tag command.
	136
Compatibility of standard footnotes with some biblatex styles.
	1
New \stanzaindent command.
	1
v1.16.1.	
\xlineref:	\lineref is not defined if defined by some other package, like lineno. Eledmac provides \edlineref instead.
	289
v1.17.0.	
\edtext:	Error message when calling \edtext outside of a numbered paragraph. . . .
	136
v1.18.0.	
\@edindex@hyperref:	Fix spurious space with \edindex when using imakeidx/indextools + hyperref.
	318
\edlabel:	\edlabel is now defined only one time for both eledmac and eledpar
	283
\l@d@section:	Option parapparatus works for endnotes.
	241
\l@dnumpstartsl:	Add \ifl@dprintingpages and \@dprintingcolumns for eledpar
	90
\print@line:	Compatibility with Lua \TeX RTL languages.
	158
\printlinefootnote:	Code refactoring in \printlinefootnote: the printing of the numbers are factorized in \printlinefootnotearea
	204
\printpstart:	Debug \Xpstart with parallel pages and columns (eledpar)
	203
General:	Add \Xpstarteverytime
	41
Compatibility with Lua \TeX RTL languages.
	1
Debug \Xonlypstart when using \Xnumberonlyfirstinline and the current line number differs from the previous.
	41
v1.19.0.	
\footplitskips:	\footplitskips doesn't set \floatingpenalty to \@MM when processing parallel pages.
	182
\xxref:	\xxref works also with right side numbers, when \@Rlineflag is not empty. . . .
	292
General:	\Xmaxhnotes and \maxhnotesX work now for both two-columns and three-columns setting.
	1
Compatibility with eledpar v1.13.0.
	1
v1.19.1.	
General:	Call \correct@footinsX@box and \correct@Xfootins@box directly in \print@notesX@forpages and \print@Xnotes@forpages, that is in eledpar. . . .
	1
v1.20.0.	
\printlines:	Added \ifl@d@Xmorethantwolines and \ifl@d@Xmorethantwolines to \printlines
	213
\stanza:	& and \& can be preceded by spaces.
	328
\xxref:	Debug \xxref when not loading eledpar (Fix a bug added in 1.19.0).
	292
General:	Add \Xendboxlinenum
	44
Add \Xtwolines and \Xmorethantwolines hooks
	40
Add series option.
	1
Correct \Xinplaceofnumber hook.
	1
Explicit error message when calling \Xfootnote outside of \edtext.
	1
Fix a bug with line number typesetting direction when using \eledsection and similar commands for RTL texts with Lua \TeX
	1

Fix issues with RTL text in notes when using Lua \TeX .	1
Options fulllines in \backslash Xfootnote.	25
The \backslash newifs are not followed by boolean values set to false, because it is the \TeX default setting.	1
v1.21.0.	
\backslash @edindex@hyperref: Look at the hyperindex option of hyperref before inserting hyperref	318
\backslash l@d@section: \backslash endnotes take five arguments.	241
\backslash ledinnotemark: Add \backslash ledinnotemark.	315
\backslash ledsetnormalparstuffX: \backslash ledsetnormalparstuff is deprecated and becomes \backslash ledsetnormalparstuffX and \backslash Xledsetnormalparstuff.	235
\backslash n@num: \backslash n@num@ref deleted	120
\backslash n@num defined only one time for both Eledmac and Eledpar	120
\backslash newhookcommand@series: \backslash newhookcommand@series can take an optional argument.	268
\backslash newhooktoggle@series: \backslash newhooktoggle@series can take an optional argument.	269
\backslash print@footnoterule: Code refactoring: the spaces after the footnote rules are directly managed in \backslash print@Xfootnoterule and \backslash print@footnoterule	237
\backslash seriesatend: Fix spurious space in \backslash seriesatend	265
\backslash skipnumbering: \backslash skipnumbering defined only one time for both Eledmac and Eledpar.	132
Correct \backslash skipnumbering for stanza.	132
Delete \backslash skipnumbering@reg.	132
General: \backslash AtEveryPstart and \backslash AtEveryPend are now compatible with \backslash autopar	1
\backslash Xafterrule and \backslash afterruleX features no longer create problems of overflowing at the bottom of the page.	1
\backslash chapter inside optional argument of \backslash pstart works when typesetting parallel pages	1
\backslash preXnotes and \backslash prenotesX features no longer create problems of overflowing at the bottom of the page.	1
\backslash seriesatbegin and \backslash seriesatbegin more efficient	265
Add \backslash applabel and related	62
Add \backslash beforenotesX and \backslash Xbeforenotes features for notes set in two and three column.	1
Add \backslash hidenummering	23
Add \backslash Xcolalign and \backslash colalignX hooks	47
Add \backslash Xendtwolines, \backslash Xendmoreethantwolines, \backslash Xendtwolinesbutnotmore and \backslash Xendtwolinesonlyinsamepage.	41
Add \backslash Xparindent and \backslash hangindentX	46
Add \backslash Xtwolinesbutnotmore and \backslash Xtwolinesonlyinsamepage.	1
Add nocritical, noend, nofamiliar and noledgroup options.	1
Add noeledsec package option	1
Debug \backslash beforenotesX \backslash maxhnotesX \backslash noteswidthliketwocolumnsX and \backslash afterruleX with footnotes set in two and three columns.	1
Fix a bug when a \backslash Xfootnote follows a \backslash Xendnote in the second argument of \backslash edtext (bug added in eledmac 1.0.0).	1
Fix a bug with \backslash maxhnotesX when using \backslash foottwocolX or \backslash foottwocolX.	1
Fix a 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>	89
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	315
General: Add <code>\doendnotesbysection</code> command.	26
Add option for lemma separator inside endnotes	44
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.	26
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> .	194
General: Fix a 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> .	136
<code>\Serefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmoreethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code> .	296
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	141
<code>\newsseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	255
<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.	146
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code> .	43
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code> .	43
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 a bug with <code>\sameword</code> when the lemma overlaps multiple line.	28
Fix a bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code> .	28
Fix a bug with <code>\skipnumbering</code> called immediately after a <code>\pstart</code> .	1
Fix error of <code>\iftrue</code> not closed.	1
Fix spurious space with <code>\skipnumbering</code> (bug added on v1.21.0).	1
New tools to ensure the line-list file uses the right version of commands when upgrading the <code>eledmac</code> version.	1
Optional argument of <code>\sameword</code> can be a comma-separated list of <code>\edtext</code> depth.	28
v1.23.1.	
General: Fix a 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 a 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.	309
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	299
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	310
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	362
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some code of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	310
<code>\newseries@</code> : One endnotes file by series.	261
General: <code>\makecol</code> , <code>\reinserts</code> and <code>\doclearpage</code> are patched instead of begin redefined	278
<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>	278
Add <code>\Xendinplaceofnumber</code> hook.	1
Add <code>\Xendnonumber</code> hook.	1
Add <code>nonum</code> option for endnotes.	1
Fix a 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	216
Suppress indexing command specific to <code>memoir</code>	316
v2.0.1.	
General: Fix a 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 a bug with <code>\advanceline</code> at the beginning of a <code>\pstart</code>	1
Fix a bug with <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting with <code>scrbook</code>	1
Fix a bug with <code>\eledchapter</code> in parallel typesetting with <code>scrbook</code>	1
Fix a 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 a bug with <code>\ledpbsetting{before}</code>	1
v2.1.2.	
General: Fix a 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.	235
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 a bug which inserted double space before paragraphed familiar notes.	1
Fix a bug with <code>\edindex</code> when using not-Latin characters without UTF-8 engines	1
v2.2.0.	
General: Fix a bug with combination of <code>\onehalfspacing</code> and two columns and three columns notes typeset.	1
Fix a bug with some setting command and optimization option.	1
Fix spurious space with paragraphed critical notes when using $\text{Lua}\TeX$	1
Increase line list version number to ensure compatibility with new options of <code>reledpar</code> package.	1
New setting tools for endnotes: <code>\Xendnumberonlyfirstinline</code> , <code>\Xendnumberonlyfirstintwolines</code> , <code>\Xendsymmlinenumber</code> , <code>\Xendbeforenumber</code> , <code>\Xendafternumber</code> , <code>\Xendbeforemsymmlinenumber</code> , <code>\Xendaftersymmlinenumber</code> , <code>\Xendboxsymmlinenumber</code> , <code>\Xendhangindent</code> , <code>\Xendbhooklinenumber</code> , <code>\Xendahooklinenumber</code> , <code>\Xendbhookinplaceofnumber</code> , <code>\Xendahookinplaceofnumber</code>	1
v2.2.1.	
General: Compatibility with TeXformat 2015/10/01.	1
v2.2.2.	
General: Fix a bug in <code>\sethangingsymbol</code>	1
Fix a bug with old version of <code>etex</code>	1
v2.3.0.	
General: Disable empty lines as paragraph in stanza.	1
Fix incompatibility of paragraphed footnotes with <code>bidi</code> v17.9 and following.	1
Warning message when using some setting commands inside <code>rightside</code> environment (deprecated behavior)	1
v2.3.1.	
General: Fix spurious space when using optional argument of <code>\stanza</code> (introduced in v2.3.0).	1
v2.4.0.	
<code>\footnoteoptions@</code> : First argument of <code>\footnoteoption@</code> is now mandatory, not optional.	179
General: <code>\Xbhooknote</code> and <code>\bhooknoteX</code> work with notes in columns.	1
Fix a bug of <code>\parindentX</code> and <code>\Xparindent</code> with two columns and three columns notes.	1
Fix a 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>	40

Option linerangesep for critical footnotes and endnotes.	40
v2.4.1.	
General: Fix a bug with <code>\appref</code> and <code>\apprefwithpage</code> (introduced in v2.4.0).	1
Fix a bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages	
that override \TeX <code>\roman</code> command, like Greek language.	1
Fix a bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages	
that override \TeX <code>\roman</code> command, like Greek.	1
v2.5.0.	
<code>\Serefwithpage</code> : Debug <code>\setapprefprefixsingle</code>	295
<code>\edlabel</code> : Fix a bug when calling <code>\edlabel</code> in a footnotes of the right side	283
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments.	241
<code>\printlines</code> : <code>\printlines</code> takes an eighth argument: the line flag	212
<code>\xlineref</code> : <code>\xlineref</code> does not include anymore the side flag. Use <code>\xflagref</code> to get	
it. Not that <code>\edlineref</code> still contains the flag.	289
General: <code>\apprefwithpage</code> and <code>\appref</code> print double quotation mark when the label	
was not defined.	1
<code>\apprefwithpage</code> and <code>\appref</code> work with right side crossref.	1
<code>\apprefwithpage</code> works also when <code>noend</code> option is enabled.	1
<code>\appref</code> and <code>\apprefwithpage</code> can take <code>linerangesep</code> optional argument.	1
<code>\edlabel</code> works now in \TeX footnote.	1
<code>\lemma</code> can be used even when the <code>nocritical</code> is enabled.	1
Compatibility with new hook and tools of <code>reledpar</code> 2.6.0.	1
Fix spurious vertical space in <code>astanza</code> environment (<code>reledpar</code>)	1
Log now states ‘There were undefined references’ when using wrong references in	
<code>\edlineref</code> or <code>edpageref</code>	1
New hooks to customize page and line number appearance in endnotes.	1
New hooks: <code>\Xhookgroup</code> and <code>\hookgroupX</code>	1
New tools to easily make cross-reference to a passage defined by a start and an end line	
61	
v2.6.0.	
General: Adds compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code>	
package.	1
Fix a bug with footnote counter in <code>ledgroup</code> (added in v2.5.0).	1
Fix bug, introduced in v2.5.0, with footnote numbering in parallel typesetting when	
using <code>perpage</code> package.	1
v2.7.0.	
<code>\@k</code> : <code>\rigidbalance</code> is split in <code>\Xrigidbalance</code> and <code>\rigidbalanceX</code>	195
<code>\l@d@section</code> : <code>\endnotes</code> take seven arguments.	241
General: Add dash as default page range separator for <code>\SEonlypage</code>	1
Debug <code>\Serefonlypage</code> when referring to only one page.	1
Delete parenthesis after <code>\Serefonlypage</code>	1
Fix (again) bugs with footnote numbering in parallel typesetting while using	
<code>ledgroup</code> environments (bug added in v2.5.0).	1
Fix a bug with <code>\Serefwithpage</code>	1
Fix bugs in compatibility with <code>innote</code> and <code>notenumber</code> options of <code>indextools</code>	
package, when indexing outside of a <code>ledgroup</code>	1
New commands to make glossaries connected to page and linenumber with the	
<code>glossaries</code> package	1
New hooks: <code>\Xhsize</code> and <code>\hsizeX</code>	50
New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</code>	45

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>	50
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.	241
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 a bug concerning indent in a paragraph immediately following a sectioning command (bug NOT fixed on <code>reledpar</code>)	1
Fix a bug with <code>\AtEveryPstart</code> added in version 2.0.0.	1
Fix a 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 a bug when using <code>\lineneation{page}</code> and <code>\pausenumbering...</code> <code>\resumenumbering</code>	1
Fix a 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...</code> <code>\resumenumbering</code>	1
v2.9.1.	
General: Fix a bug when notes start with “plus” or “minus”.	1
v2.9.2.	
General: Fix a 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 a bug with line number position and reset added by v. 2.9.0	1

v2.10.0.	
\print@lemma: Code refactoring between \parafootfmt, \twocolfootfmt, \threecolfootfmt and \normalfootfmt.	203
General: Add \AtEveryStanza and \AtEveryStopStanza.	1
Fix a bug in \ledlsnotefontsetup and \ledrsnotefontsetup which could not handle \color command properly.	1
More specific error messages.	1
New hooks: \Xwrapcontent, \Xendwrapcontent and \wrapcontentX.	46
New hooks: \Xwraplemma and \Xendwraplemma	46
v2.10.1.	
General: Add ‘nopenalties’ option.	1
Fix a bug introduced in v. 1.4: not paragraphed critical footnotes could prevent marginal line number from being displayed	1
v2.11.0.	
\do@actions@fixedcode: Add action 1010	164
General: Add new tools to produce an apparatus of manuscripts	1
Fix a bug in \Xparafootsep in parallel typesetting	1
Make \parafootsepX work	1
Prevent \Xtxtbeforenotes hook from causing notes to go beyond the bottom margin	1
v2.12.0.	
General: \preXnotes becomes \Xprenotes (naming convention)	1
Add \hidenumberingonleftpage and \hidenumberingonrightpage	1
Add \toendnotes and related.	1
Add auxdir option.	1
Fix a bug in critical and familiar footnotes when using uppercase letters with accent mark	1
Fix a bug when using \chapter in optional argument of \pstart in parallel typesetting in combination with the noeledsec option.	1
Fix a bug with \ledinnernote and \ledouternote in parallel typesetting	1
Fix a bug with familiar footnote number in optional argument of \pstart or \pend in parallel typesetting	1
Fix spurious vertical space in \chapter when used as optional argument of \pstart in parallel typesetting.	1
Make endnote compatible with \sameword mechanism	1
More accurate message to control the position of \Xfootnote and \applabel in the \LaTeX 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 \Xlineflag (bug added in v. 2.5.0).	1
v2.13.2.	
General: Fix a bug added in v. 11.2 which could make parallel typesetting not work.	1
v2.13.3.	
General: Makes \Xendafterpagenumbe affecting \Serefwithpage	1
v2.14.0.	
General: Hyperref with the line number inside critical footnotes is correct when using \xxref	1
Some internal changes for new features of reledpar.	1

v2.14.1.	
General: Fix a 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 a bug with <code>fancyhdr</code> package 3.8 and later.	1
v2.15.3.	
General: Fix a 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>	278
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1
Add <code>\Xpagelinesep</code> hook.	42
Compatibility with new features of <code>reledpar</code>	1
Deleted dead code.	1
Display a warning message if using a version of \TeX that is too old.	1
Fix a bug with <code>\Xgroupbylines</code> for notes in two columns	1
Fix a bug with <code>\Xtxtbeforenotes</code> for notes in three or two columns	1
Fix a bug with ‘notenumber’ option of <code>indextools</code> package when indexing texts in familiar footnotes.	1
Fix potential bug when using <code>\edindex</code> in critical footnotes.	1
More explicit error message in case the stanza indentation is not defined.	278
New options for <code>\fnpos</code> and <code>\mpfnpos</code> to set a customized order for familiar and critical footnotes.	1
When <code>\edindex</code> is called outside of a <code>\beginnumbering... \endnumbering</code> structure, it is automatically switched to <code>\index</code> , with a warning message.	1
When indexing texts in familiar footnotes with <code>\edtext</code> , refer to the line number where the footnote is called.	1
When indexing texts in sidenotes with <code>\edtext</code> , refer to the line number where the sidenote is called.	1
v2.16.1.	
General: Fix a 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 a bug introduced by v2.16.1 when using non-expandable control sequence, like <code>\normalfont</code> , in the footnote number style.	1
v2.16.3.	
General: Fix a bug with <code>\Seref</code> (bug added in v2.7.0).	1
v2.16.4.	
General: Fix a 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 a bug with the line number style in <code>\doenotes</code> when referring to right side line in parallel typesetting.	1
Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code>	1
v2.16.7.	
General: Fix a bug with <code>\msdata</code> when using multiple <code>\beginnumbering ... \endnumbering</code>	1
Fix a bug with <code>\numberpstarttrue</code> when using multiple <code>\beginnumbering ... \endnumbering</code>	1
v2.16.8.	
General: Fix a bug with <code>\edindex</code> in footnotes in parallel typesetting.	1
v2.17.0.	
General: Add <code>\edglsadd</code> command.	1
Add <code>\setmsdataposition</code> setting.	1
v2.17.1.	
General: Fix spurious space in paragraphed footnotes when using <code>Lua\TeX</code> without using Right-To-Left text.	1
v2.17.2.	
General: Change log message when numbered files still don't exist, in order to improve compatibility with <code>latexmk</code>	1
v2.17.3.	
General: Fix a bug with <code>\doendnotesbysection</code> and <code>\doendnotes</code>	1
v2.17.4.	
General: Fix a bug with <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code>	1
v2.17.5.	
General: Fix a bug with <code>\pstartref</code> when referring to the left side in parallel typesetting.	1
v2.18.0.	
General: Fix a bug when using a <code>\edtext</code> in two lines or more in right-to-left typesetting with <code>X\TeX</code>	1
Fix a bug when using both <code>\Xnumberonlyfirstintwolines</code> or <code>\Xnumberonlyfirstinline</code> and <code>\Xparafootsep</code> and <code>\Xsymlinenum</code>	1
v2.18.1.	
General: Fix a bug when using <code>\msdata</code> with <code>Lua\TeX</code> or with the <code>hyperref</code> package.	1
v2.19.0.	
General: Add <code>\footnoteXmark</code> and <code>\footnoteXtext</code> commands.	1
Add better compatibility with the <code>csquotes</code> package when using familiar footnotes.	1
Fix a bug with paragraph indent after sectioning command.	1
v2.20.0.	
General: Add <code>\AtStartEveryStanza</code> , <code>\BeforeEveryStopStanza</code> , <code>\AtEndEveryPend</code> , <code>\AtStartEveryPstart</code>	1
Add second optional argument to <code>\pstart</code> , <code>\pend</code> and <code>\stanza</code>	1
Add starred version of <code>\AtEveryPstart</code> , <code>\AtEveryPend</code> , <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code>	1
Add third and fourth optional argument to <code>\newverse</code>	1
Fix a bug when using familiar footnotes in <code>\eledsection</code> and related.	1
Reset font specification at the beginning of familiar footnotes.	1

v2.21.0.	
General: Add the possibility of nested <code>\sameword</code> commands.	1
Fix a bug when using formatting command in the argument of <code>\edindex</code> inside <code>\edtext</code>	1
Now, as explained in the handbook, a <code>\edindex</code> inside <code>\edtext</code> only creates an index reference to the main text, and not to the critical footnote.	1
v2.22.0.	
General: Add <code>\txtbeforenotesonlyonceX</code> and <code>\Xtxtbeforenotesonlyonce</code> hooks. . .	1
Add <code>\txtbeforenotesX</code> hook.	1
Fix a bug added in v2.16.0 when using <code>\Xtxtbeforenotes</code> with paragraphed or normal footnotes.	1
Fix a bug with three and two columns critical footnotes, broken in v. 2.17.6.	1
v2.22.1.	
General: Compatibility with new version of <code>reledpar</code>	1
Fix a bug with some commands inside <code>\sameword</code>	1
v2.22.2.	
General: Do not print footnotes at the first run.	1
v2.23.0.	
General: Add <code>swcaseinsensitive</code> option	1
Optimisation of the code added on v2.22.2 to not print footnotes at the first run.	1
v2.24.0.	
<code>\next@line@list@stuff</code> : Add <code>\next@line@list@stuff</code> internal hook.	125
General: Add <code>\swnoexpands</code> macro to avoid problems with not fully expandable macro inside <code>\sameword</code>	1
Fix a bug with <code>\edlabel</code> at the beginning of a <code>\pstart</code>	1
Fix spurious space with <code>\labelpstarttrue</code>	1
When a <code>\setlinenum</code> is used, it is stronger than the <code>\lineation{page}</code> setting.	1
v2.24.1.	
General: Compatibility with <code>reledpar 2.20.2</code>	1
v2.24.2.	
General: Fix bug between tabular environments and endnotes	1
v2.25.0.	
General: Add <code>\Xendtxtbeforenotes</code> hook	1
v2.26.0.	
General: Add a mechanism for annotation of stanza line numbering	1
Fix bug in <code>\xpstartref</code>	1
v2.26.1.	
General: Fix bug with <code>\linenumberannotation</code>	1
v2.26.2.	
General: Fix bug with <code>\linenumberannotation</code> in parallel typesetting	1
Fix bug with <code>\Xgroupbyline</code> and <code>\Xparafootsep</code>	1
Fix bug with <code>\Xgroupbyline</code> and optional argument of critical footnotes.	1
Fix bug with <code>\Xgroupbyline</code> for the series after the first one.	1
v2.26.3.	
General: Fix bug added by v2.26.0 with notes in parallel typesetting	1
v2.26.4.	
General: Fix bug when using <code>\numberpstarttrue</code> and alternating between manual <code>\pstart</code> and automatic <code>\pstart</code> produced by <code>\autopar</code>	1
v2.26.5.	
General: Check that users have a enough recent version of <code>xparse</code>	1

Fix bug when using <code>\eledsection</code> with the <code>continuousnumberingwithcolumns</code> of <code>reledpar</code>	1
Fix bug with <code>continuousnumberingwithcolumns</code>	1
Fix bug with using <code>\eledsection</code> and related when alternate normal typesetting and parallel typesetting	1
v2.26.6.	
General: Fix bug with <code>\lineation</code> by page and nexted <code>\edtext</code> between two pages (added in v. 2.24.0)	1
v2.26.7.	
General: Fix page number in footnote and endnotes when using <code>sameparallelpagenumbers</code> option of <code>reledpar</code> and typesetting critical text in single mode after critical text in double page mode	1