

reledmac

Typeset scholarly editions with L^AT_EX*

Maïeul Rouquette[†]

based on the original ledmac by

Peter Wilson

Herries Press

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

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

Abstract

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

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

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

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

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

Contents

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

*This file (**reledmac.dtx**) has version number v2.16.8, last revised 2016/12/16.

[†]maieul at maieul dot net

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

6.3.2 Notes about input encoding with UTF-8 processor	28
6.3.3 Use with <code>\lemma</code> command	28
6.3.4 Customizing	30
6.4 Apparatus of Manuscripts	30
6.4.1 Marking sections of text	31
6.4.2 Layout of the apparatus of manuscripts	31
6.4.3 Settings	32
6.5 Familiar notes	32
6.5.1 Basic use	32
6.5.2 Customizing mark	32
6.5.3 Separator for multiple footnotes	32
6.6 Changing series	33
6.6.1 Create a new series	33
6.6.2 Delete series	33
6.6.3 Series order	33
6.7 Position of critical and familiar footnotes	33
7 Critical apparatus appearance	34
7.1 Notes arrangement in a series	34
7.2 Control line number printing	35
7.2.1 Print line number only at first time	35
7.2.2 Print page number only at first time	36
7.2.3 Arbitrary text before line number	36
7.2.4 Separator for line range	36
7.2.5 Abbreviate line range	37
7.2.6 Disable line number	37
7.2.7 Printing pstart number	37
7.2.8 Printing stanza number	38
7.2.9 Separator between line and subline numbers	38
7.2.10 Separator between page and line numbers	38
7.2.11 Space around number	38
7.2.12 Space around line symbol	39
7.2.13 Space in place of number	39
7.2.14 Boxing line number and line symbol	39
7.3 For endnotes	40
7.4 Arbitrary code around line number	40
7.5 Separator between the lemma and the note	40
7.5.1 For footnotes	40
7.5.2 For endnotes	41
7.6 Font style	41
7.6.1 For line number	41
7.6.2 For the lemma	41
7.6.3 For all notes	42
7.7 Wrapping notes	42
7.7.1 Wrapping lemmas	42
7.7.2 Wrapping contents	42

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

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

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

VII.2.5 Sidenotes and marginal line number initialization	143
VIII Line and page number computation	144
IX Line number printing	147
X Pstart number printing in side	151
XI Restoring footnotes and penalties	152
XI.1 Add insertions to the vertical list	152
XI.2 Penalties	154
XI.3 Printing leftover notes	155
XI.4 Text before notes	156
XII Critical footnotes	157
XII.1 Fonts	157
XII.2 Individual note options	157
XII.3 Notes language	158
XII.4 General survey of the way we manage notes	159
XII.5 General setup	159
XII.6 Footnotes arrangement	160
XII.6.1 User level macro	160
XII.6.2 Normal footnote	160
XII.6.3 Paragraphed footnotes	165
XII.6.4 Columnar footnotes	172
XII.7 Critical notes presentation	179
XII.7.1 Font tools	179
XII.7.2 Pstart number in footnote	180
XII.7.3 Lemma printing	181
XII.7.4 Line number printing	181
XII.7.5 Footnote grouped by line	190
XIII Familiar footnotes	191
XIII.1 Adjacent footnotes	191
XIII.2 Regular footnotes for numbered texts	193
XIII.3 Footnote formats	195
XIII.4 Footnote arrangement	195
XIII.4.1 User level macro	195
XIII.4.2 Normal footnotes	196
XIII.4.3 Two columns footnotes	201
XIII.4.4 Three columns footnotes	204
XIII.4.5 Paragraphed footnotes	206
XIII.5 Wrapping footnote marks in hyperlink	210
XIV Code common to both critical and familiar footnote in normal arrangement	211

XV Footnotes' width for two columns	211
XVI Footnotes' order	213
XVII Footnotes' rule	213
XVIII Specific skip for first series of footnotes	214
XVIII.0.1 Overview	214
XVIII.0.2 User level command	214
XVIII.0.3 Internal commands	215
XIX Endnotes	216
XIX.1 Internal commands	216
XIX.2 User level commands	220
XIX.2.1 Inserting contents to endnotes	220
XIX.2.2 Printing endnotes	220
XX Generate series of notes	227
XX.1 Test if series is still existing	227
XX.2 Init specific to reledpar	227
XX.3 For critical footnotes	228
XX.3.1 Options	228
XX.3.2 Create inserts, needed to add notes in foot	229
XX.3.3 Create commands for critical apparatus, \Afootnote, \Bfootnote etc.	229
XX.3.4 Set standard display	232
XX.4 For familiar footnotes	232
XX.4.1 Options	233
XX.4.2 Create tools for familiar footnotes (\footnoteX)	233
XX.5 The endnotes	235
XX.5.1 The auxiliary file	235
XX.5.2 The main macro	235
XX.5.3 Tools	236
XX.5.4 Internal commands	236
XX.5.5 The options	237
XX.6 Init standards series (A,B,C,D,E)	238
XXI Setting series display	238
XXI.1 Change series order	238
XXI.2 Test series order	239
XXI.2.1 Get the first series	239
XXI.3 Series setting	239
XXI.3.1 General way of working	239
XXI.3.2 Tools to set options	240
XXI.3.3 Tools to generate options commands	241
XXI.3.4 Options for critical notes	243
XXI.3.5 Options for familiar notes	244

XXI.3.6 Options for endnotes	245
XXI.4 Hooks for a particular footnote	247
XXI.5 Alias	248
XXII Output routine	248
XXII.1 Extra footnotes output	248
XXII.2 Patching standard output's commands	251
XXIII Cross referencing	254
XXIII.1 Compatibility with xref	267
XXIV Side notes	268
XXV Minipages and such	276
XXVI Indexing	281
XXVI.1 Looking on package order	281
XXVI.2 Auxiliary macros for \edindex	281
XXVI.3 Code specific to \edindexin critical footnotes	282
XXVI.4 Analysis of command in indexed text	284
XXVI.5 Code for the formatted index	284
XXVI.6 Main code	285
XXVI.7 Hyperlink	286
XXVI.8 'innote' and 'notenumber' option of indextols package	289
XXVII Glossaries	289
XXVIII Verse	291
XXVIII.1 Hanging symbol management	291
XXVIII.2 Using & character	292
XXVIII.3 Code category setting	292
XXVIII.4 Stanza count and indent	292
XXVIII.5 Numbering stanza	294
XXVIII.6 Stanza number in note	295
XXVIII.7 Main work	295
XXVIII.8 Restore catcode and penalties	298
XXIX Apparatus of Manuscripts	298
XXIX.1 User level macro	298
XXIX.2 Setting macro	299
XXIX.3 Counters and lists	300
XXIX.4 Auxiliary file macros	300
XXIX.5 Action macro	301
XXIX.6 Inserting footnote	305
XXIX.7 Other	306

XXX Arrays and tables	306
XXX.1 Preamble: macro as environment	306
XXX.2 Tabular environments	309
XXX.2.1 Disabling and restoring commands	310
XXX.2.2 Counters, boxes and lengths	313
XXX.2.3 Tabular typesetting	317
XXX.2.4 Environments	328
XXXI Quotation's commands	328
XXXII Section's title commands	329
XXXII.1 Commands to disable some feature	329
XXXII.2 General overview	330
XXXII.3 \beforeeledchapter command	330
XXXII.4 Auxiliary commands	331
XXXII.5 Patching standard commands	332
XXXII.6 Main code of \eledxxx commands	337
XXXII.7 Macros written in the auxiliary file	339
XXXIII Page breaking or no page breaking depending of specific lines	341
XXXIV Long verse: prevents being separated by a page break	343
XXXV Tools for hyperref package	344
XXXVI Compatibility with eledmac	344
Appendix A Things to do when changing versions	347
Appendix A.1 Migrating from edmac to ledmac	347
Appendix A.2 Migration from ledmac to eledmac	348
Appendix A.3 Migration to eledmac 1.5.1	349
Appendix A.4 Migration to eledmac 1.12.0	349
Appendix A.5 Migration to eledmac 17.1	350
Appendix A.6 Migration to eledmac 1.21.0	350
Appendix A.6.1 \Xledsetnormalparstuffand\ledsetnormalparstuffX	350
Appendix A.6.2 Endnotes	350
Appendix A.7 Migration to eledmac 1.22.0	350
Appendix A.8 Migration to eledmac 1.23.0	350
Appendix A.9 Migration from eledmac to reledmac	351
Appendix A.9.1 Risk of 'no room for a new'	351
Appendix A.9.2 Multiple indices with memoir	351
Appendix A.9.3 Deprecated commands and options	351
Appendix A.9.4 \renewcommandreplaced by command	352
Appendix A.9.5 Commands the names of which have been changed	352
Appendix A.9.6 Endnotes	354
Appendix A.9.7 Z Series	354
Appendix A.9.8 Internal commands	354

Appendix A.10 Migration to <code>reledmac</code> 2.1.0	354
Appendix A.11 Migration to <code>reledmac</code> 2.1.3	354
Appendix A.12 Migration to <code>reledmac</code> 2.3.0	354
Appendix A.13 Migration to <code>reledmac</code> 2.4.0	355
Appendix A.14 Migration to <code>reledmac</code> 2.5.0	355
Appendix A.15 Migration to <code>reledmac</code> 2.7.0	355
Appendix A.16 Migration to <code>reledmac</code> 2.7.2	355
Appendix A.17 Migration to <code>reledmac</code> 2.8.0	355
Appendix A.18 Migration to <code>reledmac</code> 2.13.1	355
References	357
Index	357
Change History	404

1 Introduction

1.1 Aim of the package

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

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

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

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

Any suggestions for new features are welcome.

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

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

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

1.2 History

1.2.1 `edmac`

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

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

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

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

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

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

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

1.2.2 ledmac

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

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

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

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

1.2.3 eledmac

Important changes were put in version 1.0, to make `ledmac` more easily extensible (see 7 p. 34). These changes can trigger small problems with the old customization. That is why a new name was selected: `eledmac` (extended `ledmac`).

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

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

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

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

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

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

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

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

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

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

¹²Being prepared at Poona and Lausanne Universities.

To migrate from `ledmac` to `eledmac`, please read Appendix A.2 p. 348.

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. 351.

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

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

2 How the package works

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

3 Compatibility warning

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

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

Please read 7.13.6 p. 46.

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

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

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

4 Options

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

4.1 Specific features

draft underlines lemmas in the main text.

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

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

eledmac-compat help to migrate from `eledmac` to `reledmac` (see Appendix A.9.5 p. 352).

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

nopbinverse prevents page break within verse environment.

noquotation by default, the quotation environment is redefined within numbered text. You can disable this redefinition with `noquotation` (see 17 p. 65).

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

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

xindy and `xindy+hyperref` select `xindy` as the index processor (13.4 p. 59).

4.2 Optimizing package performance

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

noeledsec disables tools for `\eledsection` and related commands (16.2 p. 64).

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

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

noledgroup `reledmac` allows use of a series of critical notes and a new series of normal notes inside `minipage` and `ledgroup` environments (see 10 p. 51). However, such features use up computer memory, at the expense of other processing needs. So if you do not need this feature, use `noledgroup` option. This should make `reledmac` faster.

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

5 Text lines and paragraphs numbering

5.1 Text lines numbering

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

```
\beginnumbering
Text
\endnumbering
```

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

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

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

5.2 Paragraphs

5.2.1 Basics

`\pstart` Within a numbered section, each paragraph of numbered text must be marked using the `\pend`

`\pstart` and `\pend` commands like this:

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

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

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

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

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

The lines of this paragraph are
not numbered.

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

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

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

```
\begingroup
\beginnumbering
\autopar

A paragraph of numbered text.

Another paragraph of numbered
text.

\endnumbering
\endgroup
```

`\autopar` fails, however, on paragraphs that start with a `{` or with any other command that starts a new group before it generates any text. Such paragraphs need to

be started explicitly, before the new group is opened, using `\indent`, `\noindent`, or `\leavevmode`, or using `\pstart` itself.¹³

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

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

This feature is also useful when typesetting verses (see 9 p. 48) or `reledpar` (see 19.1.8 p. 69).

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

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

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

5.2.5 Numbering paragraphs (`\pstart`)

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

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

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

`\sidepstartnumtrue`
`\labelpstarttrue`

On each `\beginnumbering` the numbering restarts.

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

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

5.2.6 Languages written in Right to Left

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

5.2.7 Memory limits

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

`\pausenumbering`
`\resumenumbering`

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

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

memory out, so to speak. If your text has a very long numbered section it is possible that your \LaTeX may reach its memory limit. There are two solutions to this.

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

The second solution is to split your long section into several smaller ones. The trouble with this is that your line numbering will start again at zero with each new section. To avoid this problem, we provide `\pausenumbering` and `\resumenumbers` 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

\resumenumbers
\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\resumenumbers}
```

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

5.3 Lineation commands

5.3.1 Disabling lineation

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

5.3.2 Setting lineation start and step

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

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

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

`\sublinenumincrement` You can define `\linenumberlist` to specify a non-uniform distribution of printed

`\linenumberlist`

line numbers. For example:

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

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

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

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

5.3.3 Setting lineation reset

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

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

5.3.4 Setting line number margin

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

```
\linenummargin{left}
```

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

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

5.3.5 Other settings

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

5.4 Changing the line numbers

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

5.4.1 Sublineation

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

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

You can change the separator between line number and subline number or using `\Xsublinesep` without any option argument (7.2.9 p. 38 or using `\Xsublinesepside`. But in the second case, it will change the separator only for line number in side, not for the footnotes.

5.4.2 Locking lineation

`\startlock` `\endlock` The `\startlock` command, used in running text, locks the line number at its current value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 48.

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

5.4.3 Setting and changing line number

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

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

5.4.4 Line number style

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

Alph Uppercase letters (A ... Z).

alph Lowercase letters (a ... z).

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

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

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

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

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

5.4.5 Skipping and hiding number

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

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

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

5.5 Executing code at each line

`\dolinehook` `\doinsidelinehook` `reledmac` provides an advanced feature for users. The argument passed to `\dolinehook{⟨arg⟩}` will be executed before slicing a new line in the paragraph. The argument passed to `\doinsidelinehook{⟨arg⟩}` will be executed before printing a new line, when the line number 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/>
		1 Smith] Jones C, D.

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

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

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

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

6.2.2 Footnotes

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

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

If you need more series of critical notes, please look at 6.6.1 p. 33.

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

- `fulllines` to disable `\Xtwolines` and `\Xmorethantwolines` features for this note (cf. 7.2.5 p. 37).
- `nonum` disables line numbering for this note. A horizontal blank space is added instead. You can use `\Xinplaceoflemmaseparator` to set it (7.5.1 p. 41).
- `nosep` to disable the lemma separator for this note.
- `linangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

6.2.3 Endnotes

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

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

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

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

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

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 7.2.5 p. 37).
- `nonum` to disable line number for this particular note.
- `nosep` to disable the lemma separator for this particular note. A horizontal blank space is added instead. You can use `\Xendinplaceoflemmaseparator` to set it (7.5.2 p. 41).

- `linangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

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

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

Note that by default inside endnotes no separator is used between the lemma and the content. However you can use the `\Xendlemmaseparator` macro to define one (7.5.2 p. 41).

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

`toendnotes` **Code between endnotes** Sometimes, it is useful to insert content between endnotes of the same series: for example to separate endnotes of different sections of the same text. In this case, you could use *inside numbered text* the command:
`Xtoendnotes` `\toendnotes[<series>]{<content>}` where `<series>` is a comma-separated list of the series of endnotes where `<content>` must be inserted. If `<series>` is empty, then `<content>` is inserted to all the series.

For example:

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

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

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

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

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

6.2.4 Paragraph in critical apparatus

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

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

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

6.2.5 Change lemma and line number

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

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

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

```
\linenum{ $\langle start page \rangle$  |  $\langle s. line \rangle$  |  $\langle s. sub-l. \rangle$  |  $\langle end p. \rangle$  |  $\langle e. l. \rangle$  |  $\langle e. sub-l. \rangle$  |  $\langle font \rangle$  | }
```

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

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

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

nested, for instance, you can use `\lemma` and `\linenum` to generate such notes despite the limitations of `\edtext`. If the $\langle lemma \rangle$ argument to `\edtext` is extremely long, you may run out of memory; here again you can specify a note with an abbreviated lemma using `\lemma` and `\linenum`. The numbers used in `\linenum` need not be entered manually; you can use the ‘x-’ symbolic cross-referencing commands below (11 p. 52) to compute them automatically.

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

6.2.6 Changing the names of commands for critical apparatus

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

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

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

6.3 Disambiguation of identical words in the apparatus

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

6.3.1 Basic use

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

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

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

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

6.3.2 Notes about input encoding with UTF-8 processor

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

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

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

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

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

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

With these tools, $\text{Xe}\text{\LaTeX}$ / $\text{Lua}\text{\LaTeX}$ will dynamicaly normalize unicode input when reading the file. Consequently, you will have no problems with the `\sameword` command.

6.3.3 Use with `\lemma` command

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

For example in the following example:

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

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

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

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

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

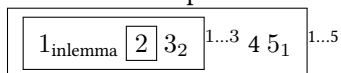
Note that only words that are actually referenced in a `\lemma` need the optional argument. Therefore, the first `\sameword` in the example above should have “1” as its optional argument, to be referenced correctly in the lemma.

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

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

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

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



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

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

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

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

Here, the corresponding code:

```

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

```

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

6.3.4 Customizing

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

```

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

```

6.4 Apparatus of Manuscripts

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

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

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

6.4.1 Marking sections of text

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

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

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

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

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

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

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

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

6.4.2 Layout of the apparatus of manuscripts

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

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

6.4.3 Settings

As the apparatus of manuscripts technically consists of first-level critical notes ('A' series), any setting available for critical notes can be applied (7 p. 34). However, the following *additional* commands are available.

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

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

6.5 Familiar notes

6.5.1 Basic use

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

`\footnoteB`

`\footnoteC`

`\footnoteD`

`\footnoteE`

6.5.2 Customizing mark

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

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

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

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

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

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

There are similar command triples for the other series.

6.5.3 Separator for multiple footnotes

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

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

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

and can be changed if necessary.

6.6 Changing series

6.6.1 Create a new series

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

6.6.2 Delete series

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

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

6.6.3 Series order

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

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

6.7 Position of critical and familiar footnotes

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

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

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

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

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

`<series><type>`

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

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

```

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

```

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

7 Critical apparatus appearance

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

When a length, noted $\langle l \rangle$, is used, it can be stretchable: a plus b minus c . The final length m is calculated by \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.

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

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

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

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

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

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

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

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

7.2 Control line number printing

7.2.1 Print line number only at first time

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

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

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

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

7.2.2 Print page number only at first time

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

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

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

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

In any case, you can use:

`\Xendsympagenum`

- `\Xendsympagenum[⟨series⟩]{⟨c⟩}` to print `⟨c⟩` when the page number is not printed.

`\Xendinplaceofpagenumber`

- `\Xendinplaceofpagenumber[⟨series⟩]{⟨l⟩}` to print a `⟨l⟩` length horizontal space in case no symbol is printed instead of the page number.

7.2.3 Arbitrary text before line number

`\Xbeforenumber`

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

7.2.4 Separator for line range

`\Xlinerangeseparator`

`\Xendlinerangeseparator`

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

7.2.5 Abbreviate line range

`\Xtwolines`
`\Xmorethantwolines`

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

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

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

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

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

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

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

`\Xtwolinesonlyinsamepage`

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

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

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

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

7.2.6 Disable line number

`\Xnonumber`
`\Xendnonumber`

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

7.2.7 Printing pstart number

`\Xpstart`

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

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

`\Xpstarteverytime`

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

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

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

7.2.8 Printing stanza number

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

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

`\Xstanzaseparator`

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

7.2.9 Separator between line and subline numbers

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

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

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

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

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

7.2.10 Separator between page and line numbers

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

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

7.2.11 Space around number

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

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

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

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

7.2.12 Space around line symbol

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

7.2.13 Space in place of number

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

7.2.14 Boxing line number and line symbol

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

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

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

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

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

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

7.3 For endnotes

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

7.4 Arbitrary code around line number

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

7.5 Separator between the lemma and the note

7.5.1 For footnotes

<code>\Xlemmaseparator</code> <code>\Xbeforelemmaseparator</code>	By default, in a footnote, the separator between the lemma and the note is a right bracket (<code>\rbracket</code>) ¹⁸ . You can use <code>\Xlemmaseparator[⟨s⟩]{⟨Xlemmaseparator⟩}</code> to change it. The optional argument can be used to specify the series in which it is used. Note that there is a non-breakable space between the lemma and the separator, but a breakable space between the separator and the following text. Using <code>\Xbeforelemmaseparator[⟨s⟩]{⟨l⟩}</code> you can add some space between
--	---

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

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

`\Xafterlemmaseparator`

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

`\Xnolemmaseparator`

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

`\Xinplaceoflemmaseparator`

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

7.5.2 For endnotes

`\Xendlemmaseparator`

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

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

`\Xendbeforelemmaseparator`

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

`\Xendafterlemmaseparator`

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

`\Xinplaceoflemmaseparator`

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

7.6 Font style

7.6.1 For line number

`\Xnotenumfont`

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

`\Xendnotenumfont`

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

`\notenumfontX`

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

7.6.2 For the lemma

`\lemmadisablefontselection`

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

`\Xendlemmadisablefontselection`

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

`\Xlemmafont`
`\Xendlemmafont`

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

`\Xlemmafont{\bfseries}`

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

7.6.3 For all notes

`\Xnotefontsize`

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

`\notefontsizeX`

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

`\Xendnotefontsize`

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

7.7 Wrapping notes

7.7.1 Wrapping lemmas

`\Xwraplemma`

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

`\Xwrapendlemma`

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

7.7.2 Wrapping contents

`\Xwrapcontent`

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

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

`\Xendwrapcontent`

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

`\wrapcontentX`

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

7.8 Indent of notes content

`\Xparindent`

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

`\parindentX`

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

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

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

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

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

7.9 Arbitrary code at the beginning of notes

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

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

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

7.10 Arbitrary code before inserting note

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

They allow one to add any arbitrary code just before the footnotes are added in the list of footnotes. The main use is to insert text direction code. For example, if you edit right-to-left text with `bidi` , but want your critical footnote be left-to-right, use `\Xbeforeinserting\LTR`. You should also use `\Xwraplemma` to ensure your lemmas are right-to-left in a left-to-right paragraph (7.7.1 p. 42)).

Note that the changes are local to the footnote.

7.11 Options for footnotes in columns

7.11.1 Alignment

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

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

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

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

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

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

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

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

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

7.11.2 Size of the columns

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

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

7.12 Options for paragraphed footnotes and notes grouped by line

7.12.1 Mark separation of notes

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

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

7.12.2 Ragged text

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

7.13 Options for block of notes

7.13.1 Grouping notes by line

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

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

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

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

7.13.2 Text before notes

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

7.13.3 Code before notes

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

7.13.4 Spacing

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

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

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

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

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

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

7.13.5 Rule

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

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

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

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

7.13.6 Maximum height

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

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

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

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

7.13.7 Width

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

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

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

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

7.14 Footnotes and the `reledpar` columns

`\Xnoteswidthliketwocolumns`
`\noteswidthliketwocolumnsX`

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

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

7.15 Endnotes in one paragraph

`\Xendparagraph`

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

`\Xendafternote`

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

`\Xendsep`

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

8 Fonts

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

`\numlabfont`

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

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

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

`\select@lemmafont`

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

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

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

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

9 Verse

9.1 Basic

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

9.2 Define stanza indents

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

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

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

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

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

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

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

9.3 Repeating stanza indents

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

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

is like

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


Be careful: the feature is changed in eledmac 1.5.1. See Appendix A.3 p. 349.

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

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

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

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

9.4 Manual stanza indent

`\stanzaindent`
`\stanzaindent*`

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

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

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

9.5 Stanza breaking

`\setstanzapenalties`

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

The command

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

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

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

9.6 Hanging symbol

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

```
\sethangingsymbol
```

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

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

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

9.7 Long verse and page break

If you want to prevent page breaks inside long verses, use the option `nopbinverse` when loading package, or use `\lednopbinversetrue`. Read 18.2 p. 65 for further details.

9.8 Content before/after verses

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

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

9.9 Numbering stanza

```
\numberstanzatrue
\numberstanzafalse
```

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

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

You can use this feature in combination with `\Xstanza` (7.2.8 p. 38).

```
thestanza
```

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

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

}

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

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

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

9.10 Various tools

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

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

9.11 Notes on empty lines

Since v2.3.0 of `reledmac`, empty lines when typesetting verses no longer produce new paragraphs, and consequently, do not insert vertical spaces. Use optional argument of `\stanza` or `\newverse` to insert vertical space (9.8 p. 50).

10 Grouping

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

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

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

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

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

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

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

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

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

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

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

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

11 Cross referencing

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

11.1 Basic use

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

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

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

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

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

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

11.2 Cross-referencing to a critical note

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

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

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

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

11.3 Cross-referencing which return a number in any case

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

Where #1 stands for the reference.

However, there are situations in which you will want `reledmac` to return a number without displaying any warning messages about undefined labels or the like: if you want to use the reference in a context where \TeX is looking for a number, such a warning will lead to a complaint that the number is missing. This is the case for references used within the argument to `\linenum`, for example (see 6.2.5 p. 26).

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

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

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

`\xxref`

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

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

It automatically calls `\linenum` (q.v., 6.2.5 p. 26 above) and sets the beginning page, line and subline numbers to those of the place where `\edlabel{mouse}` was placed, and the ending numbers to those where `\edlabel{elephant}` occurs.

11.4 Not automatic cross-referencing

`\edmakelabel`

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

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

11.5 Normal \LaTeX cross-referencing

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

`\ref`

`\pageref`

11.6 References to start and end lines

11.6.1 Reference to main text lines

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

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

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

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

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

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

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

11.6.2 References to lines that are commented on in the apparatus

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

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

 If you use `\linenum` (6.2.5 p. 26) to refer to these labels, `reledmac` will use your line settings to refer to the passage.

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

`\apprefwithpage`

11.6.3 Settings

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

`\setapprefprefixmore`

For example, you may use:

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

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

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

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

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

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

Also note that `\setSerefonlypageprefixsingle` is only a shortcut for `\XendbeforepagenumberSerefonly` (see 11.6.3 p. 55). So if you use `\Xendbeforepagenumber` without any optional argument, it will override this setting.

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

These commands are the following:

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

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

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

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

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

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

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

11.7 Compatibility with `xr` package

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

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

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

12 Side notes

12.1 Basics

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

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

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

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

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

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

12.2 Setting

12.2.1 Width

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

12.2.2 Vertical position

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

12.2.3 Distance to the main text

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

12.2.4 Font

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

These can of course be changed to suit.

12.2.5 Separator between notes

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

13 Indexing

13.1 Basics

`\edindex` \LaTeX provides the `\index{<item>}` command for specifying that `<item>` and the current

page number should be added to the raw index (`idx`) file. The `\edindex{⟨item⟩}` macro can be used in numbered text to specify that `⟨item⟩` and the current page & linenumber should be added to the raw index file.

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

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

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

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

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

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

13.2 Referring to critical notes

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

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

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

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

13.3 Separator between page and line numbers

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

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

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

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

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

13.4 Using xindy

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

`pagenumber-linenumber`

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

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

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

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

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

13.5 Advanced setting

`\edindexlab`

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

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

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

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

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

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

14 Glossary

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

14.1 Preamble setting

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

- Or set `.glossaries`:
`\glsSetCompositor{-}`
- Or set `reledmac`:
`\renewcommand{\pagelinesep}{-}`

In this case, the above will have consequences for your use of `\edindex` and you should set your `.ist` file (13.3 p. 58).

14.2 Commands

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

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

15 Tabular material

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

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

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

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

1	2	3
a	bb	ccc
AAA	BB	C

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

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

For example:

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

produces the following parallel pair of verses.

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

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

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

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

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

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

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

```
\begin{edtabularr}
```

```

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

```

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

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

```

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

```

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

```

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

```

1	2	3	4
a	┌	┐	d
A	B	C	D

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

```

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

```

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

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

For example:

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

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

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

`\edvertdots`

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

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

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

16 Sectioning commands

16.1 Sectioning commands without line numbers or critical notes

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

p. 18):

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

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

16.2 Sectioning commands with line numbering and critical notes

You have to use the following commands:

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

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

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

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

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

16.3 Optimization

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

17 Quotation environments

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

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

18 Page breaks

18.1 Control page breaking

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

`\ledpb`
`\lednopb`

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

These commands have effect only at the second run.

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

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

18.2 Prevent page break in a long verses

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

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

19 Miscellaneous

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

19.1 Known and suspected limitations

19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.13.6 p. 46 in order to prevent footnotes from overlapping the bottom margin.

19.1.2 floatrow package compatibility

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

19.1.3 ‘No room for a new’

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

In order to prevent such problems, the first thing is to use the options to optimize `reledmac`. For example, if you need only two series of notes, use the `series={A,B}` option. Read 16.3 p. 65 in order to know which are the available options.

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

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

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

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

19.1.4 Marginal notes

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

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

19.1.5 Paragraph shape

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

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

19.1.6 Paragraphed footnotes

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

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

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

`\footfudgefiddle`

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

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

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

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

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

19.1.7 Use with other packages

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

`\morenoexpands`

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

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

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

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

or

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

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

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

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

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

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

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

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

19.1.8 Parallel typesetting

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

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

I Implementation overview

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

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

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

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

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

II Preliminaries

II.1 Links with original `edmac`

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

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

II.2 Package declaration

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

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledmac}[2016/12/16 v2.16.8 typesetting critical editions]
4 %

```

II.3 Package options

```

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

```

```

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

```

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

```

78 \ProcessOptionsX*\relax
79
80 %

```


II.4 Loading packages

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

```

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

```

II.5 Compatibility with Lua_{TeX}

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

```

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

```

II.6 Boolean flags

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

```

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

```

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

```

106 \newif\if@ledgroup%
107 %

```

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

```

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

```

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

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

II.7 Messages

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

\reledmac@warning Write a warning message.

```

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

```

`\reledmac@error` Write an error message.

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

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

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

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

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

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

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

```

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

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

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

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

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

```

```

\led@warn@BadAction01 \newcommand*{\led@warn@BadAction}{%
202 \reledmac@warning{Bad action code, value \next@action.}}
203 %

```

```

\led@warn@DuplicateLabel204 \newcommand*{\led@warn@DuplicateLabel}[1]{%
ppLabelOutSecondArgEdtext05 \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined06 \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined07 }%
208 \newcommand*{\led@warn@AppLabelOutSecondArgEdtext}[1]{%
209 \reledmac@warning{\string\applabel\space outside of the second argument
of an \string\edtext\space `#1' on page \thepage.}}%
210 \newcommand*{\led@warn@RefUndefined}[1]{%
211 \G@refundefinedtrue%
212 \reledmac@warning{Reference `#1' on page \thepage\space undefined.%
213 Using `000'.}%
214 \@latex@warning{Reference `#1' undefined\on@line}%
215 }%
216 \newcommand*{\led@warn@pairRefUndefined}[1]{%
217 \G@refundefinedtrue%
218 \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \thepage\
space undefined.
219 Using `??'.}%
220 \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
221 }
222 %

```

```

\led@warn@NoMarginpars23 \newcommand*{\led@warn@NoMarginpars}{%
224 \reledmac@warning{You can't use \string\marginpar\space in numbered text
}}
225 %

```

```

\led@warn@BadSidenotemargin26 \newcommand*{\led@warn@BadSidenotemargin}{%
227 \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
228 %

```

```

\led@warn@NoIndexFile29 \newcommand*{\led@warn@NoIndexFile}[1]{%
230 \reledmac@warning{Undefined index file #1}}
231 %

```

```

\led@warn@SeriesStillExist32 \newcommand{\led@warn@SeriesStillExist}[1]{%
233 \reledmac@warning{Series #1 is still existing !}%
234 }%
235 %

```

```

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

```

```

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

```

```

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

```

```

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

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

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

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

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

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

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

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```


II.8 Gobbling

Here, we define some commands which gobble their arguments.

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

II.9 Miscellaneous commands

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

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

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

Initialize it as `\empty`.

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

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

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

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

II.10 Prepare reledpar

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

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

```

354 \newcount\l@dnumpstartsL
355 %

```

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

```

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

```

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

```

359 \newif\ifnumberingR
360 %

```

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

```

361 \newif\ifXnote@%
362 %

```

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

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

```

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

```

III Sectioning commands

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

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

```

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

```

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

```

370 \newif\ifnumbering
371 %

```

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

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

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

```

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

```

The tools for section's title commands are called:

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

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

```

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

```

`\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.

```

425 \def\endnumbering{%

```

```

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

```

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

```

458 \newcommand{\pausenumbering}{%
459   \ifautopar\global\autopar@pausetrue\fi%
460   \endnumbering\global\numberingtrue}
461 %

```

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.

```

462 \newcommand*{\resumenumbering}{%
463   \ifnumbering
464     \ifautopar@pause\autopar\fi

```

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

```

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

```

IV List macros

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

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

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

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

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

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

```

496 \newcommand*\list@create}[1]{%
497   \global\let#1=\empty%
498 }%
499 %

```

\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 .

```

500 \newcommand*\list@clear}[1]{%
501   \global\let#1=\empty%
502 }
503 %

```

\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`.

```

504 \newtoks\led@toksa \newtoks\led@toksb
505 \global\led@toksa={\}
506 \long\def\xright@appenditem#1\to#2{%
507   \global\led@toksb=\expandafter{#2}%
508   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
509   \global\led@toksb={}}
510 %

```

\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`.

```

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

```

\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.

```

516 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
517 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
518
519 %

```

V Line counting

V.1 Choosing the system of lineation

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

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

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

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

`\ifbypage@`

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

`\bypage@false`

```
520 \newif\ifbypage@
521 \newif\ifbypstart@
522 %
```

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@R23 \newif\ifbypage@R
\ifbypstart@R24 \newif\ifbypstart@R
525 %
```

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

```
526 \newcommand*{\lineation}[1]{%
527 %
```

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

```
528 \ifnumbering
529 \led@err@LineationInNumbered
530 \else
531 %
```

If the argument is page.

```
532 \def\@tempa{#1}\def\@tempb{page}%
533 \ifx\@tempa\@tempb
534 \global\bypage@true
535 \global\bypstart@false
536 \unless\ifnocritical@%
537 \Xpstart[] [false]%
538 \fi%
539 %
```

If the argument is pstart.


```

540 \else
541   \def\@tempb{pstart}%
542   \ifx\@tempa\@tempb
543     \global\bypage@false
544     \global\bystart@true
545     \unless\ifnocritical@%
546       \Xpstart%
547     \fi%
548 %

```

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

```

549 \else
550   \def\@tempb{section}
551   \ifx\@tempa\@tempb
552     \global\bypage@false
553     \global\bystart@false
554     \unless\ifnocritical@%
555       \Xpstart[] [false]%
556     \fi%
557 %

```

In other case, it is an error.

```

558 \else
559   \led@warn@BadLineation
560 \fi
561 \fi
562 \fi
563 \fi}}
564 %

```

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`

```

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

```

```

577 \fi
578 \fi}}
579
580 \newcommand*{\l@dgetline@margin}[1]{%
581 \def\@tempa{#1}\def\@tempb{left}%
582 \ifx\@tempa\@tempb
583 \l@dtempcntb \z@
584 \else
585 \def\@tempb{right}%
586 \ifx\@tempa\@tempb
587 \l@dtempcntb \@ne
588 \else
589 \def\@tempb{outer}%
590 \ifx\@tempa\@tempb
591 \l@dtempcntb \tw@
592 \else
593 \def\@tempb{inner}%
594 \ifx\@tempa\@tempb
595 \l@dtempcntb \thr@@
596 \else
597 \led@warn@BadLinenummargin
598 \l@dtempcntb \m@ne
599 \fi
600 \fi
601 \fi
602 \fi}
603
604 %

```

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.

```

605 \newcounter{firstlinenum}
606 \setcounter{firstlinenum}{5}
607 \newcounter{linenumincrement}
608 \setcounter{linenumincrement}{5}
609 %

```

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

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

```

610 \newcounter{firstsublinenum}
611 \setcounter{firstsublinenum}{5}
612 \newcounter{sublinenumincrement}

```

```

613 \setcounter{sublinenumincrement}{5}
614
615 %

```

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

```

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

```

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.

```

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

```

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

```

678 \newcount\sublock@disp
679 \newcommand{\sublockdisp}[1]{%
680   \l@getlock@disp{#1}%
681   \ifnum\@l@tempcntb>\m@ne
682     \global\sublock@disp=\@l@tempcntb
683   \else
684     \led@warn@BadSublockdisp
685   \fi}}
686
687 %

```

V.5 Line number style

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

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

```

688 \newcommand*{\linenumberstyle}[1]{%
689   \def\linenumrep##1{\@nameuse{@#1}{##1}}}
690 \newcommand*{\sublinenumberstyle}[1]{%
691   \def\sublinenumrep##1{\@nameuse{@#1}{##1}}}
692 %

```

Initialise the number styles to arabic.

```

693 \linenumberstyle{arabic}
694 \let\linenumr@p\linenumrep
695 \sublinenumberstyle{arabic}
696 \let\sublinenumr@p\sublinenumrep
697 %
698 %

```

V.6 Line number printing

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

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

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

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

```

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

```

```

713 \fi
714 \fi%
715 \egroup%
716 }%
717
718 \newcommand*{\leftlinenum}{%
719 \ledlinenum
720 \kern\linenumsep}
721 \newcommand*{\rightlinenum}{%
722 \kern\linenumsep
723 \ledlinenum}
724
725 %

```

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`.

```

726 \newcount\line@num
727 %

```

`\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.

```

728 \newcount\subline@num
729 %

```

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

`\sublines@true`

`\sublines@false` You may wonder why we do not just use the value of `\subline@num` to determine this—treating anything greater than 0 as an indication that sub-lineation is on. We need a separate flag because sub-lineation can be used together with line-number locking in

odd ways: several pieces of a logical line might be interrupted by pieces of sub-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.

```
730 \newif\ifsublines@
731 %
```

\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.

```
732 \newcount\absline@num
733 %
```

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.

```
734 \newcount\@lock
735 \newcount\sub@lock
736 %
```

V.9 Line number associated to lemma

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Here are the commands to create these lists:

```

737 \list@create{\line@list}
738 \list@create{\insertlines@list}
739 \list@create{\actionlines@list}
740 \list@create{\actions@list}
741
742 %

```

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

```

743 \newcount\page@num
744 \newcount\endpage@num
745 \newcount\endline@num
746 \newcount\endsubline@num
747 %

```

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

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

```
748 \newif\ifnoteschanged@
749 %
```

`\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@50 \newcommand*\resetprevline@{%
751   \def\do##1{\global\csundef{prevline##1}}%
752   \dolistloop{\@series}%
753 }
754 %
```

`\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@55 \newcommand*\resetprevpage@num{%
756   \def\do##1{%
757     \ifcsdef{prevpage##1@num}{%
758       \global\csname prevpage##1@num\endcsname=\z@%
759       \global\csname prevpage##1@numR\endcsname=\z@%
760     }%
761     {}%
762     \ifcsdef{##1prevpage@num}{%
763       \global\csname ##1prevpage@num\endcsname=\z@%
764       \global\csname ##1prevpage@numR\endcsname=\z@%
765     }%
766     {}%
767   }%
768   \dolistloop{\@series}%
769 }
770 %
```

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.

```

771 \newread\@inputcheck
772 \newcommand*{\read@linelist}[1]{%
773   \ifledRcol%
774     \list@clearing@regR%
775   \else%
776     \list@clearing@reg%
777   \fi%
778   %

```

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).

```

779   \list@clear{\maxlinesinpar@list}
780   %

```

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

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

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

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

```

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

```

```

795 \fi%
796 \endgroup
797 %

```

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.

```

798 \ifledRcol
799   \global\page@numR=\m@ne
800   \ifx\actionlines@listR\empty
801     \gdef\next@actionlineR{1000000}%
802   \else
803     \gl@p\actionlines@listR\to\next@actionlineR
804     \gl@p\actions@listR\to\next@actionR
805   \fi
806 \else
807   \global\page@num=\m@ne
808   \ifx\actionlines@list\empty
809     \gdef\next@actionline{1000000}%
810   \else
811     \gl@p\actionlines@list\to\next@actionline
812     \gl@p\actions@list\to\next@action
813   \fi
814 \fi
815 }
816 %

```

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

```

817 \newcommand*\list@clearing@reg{%
818   \list@clear{\line@list}%
819   \list@clear{\insertlines@list}%
820   \list@clear{\actionlines@list}%
821   \list@clear{\actions@list}%
822   \list@clear{\linesinpar@listL}%
823   \list@clear{\linesonpage@listL}%
824 }%
825 %

```

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

```

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

```

```

832 \led@warn@NoLineFile{\l@auxdir#1}%
833 \global\noteschanged@true
834 \begingroup}%
835 }
836
837 %

```

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

V.11 Commands within the line-list file

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

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

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

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

```

838 \newcommand{\line@list@version}[1]{%
839 \IfStrEq{#1}{\this@line@list@version}%
840 {}%
841 {\ifledRcol%
842 \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
843 \else%
844 \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
845 \fi%
846 \endinput%
847 }%
848 }%
849 %

```

`\@nl` `\@nl` does everything related to the start of a new line of numbered text.

`\@nl@reg` In order to get the `\setlinenum` to work Peter Wilson had to slip in some new code at the start of the macro, to get the timing of the actions correct. The problem was that his original naive implementation of `\setlinenum` had a unfortunate tendency to change the number of the last line of the *preceding* paragraph. The new code is sort of based on the page number handling and `\setline`. It seems that a lot of fiddling with the line number internals is required.

In November 2004 in order to accurately determine page numbers Peter Wilson added these to the macro. It is now:

`\@nl{<page counter number>}{<printed page number>}`

We do not (yet) use the printed number (i.e., the `\thepage`) but it may come in handy later. The macro `\fix@page` checks if a new page has started.

Exactly what `\@nl` does depends on whether right text is being processed. That's why many code is defined in `\@nl@reg` or `\nl@regR`.

```

850
851 \newcommand*{\@nl}[2]{%
852   \fix@page{#1}%
853   \ifledRcol%
854     \@nl@regR%
855   \else%
856     \@nl@reg%
857   \fi%
858 }
859 \newcommand*{\@nl@reg}{%
860   \ifx\l@dchset@num\relax \else
861     \advance\absline@num \@ne
862     \set@line@action
863     \let\l@dchset@num=\relax
864     \advance\absline@num \m@ne
865     \advance\line@num \m@ne
866   \fi
867 %

```

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

```

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

```

```

881 \fi
882 %

```

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

```

883 \ifcase\@lock
884 \or
885 \@lock \tw@
886 \or \or
887 \@lock \z@
888 \fi
889 \ifcase\sub@lock
890 \or
891 \sub@lock \tw@
892 \or \or
893 \sub@lock \z@
894 \fi
895 %

```

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

```

896 \ifsublines@
897 \ifnum\sub@lock<\tw@
898 \advance\subline@num \@ne
899 \fi
900 \else
901 \ifnum\@lock<\tw@
902 \advance\line@num \@ne \subline@num \z@
903 \fi
904 \fi}
905 %
906 %

```

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

```

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

```

```

922 \fi
923 \else
924   \ifnum #1=\last@page@num
925   \else
926     \csnumgdef{\lastabsline@forpage@the\page@num}{\the\absline@num}%
927     \csnumgdef{\lastline@forpage@the\page@num}{\the\line@num}%
928     \ifbypage@
929       \line@num \z@ \subline@num \z@
930     \fi
931     \page@num=#1\relax
932     \last@page@num=#1\relax
933     \def\next@page@num{#1}%
934     \listxadd{\normal@page@break}{\the\absline@num}
935   \fi
936 \fi}
937 %

```

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

```

938 \newcommand*{\@pend}[1]{}
939 \newcommand*{\@pendR}[1]{}
940 \newcommand*{\@lopL}[1]{}
941 \newcommand*{\@lopR}[1]{}
942
943 %

```

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

```

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

```

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

```

956
957 \newcommand*{\@adv}[1]{}

```



```

958 \ifsublines@
959   \ifledRcol
960     \advance\subline@numR by #1\relax
961     \ifnum\subline@numR<\z@
962       \led@warn@BadAdvancelineSubline
963       \subline@numR \z@
964     \fi
965   \else
966     \advance\subline@num by #1\relax
967     \ifnum\subline@num<\z@
968       \led@warn@BadAdvancelineSubline
969       \subline@num \z@
970     \fi
971   \fi
972 \else
973   \ifledRcol
974     \advance\line@numR by #1\relax
975     \ifnum\line@numR<\z@
976       \led@warn@BadAdvancelineLine
977       \line@numR \z@
978     \fi
979   \else
980     \advance\line@num by #1\relax
981     \ifnum\line@num<\z@
982       \led@warn@BadAdvancelineLine
983       \line@num \z@
984     \fi
985   \fi
986 \fi
987 \set@line@action}
988
989 %

```

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

```

990
991 \newcommand*{\@set}[1]{%
992   \ifledRcol
993     \ifsublines@
994       \subline@numR=#1\relax
995     \else
996       \line@numR=#1\relax
997     \fi
998   \set@line@action
999 \else
1000   \ifsublines@
1001     \subline@num=#1\relax
1002   \else
1003     \line@num=#1\relax

```

```

1004 \fi
1005 \set@line@action
1006 \fi}
1007
1008 %

```

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

\l@dchset@num `\l@dchset@num` is a flag to the `\@n1?` macro. If it is not `\relax` then a linenum change is to be done.

```

1009
1010 \newcommand*{\l@d@set}[1]{%
1011 \ifledRcol
1012 \line@numR=#1\relax
1013 \advance\line@numR \@ne
1014 \def\l@dchset@num{#1}
1015 \else
1016 \line@num=#1\relax
1017 \advance\line@num \@ne
1018 \def\l@dchset@num{#1}
1019 \fi}
1020 \let\l@dchset@num\relax
1021
1022 %

```

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

```

1023
1024 \newcommand*{\page@action}{%
1025 \ifledRcol
1026 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1027 \xright@appenditem{\next@page@numR}\to\actions@listR
1028 \else
1029 \xright@appenditem{\the\absline@num}\to\actionlines@list
1030 \xright@appenditem{\next@page@num}\to\actions@list
1031 \fi}
1032 %

```

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

```

1033
1034 \newcommand*{\set@line@action}{%
1035 \ifledRcol
1036 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1037 \ifsublines@
1038 \@l@dttempcnta=-\subline@numR
1039 \else
1040 \@l@dttempcnta=-\line@numR

```

```

1041 \fi
1042 \advance\@l@tempcnta by -5000\relax
1043 \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1044 \else
1045 \xright@appenditem{\the\absline@num}\to\actionlines@list
1046 \ifsublines@
1047 \@l@tempcnta=-\subline@num
1048 \else
1049 \@l@tempcnta=-\line@num
1050 \fi
1051 \advance\@l@tempcnta by -5000\relax
1052 \xright@appenditem{\the\@l@tempcnta}\to\actions@list
1053 \fi}
1054 %

```

\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.

```

1055 \newcommand*{\sub@action}{%
1056 \ifledRcol
1057 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1058 \ifsublines@
1059 \xright@appenditem{-1001}\to\actions@listR
1060 \else
1061 \xright@appenditem{-1002}\to\actions@listR
1062 \fi
1063 \else
1064 \xright@appenditem{\the\absline@num}\to\actionlines@list
1065 \ifsublines@
1066 \xright@appenditem{-1001}\to\actions@list
1067 \else
1068 \xright@appenditem{-1002}\to\actions@list
1069 \fi
1070 \fi}
1071 \fi}
1072 %

```

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

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

```

1073 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1074
1075 \newcommand*{\do@lockon}{%
1076 \ifx\next\lock@off

```

```

1077 \global\let\lock@off=\skip@lockoff
1078 \else
1079 \ifledRcol
1080 \do@lockonR
1081 \else
1082 \do@lockonL
1083 \fi
1084 \fi}
1085
1086
1087 \newcommand*{\do@lockonL}{%
1088 \xright@appenditem{\the\absline@num}\to\actionlines@list
1089 \ifsublines@
1090 \xright@appenditem{-1005}\to\actions@list
1091 \ifnum\sub@lock=\z@
1092 \sub@lock \@ne
1093 \else
1094 \ifnum\sub@lock=\thr@@
1095 \sub@lock \@ne
1096 \fi
1097 \fi
1098 \else
1099 \xright@appenditem{-1003}\to\actions@list
1100 \ifnum\@lock=\z@
1101 \@lock \@ne
1102 \else
1103 \ifnum\@lock=\thr@@
1104 \@lock \@ne
1105 \fi
1106 \fi
1107 \fi}
1108
1109 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1110 \newcommand*{\do@lockoffL}{%
1111 \xright@appenditem{\the\absline@num}\to\actionlines@list
1112 \ifsublines@
1113 \xright@appenditem{-1006}\to\actions@list
1114 \ifnum\sub@lock=\tw@
1115 \sub@lock \thr@@
1116 \else
1117 \sub@lock \z@
1118 \fi
1119 \else
1120 \xright@appenditem{-1004}\to\actions@list
1121 \ifnum\@lock=\tw@
1122 \@lock \thr@@
1123 \else

```

```

1124 \lock \z@
1125 \fi
1126 \fi}
1127
1128 \newcommand*{\do@lockoff}{%
1129 \ifledRcol
1130 \do@lockoffR
1131 \else
1132 \do@lockoffL
1133 \fi}
1134 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1135 \global\let\lock@off=\do@lockoff
1136
1137 %

```

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

```

1138 \newcommand*{\n@num}{%
1139 \ifledRcol%
1140 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1141 \xright@appenditem{-1007}\to\actions@listR
1142 \else%
1143 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1144 \xright@appenditem{-1007}\to\actions@list%
1145 \fi%
1146 }%
1147
1148 %

```

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

```

1149 \newcommand*{\n@num@stanza}{%
1150 \ifledRcol%
1151 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1152 \xright@appenditem{-1008}\to\actions@listR%
1153 \else%
1154 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1155 \xright@appenditem{-1008}\to\actions@list%
1156 \fi%
1157 }
1158 %

```

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

```

\hidenumberingonleftpage
\hidenumberingonrightpage
1159 \newif\ifl@dhidenumber
1160 \newcommand*{\hidenumbering}{

```

```

1161 \ifledRcol%
1162 \write\linenum@outR{\string\hide@num}%
1163 \else%
1164 \write\linenum@out{\string\hide@num}%
1165 \fi%
1166 }%
1167 \newcommand*{\hide@num}{%
1168 \ifledRcol%
1169 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1170 \xright@appenditem{-1009}\to\actions@listR%
1171 \else%
1172 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1173 \xright@appenditem{-1009}\to\actions@list%
1174 \fi%
1175 }
1176 \newcommand*{\hidenumberingonleftpage}{%
1177 \ifledRcol%
1178 \write\linenum@outR{\string\hide@num@left}%
1179 \else%
1180 \write\linenum@out{\string\hide@num@left}%
1181 \fi%
1182 }%
1183
1184 \newcommand*{\hide@num@left}{%
1185 \ifledRcol%
1186 \ifodd\page@numR\else%
1187 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1188 \xright@appenditem{-1009}\to\actions@listR%
1189 \fi%
1190 \else%
1191 \ifodd\page@num\else%
1192 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1193 \xright@appenditem{-1009}\to\actions@list%
1194 \fi%
1195 \fi%
1196 }%
1197
1198 \newcommand*{\hidenumberingonrightpage}{%
1199 \ifledRcol%
1200 \write\linenum@outR{\string\hide@num@right}%
1201 \else%
1202 \write\linenum@out{\string\hide@num@right}%
1203 \fi%
1204 }%
1205
1206 \newcommand*{\hide@num@right}{%
1207 \ifledRcol%
1208 \ifodd\page@numR%
1209 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1210 \xright@appenditem{-1009}\to\actions@listR%

```

```

1211 \fi%
1212 \else%
1213 \ifodd\page@num%
1214 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1215 \xright@appenditem{-1009}\to\actions@list%
1216 \fi%
1217 \fi%
1218 }%
1219
1220 %

```

\@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.

```

1221 \newcount\insert@count
1222 %

```

- #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.

```

1223 \newcommand*\@dummy@ref}[2]{#2}
1224 %

```

\@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.

```

1225 \newcommand*\@ref}[2]{%
1226 \ifledRcol%
1227 \@ref@regR{#1}{#2}%
1228 \else%
1229 \@ref@reg{#1}{#2}%
1230 \fi%
1231 }%
1232 \newcommand*\@ref@reg}[2]{%
1233 \global\insert@count=#1\relax
1234 \global\advance\@edtext@level by 1%
1235 \loop\ifnum\insert@count>\z@
1236 \xright@appenditem{\the\absline@num}\to\insertlines@list
1237 \global\advance\insert@count \m@ne
1238 \repeat
1239 %

```

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.

```

1240 \begingroup
1241   \let\@ref=\dummy@ref
1242   \let\@lopL\@gobble
1243   \let\page@action=\relax
1244   \let\sub@action=\relax
1245   \let\set@line@action=\relax
1246   \let\@lab=\relax
1247   \let\@lemma=\relax%
1248   \let\@sw\@gobblethree%
1249   #2
1250   \global\endpage@num=\page@num
1251   \global\endline@num=\line@num
1252   \global\endsubline@num=\subline@num
1253 \endgroup
1254 %

```

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

```

1255 \xright@appenditem%
1256   {\the\page@num|\the\line@num|%
1257    \ifsublines@ \the\subline@num \else 0\fi}%
1258   \the\endpage@num|\the\endline@num|%
1259   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1260 %

```

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

```

1261   \@ref@reg@parse{#2}%
1262 %

```

Decrease edtext level counter.

```

1263   \global\advance\@edtext@level by -1%
1264 }
1265 %

```

`\@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.

```

1266 \newcommand{\@ref@reg@parse}[1]{%
1267   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
1268   @edtext@level\endcsname}%
1269   \providebool{lemmacommand@\the\@edtext@level}%
1270   \boolfalse{lemmacommand@\the\@edtext@level}%

```


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

```
1271 #1%
1272 %
```

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.

```
1273 \ifnum\@edtext@level>0%
1274 \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1275 \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level
}{}%
1276 \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
1277 \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}%
1278 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1279 \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
1280 \fi%
1281 %

1282 }
1283
1284 %
```

`\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.

```
1285 \newcommand{\@ref@later}[1]{%
1286 \global\advance\@edtext@level by \@ne%
1287 \ifledRcol%
1288 \@ref@reg@parseR{#1}%
1289 \else%
1290 \@ref@reg@parse{#1}%
1291 \fi%
1292 \global\advance\@edtext@level by -\@ne%
1293 }%
1294 %
```

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`.

```
1295 \newwrite\linenum@out
1296 %
```

`\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.

```
1297 \newif\iffirst@linenum@out@
1298 \first@linenum@out@true
1299 %
```

`\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.

```
1300 \newcommand{\this@line@list@version}{5}%
1301 %
```

`\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.

```
1302 \newcommand*{\line@list@stuff}[1]{%
1303 %
```

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

```
1304 \read@linelist{#1}%
1305 %
```

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.

```
1306 \iffirst@linenum@out@
1307 \immediate\closeout\linenum@out%
1308 \global\first@linenum@out@false%
1309 \immediate\openout\linenum@out=\l@auxdir#1\relax%
1310 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1311 \ifl@dpaging%
1312 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
```

```

1313 \fi%
1314 \else
1315 %

```

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.

```

1316 \if@minipage%
1317 \leavevmode%
1318 \fi%
1319 \closeout\linenum@out%
1320 \openout\linenum@out=\l@auxdir#1\relax%
1321 \write\linenum@out{\string\line@list@version{\this@line@list@version}}
1322 %
1323 \ifl@dpaging%
1324 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
1325 %
1326 \fi%
1327 \fi}
1328 %

```

\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.

```

1328 \newcommand*{\new@line}{%
1329 \IfStrEq{\led@pb@setting}{after}%
1330 {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1331 {\xifinlist{\the\absline@num}{\normal@page@break}%
1332 {\numgdef{\@next@page}{\c@page+\@ne}%
1333 \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%
1334 }%
1335 {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1336 }%
1337 {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1338 }%
1339 \IfStrEq{\led@pb@setting}{before}%
1340 {\numgdef{\next@absline}{\the\absline@num+\@ne}%
1341 \xifinlist{\next@absline}{\l@prev@nopb}%
1342 {\xifinlist{\the\absline@num}{\normal@page@break}%
1343 {\numgdef{\nc@page}{\c@page+\@ne}%
1344 \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1345 }%
1346 {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1347 }%
1348 {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}%
1349 }%
1350 {}%
1351 \IfStrEqCase{\led@pb@setting}{\before}{\relax}{\after}{\relax}}{\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1352 }

```

1353

1354 %

\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

```

1355 \newif\if@noneed@Footnote%
1356
1357 \newcommand*{\flag@start}{%
1358   \ifledRcol%
1359     \edef\next{\write\linenum@outR{%
1360       \string\@ref[\the\insert@countR] []}%
1361     \next%
1362     \ifnum\insert@countR<1%
1363       \if@noneed@Footnote\else%
1364         \led@err@EdtextWithoutFootnote%
1365       \fi%
1366     \fi%
1367   \else%
1368     \edef\next{\write\linenum@out{%
1369       \string\@ref[\the\insert@count] []}%
1370     \next%
1371     \ifnum\insert@count<1%
1372       \if@noneed@Footnote\else%
1373         \led@err@EdtextWithoutFootnote%
1374       \fi%
1375     \fi%
1376   \fi}%
1377
1378 \newcommand*{\flag@end}{%
1379   \ifledRcol%
1380     \write\linenum@outR{[]}%
1381   \else%
1382     \write\linenum@out{[]}%
1383   \fi}%
1384
1385
1386 %

```

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

\flag@end@later

```

1387 \newcommand*{\flag@start@later}{%
1388   \ifledRcol%
1389     \write\linenum@outR{\string\@ref@later []}%
1390   \else%

```

```

1391 \write\linenum@out{\string\@ref@later{}}%
1392 \fi%
1393 }%
1394 \newcommand{\flag@end@later}{%
1395 \ifledRcol%
1396 \write\linenum@outR{}}%
1397 \else%
1398 \write\linenum@out{}}%
1399 \fi%
1400 }
1401 %

```

\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.

```

1402
1403
1404 \newcommand*{\startsub}{\dimen0\lastskip
1405 \ifdim\dimen0>0pt \unskip \fi
1406 \ifledRcol \write\linenum@outR{\string\sub@on}%
1407 \else \write\linenum@out{\string\sub@on}%
1408 \fi
1409 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1410 \def\endsub{\dimen0\lastskip
1411 \ifdim\dimen0>0pt \unskip \fi
1412 \ifledRcol \write\linenum@outR{\string\sub@off}%
1413 \else \write\linenum@out{\string\sub@off}%
1414 \fi
1415 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1416
1417 %

```

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

```

1418 \newcommand*{\advanceline}[1]{\leavevmode%
1419 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1420 \else \write\linenum@out{\string\@adv[#1]}%
1421 \fi%

```

```
1422 }
1423 %
```

\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.

```
1424
1425 \newcommand*{\setline}[1]{%
1426   \leavevmode%
1427   \ifnum#1<\z@
1428     \led@warn@BadSetline
1429   \else
1430     \ifledRcol \write\linenum@outR{\string\@set[#1]}%
1431     \else      \write\linenum@out{\string\@set[#1]}%
1432     \fi
1433   \fi}
1434
1435 %
```

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

```
1436
1437 \newcommand*{\setlinenum}[1]{%
1438   \ifnum#1<\z@
1439     \led@warn@BadSetlinenum
1440   \else
1441     \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1442     \else      \write\linenum@out{\string\l@d@set[#1]} \fi
1443   \fi}
1444
1445 %
```

\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.

```
1446
1447 \newcommand*{\startlock}{%
1448   \ifledRcol \write\linenum@outR{\string\lock@on}%
1449   \else      \write\linenum@out{\string\lock@on}%
1450   \fi}
1451 \def\endlock{%
1452   \ifledRcol \write\linenum@outR{\string\lock@off}%
1453   \else      \write\linenum@out{\string\lock@off}%
1454   \fi}
1455 %
```

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

```

1456 \newif\ifl@dskipnumber
1457 \newif\ifl@dskipversenumber%
1458 \newcommand*{\skipnumbering}{%
1459   \leavevmode%
1460   \ifledRcol%
1461     \ifinstanza%
1462       \write\linenum@outR{\string\n@num@stanza}%
1463     \else%
1464       \write\linenum@outR{\string\n@num}%
1465     \fi%
1466     \advanceline{-1}%
1467   \else%
1468     \ifinstanza%
1469       \write\linenum@out{\string\n@num@stanza}%
1470     \else%
1471       \write\linenum@out{\string\n@num}%
1472     \fi%
1473     \advanceline{-1}%
1474   \fi%
1475 }%
1476
1477 %

```

VI Marking text for notes

The `\edtext` macro is used to create all footnotes and endnotes, as well as to print the portion of the main text to which a given note or notes is keyed. The idea is to have that lemma appear only once in the `.tex` file: all instances of it in the main text and in the notes are copied from that one appearance.

The `\edtext` macro takes two arguments.

```
\edtext{#1}{#2}
```

- `#1` is the piece of the main text being glossed; it gets added to the main text, and is also used as a lemma for notes to it.
- `#2` is a series of subsidiary macros that generate various kinds of notes.

The `\edtext` macro may be used (somewhat) recursively; that is, `\edtext` may be used within its own first argument. The code would be much simpler without this feature, but nested notes will commonly be necessary: it is quite likely that we will have an explanatory note for a long passage and notes on variants for individual words within that passage. The situation we can't handle is overlapping notes that are not nested: for example, one note covering lines 10–15, and another covering 12–18. You can handle such cases by using the `\lemma` and `\linenum` macros within `#2`: they alter the copy of the lemma and the line numbers that are passed to the notes, and hence allow you to overcome any limitations of this system, albeit with extra effort.

The recursive operation of `\edtext` will fail if you try to use a copy that is called something other than `\edtext`. In order to handle recursion, `\edtext` needs to redefine its own definition temporarily at one point, and that does not work if the macro you are calling is not actually named `\edtext`. There is no problem as long as `\edtext` is not invoked in the first argument. If you want to call `\edtext` something else, it is best to create instead a macro that expands to an invocation of `\edtext`, rather than copying `\edtext` and giving it a new name; otherwise you will need to add an appropriate definition for your new macro to `\morenoexpands`.

Side effects of our line-numbering code make it impossible to use the usual footnote macros directly within a paragraph whose lines are numbered (see comments to `\do@line`, VII.2.1 p. 140). Instead, the appropriate note-generating command is appended to the list macro `\inserts@list`, and when `\pend` completes the paragraph it inserts all the notes at the proper places.

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

VI.1 `\edtext` itself

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

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

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

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

```
1478 \list@create{\end@lemmas}
1479 %
```


`\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.

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

`\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`.

```
1482 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1483 %
```

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

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

`\morenoexpands`

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

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

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

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

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

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

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

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

```

1484 \newcommand*{\no@expands}{%
1485   \let\select@lemmafnt=0%
1486   \let\startsub=\relax \let\endsub=\relax
1487   \let\startlock=\relax \let\endlock=\relax
1488   \let\edlabel=\@gobble
1489   \let\setline=\@gobble \let\advanceline=\@gobble
1490   \let\sameword\sameword@inedtext%
1491   \let\edtext=\dummy@edtext
1492   \l@dtabnoexpands
1493   \morenoexpands}
1494 \let\morenoexpands=\relax
1495
1496 %

```

\@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.

```

1497 \newcommand{\@tag}{}
1498 %

```

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

```

1499 \newcount\@edtext@level%
1500 \@edtext@level=0%
1501 %

```

\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.

```

1502 \newif\if@edtext@secondarg@%
1503 %

```

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

```

1504 \newcounter{edtext}
1505 \renewcommand{\theedtext}{edtxt@arabic{edtext}}%
1506 %

```

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

```
1507 \newcommand{\edtext}[2]{\leavevmode%
1508 %
```

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

```
1509 \ifnumberedpar%
1510 %
```

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

```
1511 \global\advance\@edtext@level by 1%
1512 %
```

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

```
1513 \stepcounter{edtext}%
1514 %
```

By default, we do not use `\lemma`

```
1515 \global\@lemmacommand@false%
1516 %
```

```
1517 \begingroup%
1518 %
```

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`.

```
1519 \ifledRcol%
1520 \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1521 {\global\let\sw@inthisedtext\empty}%
1522 {\expandafter\gl@p\csname sw@list@edtextR@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1523 \else%
1524 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1525 {\global\let\sw@inthisedtext\empty}%
1526 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1527 \fi%
1528 %
```

\@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.

```

1529 \global\renewcommand{\@tag}{%
1530 \no@expands #1%
1531 }%
1532 %

```

`\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`.

```

1533 \set@line%
1534 %

```

`\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`.

```

1535 \ifledRcol \global\insert@countR \z@%
1536 \else \global\insert@count \z@ \fi%
1537 %

```

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.

```

1538 \@edtext@secondarg@true%
1539 \ignorespaces #2\relax%
1540 \@edtext@secondarg@false%
1541 %

```

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

```

1542 \ifundefined{xpg@main@language}{%if not polyglossia
1543 \flag@start}%
1544 {\if@RTL\flag@end\else\flag@start\fi%
1545 }%
1546 %

```

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

```

1547 \if@lemmacommand%
1548 \ifledRcol%
1549 \write\linenum@outR{\string\@lemma}%
1550 \else%
1551 \write\linenum@out{\string\@lemma}%
1552 \fi%
1553 \fi%
1554 %

```

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.

```

1555     \endgroup%
1556     \ifdef{\hypertarget}%
1557     {%
1558         \csedef{thisedtext@the\@edtext@level}{\theedtext}%We need one
macro by level, as #1 can contain new \edtext
1559         \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}{}}%
1560         \showlemma{#1}%
1561         \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}%
1562     }%
1563     {%
1564         \showlemma{#1}%
1565     }%
1566 %

```

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.

```

1567     \ifx\end@lemmas\empty \else%
1568     \gl@p\end@lemmas\to\x@lemma%
1569     \x@lemma%
1570     \global\let\x@lemma=\relax%
1571     \fi%
1572     \ifundefined{xpg@main@language}{%if not polyglossia
1573         \flag@end}%
1574         {\ifRTL\flag@start\else\flag@end\fi% With polyglossia, you must
track whether the language reads left to right (English) or right to left
(Arabic).
1575         }%
1576 %

```

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`.

```

1577     \global\@noneed@Footnotefalse%
1578     \global\advance\@edtext@level by -1%

```

```

1579 \global\@lemmacommand@false%
1580 %

```

We also reset `\@beforeinsertofthistext`

```

1581 \global\let\@beforeinsertofthistext\relax%
1582 %

```

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

```

1583 \else%
1584 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidestart%
1585 \fi%
1586 }%
1587
1588
1589 %

```

`\@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`.

```

1590 \let\@beforeinsertofthistext\relax
1591 %

```

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

```

1592 \newif\ifnumberline
1593 \numberlinetrue
1594 %

```

`\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.

```

1595 \newcommand*{\set@line}{%
1596 \ifl@edRcol
1597 \ifx\line@listR\empty
1598 \global\noteschanged@true
1599 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1600 \else
1601 \gl@p\line@listR\to\@tempb
1602 \xdef\l@d@nums{\@tempb|\edfont@info}%

```

```

1603 \global\let\@tempb=\undefined
1604 \fi
1605 \else
1606 \ifx\line@list\empty
1607 \global\noteschanged@true
1608 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1609 \else
1610 \gl@p\line@list\to\@tempb
1611 \xdef\l@d@nums{\@tempb|\edfont@info}%
1612 \global\let\@tempb=\undefined
1613 \fi
1614 \fi}
1615
1616 %

```

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

```

1617 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1618
1619 %

```

VI.2 Substitute lemma

\lemma The `\lemma{<text>}` macro allows you to change the lemma that is passed on to the notes. Read about `\@tag` in normal `\edtext` macro for more details about `\sw@list@inedtext` and `\no@expands` (VI.1 p. 123).

```

1620 \newcommand*{\lemma}[1]{%
1621 \global\@lemmacommand@true%
1622 \global\renewcommand{\@tag}{%
1623 \no@expands #1%
1624 }%
1625 \ignorespaces%
1626 }%
1627 %

```

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

```

1628 \newcommand{\@lemma}{%
1629 \booltrue{lemmacommand@the\@edtext@level}%
1630 }%
1631 %

```

\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.

```

1632 \newif\if@lemmacommand@%
1633 %

```

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@dd@nums` (see V.9 p. 95): the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma. However, you can omit any parameters you do not want to change, and you can omit a string of vertical bars at the end of the argument. Hence `\linenum{18|4|0|18|7|1|0}` is an invocation that changes all the parameters, but `\linenum{13}` only changes the starting line number, and leaves the rest unaltered.

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

```

1634 \newcommand*{\linenum}[1]{%
1635   \xdef\@tempa{#1|}|}|}|}|}|}|noexpand\\l@d@nums}%
1636   \global\let\l@d@nums=\empty
1637   \expandafter\line@set\@tempa|\\ignorespaces}
1638 %

```

`\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.

```

1639 \def\line@set#1\#2\#3\#4\{\%
1640 \gdef\@tempb{#1}%
1641 \ifx\@tempb\empty
1642 \l@{d@add{#3}%
1643 \else
1644 \l@{d@add{#1}%
1645 \fi
1646 \gdef\@tempb{#4}%
1647 \ifx\@tempb\empty\else
1648 \l@{d@add{}}\line@set#2\#4\}%
1649 \fi}
1650 %

```

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

```

1651 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1652
1653 %

```

VI.4 Lemma disambiguation

The mechanism which counts the occurrence of a same word in a same line is quite complex, because, when `LTEX` reads a command between a `\pstart` and a `\pend`, it does not know yet which are the line numbers.

The general mechanism is the following:

- **At the first run**, each `\sameword` command increments an `etoolbox` counter the name of which contains the argument of the `\sameword` commands.
- Then this counter, associated with the argument of `\sameword` is stored with the `\@sw` command in the auxiliary file of the current `reledmac` section (the `.1`, `.2...` file).
- **When this auxiliary file is read at the second run**, different operations are achieved:

1. Get the rank of each `\sameword` in a line (relative rank) from the rank of each `\sameword` in all the numbered section (absolute rank):

- For each paired `\sameword` argument and absolute line number, a counter is defined. Its value corresponds to the number of times `\sameword{<argument>}` is called from the beginning of the lineation to the end of the current line. We also store the same data for the preceding absolute line number, if it does not have `\sameword{<argument>}`.
- For each `\sameword` having the same argument, we subtract from its absolute rank the number stored for the paired `\sameword` argument and previous absolute line number. Consequently, we obtain the relative rank.
- See the following example which explains how, for same `\sameword`, absolute ranks are transformed to relative ranks.

```
At line 1:
absolute rank 1 becomes relative rank 1-0 = 1
1 is stored for this \sameword and line 1
At line 2:
absolute rank 2 becomes relative rank 2-1 = 1
absolute rank 3 becomes relative rank 3-1 = 2
3 is stored for this \sameword and line 2
At line 3:
no \sameword for this line.
3 is stored for this \sameword and line 3
At line 4:
absolute rank 4 becomes relative rank 4-3 = 1
4 is stored for this \sameword and line 4
```

2. Create lists of lists of `\sameword` by depth of `\edtext`. That is: create a list for `\edtexts` of level 1, a list for `\edtexts` of level 2, a list for `\edtexts` of level 3 etc. For each `\edtext` in these lists, we store all of the relative ranks of `\saweword` which are called as lemma information. That is: 1) either called in the first argument of `\sameword`, or, 2) called in the `\lemma` macro of the second argument of `\sameword` AND marked by the optional argument of `\saweword` in first argument of `\edtext`.

For example, suppose a line with nested `\edtexts` which contains some word marked by `\sameword` and having the following relative rank:

bar¹ foo¹ foo² bar² foo³ (A)(B) foo⁴ bar³ (C) foo⁵ (D) bar⁴ (E)

In this example, all lemma information for `\edtext` is framed. The text in parenthesis is the content of critical notes associated to the preceding frame. As you can see, we have two level of `\edtext`.

The list for `\edtexts` of level 1 is $\{\{1, 2, 2, 3, 4, 3\}, \{5, 4\}\}$.

The list for `\edtexts` of level 2 is $\{\{1, 2, 2, 3\}, \{5\}\}$.

As you can see, the mandatory argument of `\sameword` does not matter: we store the rank informations for every word potentially ambiguous.

- At the second run, when a critical notes is called, we associate it to the next item of the list associated to its `\edtext` level. So, in the previous example:
 - Critical notes (A) and (B) are associated with $\{1, 2, 2, 3\}$.
 - Critical note (C) is associated with $\{1, 2, 2, 3, 4, 3\}$.
 - Critical note (D) is associated with $\{5\}$.
 - Critical note (E) is associated with $\{5, 4\}$.
- At the second run, when a critical note is printed:
 - The `\sameword` command is let `\sameword@inedtext`.
 - At each call of this `\sameword@inedtext`, we step to the next element of the list associated to the note. Let it be r .
 - For the word marked by `\sameword`, we calculate how many time it is called in its line. To do it:
 - * We get the absolute line number of the current `\sameword`. This absolute line number was stored with a list of relative ranks for the current `\edtext`. That means, in the previous example, that if the absolute line number of `\edtext` was 1, that critical notes (A) and (B) were not associated with $\{1, 2, 2, 3\}$ but with $\{(1, 1), (2, 1), (2, 1), (3, 1)\}$. Such a method of knowing the absolute line number associated to a `\sameword` is required because a `\edtext` can overlap many lines, but `\sameword` can't get it.
 - * When reading the auxiliary file, we get the value associated to the pair composed by the current marked word and the current absolute line number. To this value, we subtract the value associated to the pair composed by the current marked word and the previous absolute line number. Let the result be n .
 - If $n > 1$, that means the current word appears more than once in its line. In this case, we call `\showwordrank` with the word as the first argument and r as the second argument. If the word is called only once, we just print it.

After theory, implementation.

`\get@sw@txt` As the argument of `\sameword` can contain an active character if we use `inputenc` with `utf8` option instead of native UTF-8 engine, we store its detokenized content in a macro in order to allow the dynamic name of macro with `\csname`.²⁶

Because there is a bug with `\detokenize` and \LaTeX when using non BMP characters²⁷, we detokenize only for non- \LaTeX engines. In any case, in \LaTeX `\csname` construction can contain UTF-8 characters without a problem, as UTF-8 characters are not managed with category codes, but instead read directly as UTF-8 characters.

```

1654 \newcommand{\get@sw@txt}[1]{%
1655   \ifxetex%
1656     \xdef\sw@txt{#1}%
1657   \else%
1658     \expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize{#1}}%
1659   \fi%
1660 }%
1661 %

```

`\sameword` The high level macro `\sameword`, used by the editor.

```

1662 \newcommandx{\sameword}[2][1,usedefault]{%
1663   \leavevmode%
1664   \get@sw@txt{#2}%
1665 %

```

Now, the real code. First, increment the counter corresponding to the argument.

```

1666 \unless\ifledRcol%
1667   \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1668 %

```

Then, write its value to the numbered file.

```

1669   \protected@write\linenum@out{\string\sw@\sw@txt}{\csuse{sw@\sw@txt}
1670   }\{#1}}%
1671 %

```

Do the same thing if we are in the right column.

```

1671 \else%
1672   \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1673   \protected@write\linenum@outR{\string\sw@\sw@txt}{\csuse{sw@\sw@txt}
1674   }\{#1}}%
1675   \fi%
1676 %

```

And print the word.

```

1676   #2%
1677 }%
1678 %

```

²⁶See <http://tex.stackexchange.com/q/244538/7712>.

²⁷<http://sourceforge.net/p/xetex/bugs/108/>

A flag set to true if a \@sw relative rank must be added to the list of ranks for a specific \edtext.

```
\if@addsw%
1680 %
```

\@sw The command printed in the auxiliary files.

```
1681 \newcommand{\@sw}[3]{%
1682   \get@sw@txt{#1}%
1683   \unless\ifledRcol%
1684 %
```

First, define a counter which store the second argument as value for a each paired absolute line number/first argument

```
1685   \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
1686 %
```

If such argument was not defined for the preceding line, define it.

```
1687   \numdef{\prev@line}{\the\absline@num-1}%
1688   \ifcsundef{sw@\sw@txt @\prev@line @\the\section@num}{%
1689     \csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
1690   }{}%
1691 %
```

Then, calculate the position of the word in the line.

```
1692   \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1693 %
```

And do the same thing for the right side.

```
1694   \else%
1695     \csxdef{sw@\sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1696     \numdef{\prev@line}{\the\absline@numR-1}%
1697     \ifcsundef{sw@\sw@txt @\prev@line @\the\section@numR @R}{%
1698       \csnumgdef{sw@\sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1699     }{}%
1700     \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
1701   \fi%
1702 %
```

And now, add it to the list of \@sw for the current edtext, in all depth.

```
1703   \@tempcnta=\@edtext@level
1704   \@whilenum{\@tempcnta>0}\do{%
1705     \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1706     {%
1707       \@addswfalse%
1708       \notbool{lemmacommand@\the\@tempcnta}%
1709       {\@addswtrue}%

```

```

1710      {\IfStrEq{#3}{inlemma}%
1711       {\@addswtrue}%
1712       {%
1713        \def\do##1{%
1714          \ifnumequal{##1}{\the\@tempcnta}%
1715          {\@addswtrue\listbreak}%
1716          {}%
1717        }%
1718        \docsvlist{#3}%
1719      }%
1720    }%
1721    \if@addsw%
1722      \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1723      \ifledRcol%
1724        \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
1725      \else%
1726        \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
1727      \fi%
1728      \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
1729    \fi%
1730  }%
1731  {%
1732    \advance\@tempcnta by -1%
1733  }%
1734 }%
1735 %

```

`\sameword@inedtext` The command called when `\sameword` is called in a `\edtext`.

```

1736 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1737   \get@sw@txt{#2}%
1738   \unless\ifledRcol%
1739 %

```

Just a precaution.

```

1740   \ifx\sw@list@inedtext\empty%
1741     \def\the@sw{999}%
1742     \def\this@absline{-99}%
1743   \else%
1744 %

```

But in many cases, at this step, we should have some content in the list `\sw@list@inedtext`, which contains the reference for `\edtext`.

```

1745   \gl@p\sw@list@inedtext\to\@tmp%
1746   \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1747   \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1748   \fi%
1749 %

```

First, calculate the number of occurrences of the word in the current line

```

1750 \ifcsdef{sw@sw@txt @\this@absline @\the\section@num}{%
1751 \numdef{\prev@line}{\this@absline-1}%
1752 \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@num}-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1753 }%
1754 {\numdef{\sw@atthisline}{0}}%
1755 %

```

Finally, print the rank, but only if there is more than one occurrence of the word in the current line.

```

1756 \ifnumgreater{\sw@atthisline}{1}%
1757 {\showwordrank{#2}{\the@sw}}%
1758 {#2}%
1759 %

```

And the same for right side.

```

1760 \else%
1761 \ifx\sw@list@inedtext\empty%
1762 \def\the@sw{999}%
1763 \def\this@absline{-99}%
1764 \else%
1765 \gl@p\sw@list@inedtext\to\@tmp%
1766 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1767 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1768 \fi%
1769 \ifcsdef{sw@sw@txt @\this@absline @\the\section@numR @R}{%
1770 \numdef{\prev@line}{\this@absline-1}%
1771 \numdef{\sw@atthisline}{\csuse{sw@sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
1772 }%
1773 {\numdef{\sw@atthisline}{0}}%
1774 \ifnumgreater{\sw@atthisline}{1}%
1775 {\showwordrank{#2}{\the@sw}}%
1776 {#2}%
1777 \fi%
1778 }%
1779 %

```

`\showwordrank` Finally, the way the rank will be printed.

```

1780 \newcommand{\showwordrank}[2]{%
1781 #1\textsuperscript{#2}%
1782 }%
1783 %

```

VII Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than

straight onto the vertical list, and when the paragraph ends we slice the paragraph into its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

VII.1 Boxes, counters, \pstart and \pend

`\raw@text` Here are numbers and flags that are used internally in the course of the paragraph decomposition.

`\ifnumberedpar@` When we first form the paragraph, it goes into a box register, `\raw@text`, instead of onto the current vertical list. The `\ifnumberedpar@` flag will be true while a paragraph is being processed in that way. `\num@lines` will store the number of lines in the paragraph when it is complete. When we chop it up into lines, each line in turn goes into the `\one@line` register, and `\par@line` will be the number of that line within the paragraph.

`\num@lines`

`\one@line`

`\par@line`

```
1784 \newbox\raw@text
1785 \newif\ifnumberedpar@
1786 \newcount\num@lines
1787 \newbox\one@line
1788 \newcount\par@line
1789 %
```

`\pstart` `\pstart` starts the paragraph by clearing the `\inserts@list` list and other relevant variables, and then arranges for the subsequent text to go into the `\raw@text` box.

`\AtEveryPstart` `\pstart` needs to appear at the start of every paragraph that is to be numbered; the `\autopar` command below may be used to insert these commands automatically.

`\numberpstarttrue`

`\numberpstartfalse`

`\labelpstarttrue` Beware: everything that occurs between `\pstart` and `\pend` is happening within a group; definitions must be global if you want them to survive past the end of the paragraph.

`\labelpstartfalse`

`\thepstart`

```
1790
1791 \newcommand{\AtEveryPstart}[1]{%
1792   \ifstrempy{#1}%
1793   {\xdef\at@every@pstart{}}%
1794   {\gdef\at@every@pstart{\noindent#1}}%
1795 }%
1796 \xdef\at@every@pstart{}%
1797
1798 \newcounter{pstart}
1799 \renewcommand{\thepstart}{\bfseries\@arabic{c@pstart}. }
1800 \newif\ifnumberpstart
1801 \numberpstartfalse
1802 \newif\iflabelpstart
1803 \labelpstartfalse
1804 \newcommandx*\pstart[1][1]{%
1805   \normal@pars%
1806   \ifstrempy{#1}{\at@every@pstart}{\noindent#1}%
1807   \ifautopar%
```

```

1808 \autopar%
1809 \fi%
1810 \ifluatex%
1811 \edef\l@luatextextdir@L{\the\textdir}%
1812 \fi%
1813 \@nobraektrue%
1814 \ifnumbering \else%
1815 \led@err@PstartNotNumbered%
1816 \beginnumbering%
1817 \fi%
1818 \ifnumberedpar@%
1819 \led@err@PstartInPstart%
1820 \pend%
1821 \fi%
1822 \list@clear{\inserts@list}%
1823 \global\let\next@insert=\empty%
1824 \begingroup\normal@pars%
1825 \global\advance \l@dnumstartL\@ne
1826 \global\setbox\raw@text=\vbox\bgroup%
1827 \if@nobraek%
1828 \if@afterindent\else%
1829 \noindent%
1830 \fi%
1831 \fi%
1832 \ifautopar\else%
1833 \ifnumberpstart%
1834 \ifinstanza\else%
1835 \ifsidepstartnum\else%
1836 \thepstart%
1837 \fi%
1838 \fi%
1839 \fi%
1840 \fi%
1841 \numberedpar@true%
1842 \iflabelpstart\protected@edef\@currentlabel%
1843 {\p@pstart\thepstart}%
1844 \fi%
1845 \l@dzeropenalties%
1846 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
1847 }
1848 %

```

`\pend` `\pend` must be used to end a numbered paragraph.

```

1849 \newcommandx*{\pend}[1][1]{\ifnumbering \else%
1850 \led@err@PendNotNumbered%
1851 \fi%
1852 \global\l@dskipversenumberfalse%
1853 \ifnumberedpar@ \else%

```



```

1854 \led@err@PendNoPstart%
1855 \fi%
1856 %

```

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.

```

1857 \l@dzeropenalties%
1858 \endgraf\global\num@lines=\prevgraf\egroup%
1859 \global\par@line=0%
1860 %

```

We check if lineation is by `pstart`: in this case, we reset line number, but only in the second line of the `pstart`. We can't reset line number at the beginning of `\pstart`, as `\setline` is parsed at the end of previous `\pend`, and so, we must do it at the end of first line of `pstart`.

```

1861 \csnumdef{pstartline}{0}%
1862 \loop\ifvbox\raw@text%
1863   \csnumdef{pstartline}{\pstartline+\@ne}%
1864   \do@line%
1865   \ifbypstart@%
1866     \ifnumequal{pstartline}{1}{%
1867       \bgroup%
1868       \let\leavevmode\relax%
1869       \setline{1}%
1870       \egroup%
1871       \resetprevline@}{}%
1872     \fi%
1873   \repeat%
1874 %

```

Deal with any leftover notes, and then end the group that was begun in the `\pstart`.

```

1875 \flush@notes%
1876 \endgroup%
1877 \ignorespaces%
1878 %

```

Increase `pstart` counter.

```

1879 \ifnumberpstart%
1880   \global\pstartnumtrue%
1881 \fi%
1882 \addtocounter{pstart}{1}%
1883 %

```

Print the optional argument of `\pend` or the content printed after every `\pend`

```

1884 \normal@pars%
1885 \ifstrepty{#1}{\at@every@pend}{\noindent#1}%
1886 %

```

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.

```

1887 \@nobreakfalse%
1888 \ifautopar%
1889 \autopar%
1890 \fi%
1891 }
1892 %

```

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.

```

\AtEveryPend93
\at@every@pend94 \newcommand{\AtEveryPend}[1]{%
1895 \ifstrepty{#1}%
1896 {\xdef\at@every@pend{}}%
1897 {\gdef\at@every@pend{\noindent#1}}%
1898 }%
1899 \xdef\at@every@pend{}%
1900
1901 %

```

`\l@dzzeropenalties` A macro to zero penalties for `\pend` or `\pstart`.

```

1902 \newcommand*{\l@dzzeropenalties}{%
1903 \brokenpenalty \z@ \clubpenalty \z@
1904 \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
1905 \postdisplaypenalty \z@ \widowpenalty \z@}
1906
1907 %

```

`\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.

```

1908 \newif\ifautopar
1909 \autoparfalse
1910 \newcommand*{\autopar}{
1911   \ifledRcol
1912     \ifnumberingR \else
1913       \led@err@AutoparNotNumbered
1914       \beginnumberingR
1915       \fi
1916     \else
1917       \ifnumbering \else
1918         \led@err@AutoparNotNumbered
1919         \beginnumbering
1920         \fi
1921       \fi
1922     \autopartrue
1923     \everypar{\setbox0=\lastbox
1924               \endgraf \vskip-\parskip
1925               \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\fi\fi
1926               \let\par=\pend}%
1927     \ignorespaces}
1928 %

```

\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.

```

1929 \newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}
1930
1931 %

```

\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.

```

1932 \newif\ifautopar@pause
1933 %

```

VII.2 Processing one line

VII.2.1 General process

`\do@line` The `\do@line` macro is called by `\pend` to do all the processing for a single line of text.
`\l@dunhbox@line`

```

1934 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
1935 \newcommand*{\do@line}{%
1936   {\vbadness=10000
1937     \splittopskip=\z@
1938     \do@linehook
1939   \l@emptyd@ta
1940     \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
1941     \unvbox\one@line \global\setbox\one@line=\lastbox
1942     \getline@num
1943     \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{%}
1944     \ifnum\@lock>\@ne
1945       \inserthangingsymboltrue
1946     \else
1947       \inserthangingsymbolfalse
1948     \fi
1949     \check@pb@in@verse
1950     \ifl@dhidenumber%
1951       \global\l@dhidenumberfalse%
1952       \f@x@l@cks%
1953     \else%
1954       \affixline@num%
1955     \fi%
1956   %

```

Depending whether a sectioning command is called at this pstart or not we print sectioning command or normal line,

```

1957 \xifinlist{\the\l@dunpstartsl}{\eled@sections@}{%
1958   {\print@eledsection}%
1959   {\print@line}%
1960   \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{%}
1961 }%
1962 %

```

VII.2.2 Process for “normal” line

`\print@line` `\print@line` is for normal line, i. e. line without sectioning command.

```

1963 \def\print@line{
1964   %
1965   \affixpstart@num%
1966   %

```

Insert the pstart number in side, if we are in the first line of a pstart.

The line will be boxed, to have the good width.

```
1967 \hb@xt@ \linewidth{%
1968 %
```

User hook.

```
1969 \do@insidelinehook%
1970 %
```

Left line number

```
1971 \l@dld@ta%
1972 %
```

Prepare text to be inserted before notes.

```
1973 \if@firstlineofpage%
1974 \set@Xtxtbeforenotes%
1975 \global\@firstlineofpagefalse%
1976 \fi%
1977 %
```

Insert footnotes made of manuscripts data.

```
1978 \insert@msdata%
1979 %
```

Restore marginal and footnotes.

```
1980 \add@inserts%
1981 \add@Xgroupbyline%
1982 \affixside@note%
1983 %
```

Print left notes.

```
1984 \l@dlsn@te
1985 %
```

Boxes the line, writes information about new line in the numbered file.

```
1986 {\ledllfill\hb@xt@ \wd\one@line{\new@line%
1987 %
```

If we use Lua_{TEX} then restore the direction.

```
1988 \ifluatex%
1989 \texdir\l@luatextextdir@L%
1990 \fi%
1991 %
```

Insert, if needed, the hanging symbol.

```
1992 \inserthangingsymbol%
1993 %
```

And so, print the line.

```

1994 \l@dunhbox@line{\one@line}}%
1995 %

```

Right line number

```

1996 \ledrlfill\l@drd@ta%
1997 %

```

Print right notes.

```

1998 \l@drsn@te
1999 }}%
2000 %

```

And reinsert penalties (for page breaking)...

```

2001 \add@penalties%
2002 }
2003 %

```

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.

```

2004 \def\print@eledsection{%
2005   \if@firstlineofpage%
2006     \set@Xtxtbeforenotes%
2007     \global\@firstlineofpagefalse%
2008   \fi%
2009   \insert@msdata%
2010   \add@inserts%
2011   \add@Xgroupbyline%
2012   \affixside@note%
2013   \numdef{\temp@}{\l@dnumpstartsL-1}%
2014   \xifinlist{\temp@}{\eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
2015   \@eled@sectioningtrue%
2016   \csuse{eled@sectioning@the\l@dnumpstartsL}%
2017   \@eled@sectioningfalse%
2018   \global\csundef{eled@sectioning@the\l@dnumpstartsL}%
2019   \if@RTL%
2020     \hspace{-3\paperwidth}%
2021     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2022   \else%
2023     \hspace{3\paperwidth}%
2024     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2025   \fi%
2026   \vskip-\baselineskip%
2027 }
2028 %

```

VII.2.4 Hooks

`\do@linehook` Two hooks into `\do@line`. The first is called at the beginning of `\do@line`, the second is called in the line box. The second can, for example, have a `\markboth` command inside, the first can not.

```
2029 \newcommand*{\do@linehook}{}
2030 \newcommand*{\do@insidelinehook}{}
2031 %
```

`\dolinehook` These high level commands just redefine the low level commands. They have to be used by user, without `\makeatletter`.

```
2032 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
2033 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
2034
2035 %
```

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 notes.

```
2036 \l@dcsnotetext \newcommand*{\l@emptyd@ta}{%
2037 \l@dcsnotetext@l \gdef\l@dld@ta{}%
2038 \l@dcsnotetext@r \gdef\l@drd@ta{}%
2039 \gdef\l@dcsnotetext@l{}%
2040 \gdef\l@dcsnotetext@r{}%
2041 \gdef\l@dcsnotetext{}%
2042
2043 %
```

`\l@dlsn@te` Zero width boxes of the left and right side notes, together with their kerns.

```
2044 \l@drsn@te \newcommand*{\l@dlsn@te}{%
2045 \hb@xt@ \z@{\hss\box\l@dldp@rbox\kern\ledlsnotesep}}
2046 \newcommand*{\l@drsn@te}{%
2047 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
2048
2049 %
```

`\ledllfill` These macros are called at the left (`\ledllfill`) and the right (`\ledrlfill`) of each numbered line. The initial definitions correspond to the original code for `\do@line`.

```
2050 \newcommand*{\ledllfill}{\hfil}
2051 \newcommand*{\ledrlfill}{}
2052
2053 %
```

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.

```

2054 \newcommand*{\getline@num}{%
2055   \global\advance\absline@num \@ne%
2056   \do@actions
2057   \do@ballast
2058   \ifnumberline
2059     \ifsublines@
2060       \ifnum\sub@lock<\tw@
2061         \global\advance\subline@num \@ne
2062       \fi
2063     \else
2064       \ifnum\@lock<\tw@
2065         \global\advance\line@num \@ne
2066         \global\subline@num \z@
2067       \fi
2068     \fi
2069   \fi
2070 }
2071 %

```

`\do@ballast` The real work in the macro above is done in `\do@actions`, but before we plunge into that, let's get `\do@ballast` out of the way. This macro looks to see if there is an action to be performed on the *next* line, and if it is going to be a page break action, `\do@ballast` decreases the count `\ballast@count` counter by the amount of ballast. This means, in practice, that when `\add@penalties` assigns penalties at this point, TeX will be given extra encouragement to break the page here (see XI.2 p. 154).

`\ballast@count` First we set up the required counters; they are initially set to zero, and will remain so
`\c@ballast` unless you type `\setcounter{ballast}{\langle some figure \rangle}` in your document.

```

2072 \newcount\ballast@count
2073 \newcounter{ballast}
2074 \setcounter{ballast}{0}
2075 %

```

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.

```

2076 \newcommand*{\do@ballast}{\global\ballast@count \z@
2077   \begingroup
2078     \advance\absline@num \@ne
2079     \ifnum\next@actionline=\absline@num
2080       \ifnum\next@action>-1001\relax
2081         \global\advance\ballast@count by -\c@ballast
2082       \fi
2083     \fi

```



```

2084 \endgroup}
2085 %

```

\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`.

```

2086 \newcommand*{\do@actions}{%
2087   \global\let\do@actions@next=\relax
2088   \ifnum\absline@num<\next@actionline\else
2089   %

```

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.

```

2090   \ifnum\next@action>-1001
2091     \global\page@num=\next@action
2092     \global\@firstlineofpagetrue%
2093     \ifbypage@
2094       \global\line@num=\z@ \global\subline@num=\z@
2095       \resetprevline@
2096     \fi
2097     \add@msdata@firstlineofpage%
2098   %

```

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`.)

```

2099   \else
2100     \ifnum\next@action<-4999
2101       \@l@tempcnta=-\next@action
2102       \advance\@l@tempcnta by -5001
2103       \ifsublines@
2104         \global\subline@num=\@l@tempcnta
2105       \else
2106         \global\line@num=\@l@tempcnta
2107       \fi
2108   %

```

We rescale the value in `\@l@tempcnta` so that we can use a case statement.

```

2109   \else
2110     \@l@tempcnta=-\next@action
2111     \advance\@l@tempcnta by -1000
2112     \do@actions@fixedcode
2113   \fi
2114 \fi
2115 %

```

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.

```

2116 \ifx\actionlines@list\empty
2117 \gdef\next@actionline{1000000}%
2118 \else
2119 \gl@p\actionlines@list\to\next@actionline
2120 \gl@p\actions@list\to\next@action
2121 \global\let\do@actions@next=\do@actions
2122 \fi
2123 \fi
2124 %

```

Make the recursive call, if necessary.

```

2125 \do@actions@next}
2126
2127 %

```

`\do@actions@fixedcode` This macro handles the fixed codes for `\do@actions`. It is one big case statement.

```

2128 \newcommand*{\do@actions@fixedcode}{%
2129 \ifcase\@l@dttempcnta
2130 \or% % 1001 = starting sublineation
2131 \global\sublines@true
2132 \or% % 1002 = ending sublineation
2133 \global\sublines@false
2134 \or% % 1003 = starting locking number
2135 \global\@lock=\@ne
2136 \or% % 1004 = ending locking number
2137 \ifnum\@lock=\tw@
2138 \global\@lock=\thr@@
2139 \else
2140 \global\@lock=\z@
2141 \fi
2142 \or% % 1005 = starting locking subnumber
2143 \global\sub@lock=\@ne
2144 \or% % 1006 = ending locking subnumber
2145 \ifnum\sub@lock=\tw@
2146 \global\sub@lock=\thr@@
2147 \else
2148 \global\sub@lock=\z@
2149 \fi
2150 \or% % 1007 = skipping numbering
2151 \l@dskipnumbertrue
2152 \or% % 1008 = skipping numbering in stanza
2153 \l@dskipversenumbertrue%
2154 \or% % 1009 = hiding number
2155 \l@dhidenumbertrue

```

```

2156 \or% % 1010 = inserting msdata
2157 \add@msdata%
2158 \else
2159 \led@warn@BadAction
2160 \fi}
2161
2162
2163 %

```

IX Line number printing

`\affixline@num` `\affixline@num` just puts a left line number into `\l@dld@ta` or a right line number into `\l@drd@ta` if required.

To determine whether we need to affix a line number to this line, we compute the following:

$$n = \text{int}((\text{linenum} - \text{firstlinenum}) / \text{linenumincrement})$$

$$m = \text{firstlinenum} + (n \times \text{linenumincrement})$$

(where *int* truncates a real number to an integer). *m* will be equal to *linenum* only if we are to paste a number on here. However, the formula breaks down for the first line to number (and any before that), so we check that case separately: if `\line@num ≤ \firstlinenum`, we compare the two directly instead of making these calculations.

We compute, in the scratch counter `\@l@tempcnta`, the number of the next line that should be printed with a number (*m* in the above discussion), and move the current line number into the counter `\@l@tempcntb` for comparison.

First, the case when we are within a sub-line range.

```

2164 \newcommand*{\affixline@num}{%
2165 %

```

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).

```

2166 \ifledgroupnotesL@else
2167 \ifnumberline
2168 \ifl@dskipnumber
2169 \global\l@dskipnumberfalse
2170 \else
2171 \ifsublines@
2172 \@l@tempcntb=\subline@num
2173 \ifnum\subline@num>\c@firstsublinenum
2174 \@l@tempcnta=\subline@num
2175 \advance\@l@tempcnta by-\c@firstsublinenum
2176 \divide\@l@tempcnta by\c@sublinenumincrement
2177 \multiply\@l@tempcnta by\c@sublinenumincrement
2178 \advance\@l@tempcnta by\c@firstsublinenum
2179 \else

```

```

2180         \l@dttempcnta=\c@firstsublinenum
2181     \fi
2182 %

```

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.

```

2183     \ch@cksub@l@ck
2184 %

```

Now the line number case, which works the same way.

```

2185     \else
2186         \l@dttempcntb=\line@num
2187 %

```

Check on the `\linenumberlist` If it is `\empty` use the standard algorithm.

```

2188     \ifx\linenumberlist\empty
2189         \ifnum\line@num>\c@firstlinenum
2190             \l@dttempcnta=\line@num
2191             \advance\l@dttempcnta by-\c@firstlinenum
2192             \divide\l@dttempcnta by\c@linenumincrement
2193             \multiply\l@dttempcnta by\c@linenumincrement
2194             \advance\l@dttempcnta by\c@firstlinenum
2195         \else
2196             \l@dttempcnta=\c@firstlinenum
2197         \fi
2198     \else
2199 %

```

The `\linenumberlist` was not `\empty`, so here is Wayne's numbering mechanism. This takes place in `TEX`'s mouth.

```

2200         \l@dttempcnta=\line@num
2201         \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2202         \edef\sc@n@list{\def\noexpand\sc@n@list
2203             ###1,\number\l@dttempcnta,###2|{\def\noexpand\rem@inder
2204 {####2}}}%
2205         \sc@n@list\expandafter\sc@n@list\rem@inder|
2206         \ifx\rem@inder\empty%
2207             \advance\l@dttempcnta\@ne
2208         \fi
2209     \fi

```

A locking check for lines, just like the version for sub-line numbers above.

```

2210     \ch@ck@l@ck
2211 \fi
2212 %

```

The following tests are true if we need to print a line number.

```

2213 \ifnum\@l@tempcnta=\@l@tempcntb
2214 \ifl@dskipversenumber\else
2215 %

```

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
2216 \if@twocolumn
2217 \if@firstcolumn
2218 \gdef\l@dld@ta{\llap{\leftlinenum}}%
2219 \else
2220 \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2221 \fi
2222 \else
2223 \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}}%
2224 {\@l@tempcntb=\line@margin@columns}%
2225 {\@l@tempcntb=\line@margin}%
2226 \ifnum\@l@tempcntb>\@ne
2227 \advance\@l@tempcntb \page@num
2228 \fi
2229 \ifodd\@l@tempcntb
2230 \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2231 \else
2232 \gdef\l@dld@ta{\llap{\leftlinenum}}%
2233 \fi
2234 \fi
2235 \fi
2236 \fi
2237 %

```

Now fix the lock counters, if necessary. A value of 1 is advanced to 2; 3 advances to 0; other values are unchanged.

```

2238 \f@x@l@cks
2239 \fi
2240 \fi
2241 \fi
2242 }
2243
2244 %

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck`
`\ch@ck@l@ck` checks subline locking. If it fails, then we disable the line-number display by setting the
`\f@x@l@cks` counters to arbitrary but unequal values.

```

2245 \newcommand*{\ch@cksub@l@ck}{%
2246   \ifcase\sub@lock
2247     \or
2248       \ifnum\sublock@disp=\@ne
2249         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2250       \fi
2251     \or
2252       \ifnum\sublock@disp=\tw@ \else
2253         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2254       \fi
2255     \or
2256       \ifnum\sublock@disp=\z@
2257         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2258       \fi
2259   \fi}
2260 %

```

Similarly for line numbers.

```

2261 \newcommand*{\ch@ck@l@ck}{%
2262   \ifcase@lock
2263     \or
2264       \ifnum\lock@disp=\@ne
2265         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2266       \fi
2267     \or
2268       \ifnum\lock@disp=\tw@ \else
2269         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2270       \fi
2271     \or
2272       \ifnum\lock@disp=\z@
2273         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2274       \fi
2275   \fi}
2276 %

```

Fix the lock counters. A value of 1 is advanced to 2; 3 advances to 0; other values are unchanged.

```

2277 \newcommand*{\f@x@l@cks}{%
2278   \ifcase@lock
2279     \or
2280       \global@lock=\tw@
2281     \or \or
2282       \global@lock=\z@
2283     \fi
2284   \ifcase\sub@lock
2285     \or

```

```

2286 \global\sub@lock=\tw@
2287 \or \or
2288 \global\sub@lock=\z@
2289 \fi}
2290
2291 %

```

X Pstart number printing in side

In side, the printing of pstart number is running like the printing of line number. There is only some differences:

- The pstarts counter is upgrade in the \pend command. Consequently, the \affixpstart@num command has not to upgrade it, unlike the \affixline@num which upgrades the lines counter.
- To print the pstart number only at the beginning of a pstart, and not in every line, a boolean test is made. The \pstartnum boolean is set to TRUE at every \pend. It is tried in the \leftpstartnum and \rightpstartnum commands. After the try, it is set to FALSE.

```

\leftpstartnum%2
\rightpstartnum%3
\ifsidepstartnum%4
2295 \ifsidepstartnum
2296 \if@twocolumn
2297 \if@firstcolumn
2298 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2299 \else
2300 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2301 \fi
2302 \else
2303 \l@dttempcntb=\line@margin%
2304 \ifnum\l@dttempcntb>\@ne
2305 \advance\l@dttempcntb \page@num
2306 \fi
2307 \ifodd\l@dttempcntb
2308 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2309 \else
2310 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2311 \fi
2312 \fi
2313 \fi
2314
2315 }
2316 %
2317
2318 \newif\ifpstartnum

```

```

2319 \pstartnumtrue
2320 \newcommand*{\leftpstartnum}{
2321     \ifpstartnum\thepstart
2322     \kern\linenumsep\fi
2323     \global\pstartnumfalse
2324 }
2325 \newcommand*{\rightpstartnum}{
2326     \ifpstartnum
2327     \kern\linenumsep
2328     \thepstart
2329     \fi
2330     \global\pstartnumfalse
2331 }
2332 %

```

XI Restoring footnotes and penalties

Because of the paragraph decomposition process in order to number line, `reledmac` must hack the standard way \TeX works in order to manage insertion of footnotes, both critical and familiar.

We need to call the `\insert` commands not when the content of `\pstart...\pend` is read by \TeX but when each individual line is typeset.

Consequently, when reading the content of `\pstart...\pend`, we store the insertion (footnotes) in an specific `reledmac`'s list, and we restore them to the vertical list when printing each individual line.

XI.1 Add insertions to the vertical list

`\inserts@list` `\inserts@list` is the list macro that contains the inserts that we save up for one paragraph.

```

2333 \list@create{\inserts@list}
2334 %

```

`\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`.

```

2335 \newcommand*{\add@inserts}{%
2336     \global\let\add@inserts@next=\relax
2337 %

```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.


```
2338 \ifx\inserts@list\empty \else
2339 %
```

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.

```
2340 \ifx\next@insert\empty
2341 \ifx\insertlines@list\empty
2342 \global\noteschanged@true
2343 \gdef\next@insert{100000}%
2344 \else
2345 \glp\insertlines@list\to\next@insert
2346 \fi
2347 \fi
2348 %
```

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.

```
2349 \ifnum\next@insert=\absline@num
2350 \glp\inserts@list\to\@insert
2351 \@insert
2352 \global\let\@insert=\undefined
2353 \global\let\next@insert=\empty
2354 \global\let\add@inserts@next=\add@inserts
2355 \fi
2356 \fi
2357 %
```

Make the recursive call, if necessary.

```
2358 \add@inserts@next}
2359
2360 %
```

`\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`.

```
2361 \newcommand{\add@Xgroupbyline}{%
2362 \unless\ifnocritical@%
2363 \def\do##1{%Looping on the series
2364 \def\do###1{%Looping on the ##1@forinserting command
2365 \ifcsdef{##1@forinserting@###1}{%
2366 \X@beforeinsertion{##1}%
2367 \if@ledgroup%
2368 \global\setbox\@nameuse{mp##1footins}=\vbox%
2369 \else%
2370 \insert\csize ##1footins\endcsize%
2371 \fi%

```

```

2372      {%
2373      \ifcsdef{Xhsize\csuse{series@display##1}@@##1}%
2374      {\hsize \csuse{Xhsize\csuse{series@display##1}@@##1}}%
2375      }%
2376      \if@ledgroup%
2377      \unvbox\@nameuse{mp##1footins}%
2378      \fi%
2379      \X@atbegininsertion{##1}%
2380      \ifcsstring{series@display##1}%
2381      {%
2382      \Xledsetnormalparstuff{##1}%
2383      \rule{z@\splittopskip}%
2384      }%
2385      {%
2386      \csuse{##1@forinserting@####1}%
2387      \strut\par%
2388      }%
2389      \global\csundef{##1@forinserting@####1}%
2390      }%
2391      }%
2392      }%
2393      \ifcsdef{##1@forinserting}{%
2394      \dolistcsloop{##1@forinserting}%
2395      }{%
2396      \global\csundef{##1@forinserting}%
2397      }%
2398      \dolistloop{\@series}%
2399      \fi%
2400      }%
2401
2402
2403      %

```

XI.2 Penalties

\add@penalties \add@penalties is the last macro used by \do@line. It adds up the club, widow, and interline penalties, and puts a single penalty of the appropriate size back into the paragraph; these penalties get removed by the \vsplit operation. \displaywidowpenalty and \brokenpenalty are not restored, since we have no easy way to find out where we should insert them.

In this code, \num@lines is the number of lines in the whole paragraph, and \par@line is the line we are working on at the moment. The count \@l@dttempcnta is used to calculate and accumulate the penalty; it is initially set to the value of \ballast@count, which has been worked out in \do@ballast above (VIII p. 144). Finally, the penalty is checked to see that it does not go below -10000 .

```

2404 \newcommand*{\add@penalties}{\@l@dttempcnta=\ballast@count
2405   \ifnum\num@lines>\@ne
2406   \global\advance\par@line \@ne

```

```

2407 \ifnum\par@line=\@ne
2408 \advance\@l@dttempcnta \clubpenalty
2409 \fi
2410 \@l@dttempcntb=\par@line \advance\@l@dttempcntb \@ne
2411 \ifnum\@l@dttempcntb=\num@lines
2412 \advance\@l@dttempcnta \widowpenalty
2413 \fi
2414 \ifnum\par@line<\num@lines
2415 \advance\@l@dttempcnta \interlinepenalty
2416 \fi
2417 \fi
2418 \ifnum\@l@dttempcnta=\z@
2419 \relax
2420 \else
2421 \ifnum\@l@dttempcnta>-10000
2422 \penalty\@l@dttempcnta
2423 \else
2424 \penalty -10000
2425 \fi
2426 \fi}
2427
2428 %

```

XI.3 Printing leftover notes

\flush@notes The `\flush@notes` macro is called after the entire paragraph has been sliced up and sent on to the vertical list. If the number of notes to this paragraph has increased since the previous run of \TeX , then there can be leftover notes that have not yet been printed. An appropriate error message will be printed elsewhere; but it is best to go ahead and print these notes somewhere, even if it is not in quite the right place. What we do is dump them all out here, so that they should be printed on the same page as the last line of the paragraph. We can hope that is not too far from the proper location, to which they will move on the next run.

```

2429 \newcommand*{\flush@notes}{%
2430 \xloop
2431 \ifx\inserts@list\empty \else
2432 \glp\inserts@list\to\@insert
2433 \@insert
2434 \global\let\@insert=\undefined
2435 \repeat}
2436
2437 %

```

\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 Kabelschacht in *TUGboat* **8** (1987), pp. 184–5.

```

2438 \def\@xloop#1\repeat{%
2439   \def\body{#1\expandafter\body\fi}%
2440   \body}
2441
2442 %

```

XI.4 Text before notes

`\set@Xtxtbeforenotes` The `\set@Xtxtbeforenotes` macro resets the `Xtxtbeforenotes@⟨series⟩@typeset` boolean to false. Just before the first note of the `⟨series⟩` in a page, the `Xtxtbeforenotes` will be inserted.

```

2443 \newcommand{\set@Xtxtbeforenotes}{%
2444   \unless\ifnocritical@%
2445     \def\do##1{%
2446       \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2447     }%
2448     \dolistloop{\@series}%
2449   \fi%
2450 }%
2451 %

```

`\insert@Xtxtbeforenotes` `\insert@Xtxtbeforenotes{⟨series⟩}`, called when inserting a note, will insert the text before the note if it is not already inserted. For paragraphed footnotes, it will insert it as a component of the first footnote. For other types of footnotes, it will insert it as a regular footnote.

```

2452 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2453   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2454     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2455     \ifcsvoid{Xtxtbeforenotes@#1}{}%
2456     \ifcsstring{series@display#1}{paragraph}%
2457     {\noindent\csuse{Xtxtbeforenotes@#1}}%
2458     {\expandafter\insert\csname#1footins\endcsname%
2459       \bgroup%
2460         \noindent%
2461         \csuse{\csuse{series@display#1}@begin@insert}{#1}%
2462         \strut\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}%
2463       \egroup%
2464     }%
2465   }%
2466 }%
2467 {}%
2468 }%
2469 %

```

XII Critical footnotes

The footnote macros are adapted from those in PLAIN T_EX, but they differ in these respects: the outer-level commands must add other commands to a list macro rather than doing insertions immediately; there are many separate levels of the footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

XII.1 Fonts

Before getting into the details of formatting the notes, we set up some font macros. It is the notes that present the greatest challenge for our font-handling mechanism, because we need to be able to take fragments of our main text and print them in different forms: it is common to reduce the size, for example, without otherwise changing the fonts used.

`\select@lemmfont` `\select@lemmfont` 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.

```

2470 \def\select@lemmfont#1|#2|#3|#4|#5|#6|#7|{\select@lemmfont#7|}
2471 \def\select@lemmfont#1/#2/#3/#4|{%
2472   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2473    \selectfont}
2474
2475 %
```

XII.2 Individual note options

`\footnoteoptions@` The `\footnoteoption@[side]{options}{value}` changes the value of on options of Xfootnote, to switch between true and false.

```

2476 \newcommand*{\footnoteoptions@}[3]{%
2477   \def\do##1{%
2478     \ifstrequal{#1}{L}{% In Leftside
2479       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@list%
2480         \global\advance\insert@count \@ne% Increment the left insert
counter.
2481       }%
2482     }%
2483     \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@listR%
2484       \global\advance\insert@countR \@ne% Increment the right insert
counter insert.
2485     }%
2486   }%
2487   \notblank{#2}{\docsvlist{#2}}}% Parsing all options
2488 }
2489 %
```

XII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua[®]TeX is used.

```

2490 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
2491   \ifstrequal{#1}{L}{%
2492     \xright@appenditem{{\csxdef{footnote@luatextextdir}{\the\textdir}}}\to\
inserts@list%Know the dir of lemma
2493     \global\advance\insert@count \@ne%
2494     \xright@appenditem{{\csxdef{footnote@luatexpardir}{\the\pardir}}}\to\
inserts@list%Know the dir of lemma
2495     \global\advance\insert@count \@ne%
2496   }%
2497   {%
2498     \xright@appenditem{{\csxdef{footnote@luatextextdir}{\the\textdir}}}\to\
inserts@listR%Know the dir of lemma
2499     \global\advance\insert@countR \@ne%
2500     \xright@appenditem{{\csxdef{footnote@luatexpardir}{\the\pardir}}}\to\
inserts@listR%Know the dir of lemma
2501     \global\advance\insert@countR \@ne%
2502   }%
2503 }
2504 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

2505 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2506   \ifstrequal{#1}{L}{%
2507     \if@RTL%
2508       \xright@appenditem{{\csxdef{footnote@dir}{@RTLtrue}}}\to\
inserts@list%Know the language used in the lemma
2509       \global\advance\insert@count \@ne%
2510     \else
2511       \xright@appenditem{{\csxdef{footnote@dir}{@RTLfalse}}}\to\
inserts@list%Know the language of lemma
2512       \global\advance\insert@count \@ne%
2513     \fi%
2514     \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}}}\
to\inserts@list%Know the language of lemma
2515     \global\advance\insert@count \@ne%
2516   }%
2517   {%
2518     \if@RTL
2519       \xright@appenditem{{\csxdef{footnote@dir}{@RTLtrue}}}\to\
inserts@listR%Know the language of lemma
2520       \global\advance\insert@countR \@ne%
2521     \else
2522       \xright@appenditem{{\csxdef{footnote@dir}{@RTLfalse}}}\to\
inserts@listR%Know the language of lemma

```

```

2523 \global\advance\insert@countR \@ne%
2524 \fi
2525 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\languagename}}}\
to\inserts@listR%Know the language of lemma
2526 \global\advance\insert@countR \@ne%
2527 }%
2528 }
2529 %

```

XII.4 General survey of the way we manage notes

The processing of each note is done by four principal macros: the `\vfootnote` macro takes the text of the footnote and does the `\insert`; it calls on the `\footfmt` macro to select the right fonts, print the line number and lemma, and do any other formatting needed for that individual note. Within the output routine, the two other macros, `\footstart` and `\footgroup`, are called; the first prints extra vertical space and a footnote rule, if desired; the second does any reformatting of the whole set of the footnotes in this series for this page—such as paragraphing or division into columns—and then sends them to the page.

These four macros, and the other macros and parameters shown here, are distinguished by the ‘series letter’ that indicates which set of the footnotes we are dealing with—A, B, C, D, or E. The series letter always precedes the string `foot` in macro and parameter names. Hence, for the A series, the four macros are called `\vAfootnote`, `\Afootfmt`, `\Afootstart`, and `\Afootgroup`.

These macros are changed depending of the footnotes arrangement: “normal”, “paragraphed”, “two columns” or “three columns”.

XII.5 General setup

`\footsplitskips` Some setup code that is common for a variety of the footnotes. The setup is for:

- `\interlinepenalty`.
- `\splittopskip` (skip before last part of notes that flow from one page to another).
- `\splitmaxdepth`.
- `\floatingpenalty`, that is penalty values being added when a long note flows from one page to another. Here, we let it to 0 when we are processing parallel pages in `eledpar`, in order to allow notes to flow from left to right pages and *vice-versa*. Otherwise, we let it to `\@MM`, which is the standard \TeX `\floatingpenalty`.

```

2530 \newcommand*{\footsplitskips}{%
2531 \interlinepenalty=\interfootnotelinepenalty
2532 \unless\ifl@dprintingpages%
2533 \floatingpenalty=\@MM%
2534 \fi%

```

```

2535 \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2536 \leftskip=\z@skip \rightskip=\z@skip}
2537
2538 %

```

\normalfootnoterule \normalfootnoterule is a standard footnote-rule macro, for use by a footstart macro: just the same as the PLAIN T_EX footnote rule.

```

2539 \let\normalfootnoterule=\footnoterule
2540 %

```

XII.6 Footnotes arrangement

XII.6.1 User level macro

\Xarrangement \Xarrangement[*s*]{*arrangement*} The command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

2541 \newcommand{\Xarrangement}[2][1,usedefault]{%
2542   \def\do##1{%
2543     \csname Xarrangement@#2\endcsname{##1}%
2544   }%
2545   \ifstrempy{#1}%
2546     {%
2547       \dolistloop{\@series}%
2548     }%
2549     {
2550       \docsvlist{#1}%
2551     }%
2552   }%
2553   %

```

XII.6.2 Normal footnote

\Xarrangement@normal We can now define all the parameters for the series of footnotes; initially they use the “normal” footnote formatting.

What we want to do here is to insert something like the following for each footnote series. (This is an example, not part of the actual reledmac code.)

```

\skip\Afootins=12pt plus5pt minus5pt
\count\Afootins=1000
\dimen\Afootins=0.8\vsiz
\let\VAfootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule

```


(Read *The TeXbook* in order to understand what are the counter, skip and dimen associated to an insertion.)

Instead of repeating ourselves, we define a `\Xarrangement@normal` macro that makes all these assignments for us, for any given series letter. This command is called when people use `\Xarrangement[⟨series⟩]{normal}`

Now we set up the `\Xarrangement@normal` macro itself. It takes one argument: the footnote series letter.

```

2554 \newcommand*{\Xarrangement@normal}[1]{%
2555   \csgdef{series@display#1}{normal}
2556   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
2557   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
2558   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
2559   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
2560   \expandafter\let\csname #1footnoterule\endcsname=%
2561                                     \normalfootnoterule
2562   \count\csname #1footins\endcsname=1000
2563   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2564   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2565   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2566   %

```

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.

```

2567   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2568   %

```

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).

```

2569   \ifnoledgroup@else%
2570     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2571     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2572     \count\csname mp#1footins\endcsname=1000
2573     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2574     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2575     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2576   \fi
2577 }
2578 %
2579 %

```

`\normalvfootnote` We now begin a series of commands that do 'normal' footnote formatting: a format much like that implemented in `PLAIN \TeX` , in which each footnote is a separate paragraph.

`\normalvfootnote` takes the series letter as `#1` and the entire text of the footnote is `#2`. It does the `\insert` for this note, calling on the `\footfmt` macro for this note series to format the text of the note.

```

2580 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2581   \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
      insertion is done later, in \add@Xgroupbyline.
2582   \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
2583 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
2584   \X@beforeinsertion{#1}%
2585   \insert\csname #1footins\endcsname{%
2586     \X@atbegininsertion{#1}%
2587     \normalvfootnote@inserted{#1}{#2}%
2588   }%
2589 }%
2590 }%
2591 %

```

\normalvfootnote@inserted The `\normalvfootnote@inserted` macro is expanded to the content to be add to a `\insert` for normal critical footnote.

```

2592 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
normalvfootnote@inserted}[2]{%
2593   \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
2594   \csuse{Xnotefontsize@#1}%
2595   \footsplitskips
2596   \ifl@dpairing\ifl@dpaging\else%
2597     \setXnoteswidthliketwocolumns@{#1}%
2598   \fi\fi%
2599   \setXnotespositionliketwocolumns@{#1}%
2600   \spaceskip=\z@skip \xspaceskip=\z@skip%
2601   \csname #1footfmt\endcsname #2{#1}%
2602 }%
2603 %

```

```

\X@beforeinsertion\newcommand{\X@beforeinsertion}[1]{%
2605   \if@ledgroup\else%
2606     \insert@Xtxtbeforenotes{#1}%
2607   \fi%
2608   \csuse{Xbeforeinserting@#1}%
2609 }%
2610 %

```

```

\X@atbegininsertion\newcommand{\X@atbegininsertion}[1]{%
2612   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2613 }%
2614 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote 2615 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
2616 \iftoggle{Xgroupbyline@#1}{%
2617 \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
2618 }%
2619 {%
2620 \global\setbox\@nameuse{mp#1footins}%
2621 \vbox{%
2622 \unvbox\@nameuse{mp#1footins}%
2623 \mpnormalvfootnote@inserted{#1}{#2}%
2624 }%
2625 }%
2626 }%
2627
2628 %

```

```

\mpnormalvfootnote@inserted 2629 \newcommand{\mpnormalvfootnote@inserted}[2]{%
2630 \noindent\csuse{Xbhooknote@#1}%
2631 \csuse{Xnotefontsize@#1}%
2632 \hsize\columnwidth%
2633 \@parboxrestore%
2634 \color@begingroup%
2635 \csname #1footfmt\endcsname #2{#1}\color@endgroup%
2636 }%
2637 %

```

\normalfootfmt `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 95), and the desired text, and output what’s to be printed. Argument #1 contains the line and page number information and lemma font specifier; #2 is the lemma; #3 is the note’s text. This version is very rudimentary—it uses `\printlines` to print just the range of line numbers, followed by a square bracket, the lemma, and the note text.

```

2638
2639
2640 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
2641 \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}{}%
2642 \hangindent=\csuse{Xhangindent@#4}%
2643 \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2644 \nottoggle{Xgroupbyline@#4}{\rule{z@}{\splittopskip}}{}%
2645 {\printlinefootnote{#1}{#4}}%
2646 \print@lemma{#1}{#2}{#4}%
2647 \csuse{Xwrapcontent@#4}{#3}%
2648 \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
2649 }%
2650 %

```

\normalfootstart `\normalfootstart` is a standard footnote-starting macro, called in the output routine whenever there are footnotes of this series to be printed: it skips a bit and then draws a rule.

Any `\footstart` macro must put onto the page something that takes up space exactly equal to the `\skip\Xfootins` value for the associated series of notes. `TEX` makes page computations based on that `\skip` value, and the output pages will suffer from spacing problems if what you add takes up a different amount of space.

But if the skip `\Xprenotes@` is greater than 0 pt, it is used instead of `\skip\footins` for the first printed series in one page.

The `\leftskip` and `\rightskip` values are both zeroed here. Similarly, these skips are cancelled in the `\vfootnote` macros for the various types of notes. Strictly speaking, this is necessary only if you are using paragraphed footnotes, but we have put it here and in the other `\vfootnote` macros too so that the behavior of `reledmac` in this respect is general across all footnote types. What this means is that any `\leftskip` and `\rightskip` you specify applies to the main text, but not the footnotes. The footnotes continue to be of width `\hsize`.

```
2651 \newcommand*{\normalfootstart}[1]{%
2652 %
```

The first series of notes printed in a page can have a specific skip before it. In order to insert this specific skip without overlap the bottom margin of the page, Maïeul Rouquette have defined an algorithm explained in XVIII p. 214. Here is part of this algorithm, when the block of notes are ready to be printed.

```
2653 \ifdimequal{0pt}{\Xprenotes@}{}%
2654 {%
2655 \iftoggle{Xprenotes@}{%
2656 \togglefalse{Xprenotes@}%
2657 \skip\csname #1footins\endcsname=%
2658 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
2659 }%
2660 }%
2661 }%
2662 \vskip\skip\csname #1footins\endcsname%
2663 %
```

And now, the problem of left and right skip for notes. Especially when using one feature of `reledpar` which allows to have the footnotes horizontal size as the size of columns printed by `\Columns`. Read XV p. 211 for the general description of the problem.

```
2664 \leftskip0pt \rightskip0pt
2665 \ifl@dpairing\else%
2666 \hsize=\old@hsize%
2667 \fi%
2668 \setXnoteswidthliketwocolumns@{#1}%
2669 \setXnotespositionliketwocolumns@{#1}%
2670 %
```

And now, print the footnote's rule to finish the footnote's introduction.

```
2671 \print@Xfootnoterule{#1}%
2672 }%
2673 %
```

`\normalfootgroup` `\normalfootgroup` is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

2674 \newcommand*{\normalfootgroup}[1]{%
2675   \csuse{Xbhookgroup@#1}%
2676   \unvbox\csname #1footins\endcsname%
2677   \hsize=\old@hsize%
2678 }%
2679
2680 %

```

`\mpnormalfootgroup` A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the `\Xfootgroup` and `\Xfootstarts` macros.

```

2681 \unless\ifnoledgroup@
2682 \newcommand*{\mpnormalfootgroup}[1]{%
2683   \vskip\skip\@nameuse{mp#1footins}
2684   \ifl@dpairing\ifparledgroup%
2685     \leavevmode\marks\parledgroup@{begin}%
2686     \marks\parledgroup@series{#1}%
2687     \marks\parledgroup@type{Xfootnote}%
2688   \fi\fi\normalcolor%
2689   \ifparledgroup%
2690     \ifl@dpairing%
2691     \else%
2692       \setXnoteswidthliketwocolumns@{#1}%
2693       \setXnotespositionliketwocolumns@{#1}%
2694       \print@Xfootnoterule{#1}%
2695     \fi%
2696   \else%
2697     \setXnoteswidthliketwocolumns@{#1}%
2698     \setXnotespositionliketwocolumns@{#1}%
2699     \print@Xfootnoterule{#1}%
2700   \fi%
2701   \setlength{\parindent}{0pt}
2702   \csuse{Xbhookgroup@#1}%
2703   \unvbox\csname mp#1footins\endcsname}}
2704 \fi
2705 %

```

XII.6.3 Paraphrased footnotes

The paraphrased-footnote option reformats all the footnotes of one series for a page into a single paragraph; this is especially appropriate when the notes are numerous and brief. The code is based on *The TeXbook*, pp. 398–400, with alterations for our environment. This algorithm uses a considerable amount of save-stack space: a \TeX of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

`\Xarrangement@paragraph` The `\Xarrangement@paragraph` macro sets up everything for one series of the footnotes so that they will be paraphrased; 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.

```

2706 \newcommand*{\Xarrangement@paragraph}[1]{%
2707   \csgdef{series@display#1}{paragraph}
2708   \expandafter\let\csname #1footstart\endcsname=\parafootstart
2709   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
2710   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
2711   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
2712   \count\csname #1footins\endcsname=1000
2713   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2714   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2715   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2716   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2717   \para@footsetup{#1}
2718   %

```

And the extra setup for minipages.

```

2719   \ifnoledgroup@else
2720     \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
2721     \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
2722     \count\csname mp#1footins\endcsname=1000
2723     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2724     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2725     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2726   \fi
2727 }
2728 %

```

`\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.

```

2729 \providecommand{\footfudgefiddle}{64}
2730 %

```

`\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`.

```

2731 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}

```

```

2732 \setXnoteswidthliketwocolumns@{#1}%
2733 \ifcsempy{Xwidth@#1}%
2734   {}%
2735   {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
2736 \dimen0=\baselineskip
2737 \multiply\dimen0 by 1024
2738 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
2739 \csxdef{#1footfudgefactor}{%
2740   \expandafter\strip@pt\dimen0 }}
2741
2742 %

```

\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.

```

2743 \newcommand*{\parafootstart}[1]{%
2744   \rightskip=0pt \leftskip=0pt%
2745   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2746   \ifdimequal{0pt}{Xprenotes@}{}%
2747     {%
2748       \iftoggle{Xprenotes@}{%
2749         \togglefalse{Xprenotes@}%
2750         \skip\csname #1footins\endcsname=%
2751         \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
2752       }%
2753     }%
2754   }%
2755   \vskip\skip\csname #1footins\endcsname%
2756   \setXnoteswidthliketwocolumns@{#1}%
2757   \setXnotespositionliketwocolumns@{#1}%
2758   \print@Xfootnoterule{#1}%
2759   \let\bidirTL@everypar\@empty%
2760   \noindent\leavevmode}
2761 %

```

\paravfootnote \paravfootnote is a version of the \vfootnote command that is used for paragraphed notes. It gets appended to the \inserts@list list by an outer-level footnote command like \Afootnote. The first argument is the note series letter; the second is the full text of the printed note itself, including line numbers, lemmata, and footnote text.

The initial model for this insertion is, of course, the \insert\footins definition in *The TeXbook*, p. 398. There, the footnotes are first collected up in hboxes, and these hboxes are later unpacked and stuck together into a paragraph.

However, Michael Downes has pointed out that because text in `hboxes` gets typeset in restricted horizontal mode, there are some undesirable side-effects if you later want to break such text across lines. In restricted horizontal mode, where \TeX does not expect to have to break lines, it does not insert certain items like `\discretionary`s. If you later unbox these `hboxes` and stick them together, as the *TeXbook* macros do to make these footnotes, you lose the ability to hyphenate after an explicit hyphen. This can lead to overfull `hboxes` when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.²⁸

Wayne Sullivan pointed out to us another subtle problem that arises from the same cause: \TeX also leaves the `\language` whatsit nodes out of the horizontal list.²⁹ So changes from one language to another will not invoke the proper hyphenation rules in such footnotes. Since critical editions often do deal with several languages, especially in a footnotes, we really ought to get this bit of code right.

To get around these problems, Wayne suggested emendations to the *TeXbook* versions of these macros which are broadly the same as those described by Michael: the central idea (also suggested by Donald Knuth in a letter to Michael) is to avoid collecting the text in an `hbox` in the first place, but instead to collect it in a `vbox` whose width is (virtually) infinite. The text is therefore typeset in unrestricted horizontal mode, as a paragraph consisting of a single long line. Later, there is an extra level of unboxing to be done: we have to unpack the `vbox`, as well as the `hboxes` inside it, but that is not too hard. For details, we refer you to Michael's article, where the issues are clearly explained.³⁰ Michael's unboxing macro is called `\Xunvxh`: unvbox, extract the last line, and unhbox it.

Doing things this way has an important consequence: as Michael pointed out, you really can't put an explicit line-break into a note built in a `vbox` the way we are doing.³¹ In other words, be very careful not to use `\break`, or `\penalty-10000`, or any equivalent inside your para-footnote. If you do, most of the note will probably disappear. You *are* allowed to make strong suggestions; in fact `\penalty-9999` will be quite okay. Just do not make the break mandatory. We have not applied any of Michael's solutions here, since we feel that the problem is exiguous, and `reledmac` is quite baroque enough already. If you think you are having this problem, look up Michael's solutions.

One more thing; we set `\leftskip` and `\rightskip` to zero. This has the effect of neutralizing any such skips which may apply to the main text (cf. XII.6.2 p. 164 above). We need to do this, since `\footfudgefactor` is calculated on the assumption that the notes are `\hsize` wide.

So, finally, here is the modified foot-paragraph code, which sets the footnote in vertical mode so that language and discretionary nodes are included.

```
2762 \newcommand*{\paravfootnote}[2]{%
2763   \csuse{Xbeforeinserting@#1}%
2764   \insert\csname #1footins\endcsname
2765   \bgroup
```

²⁸Michael Downes, 'Line Breaking in `\unhboxed` Text', *TUGboat* **11** (1990), pp. 605–612.

²⁹See *The TeXbook*, p. 455 (editions after January 1990).

³⁰Wayne supplied his own macros to do this, but since they were almost identical to Michael's, Peter Wilson have used the latter's `\Xunvxh` macro since it is publicly documented.

³¹'Line Breaking', p. 610.


```

2766 \csuse{Xnotefontsize@#1}
2767 \footsplitskips
2768 \setbox0=\vbox{\hsize=\maxdimen%
2769   \let\bid@RTL@everypar\@empty%
2770   \insert@Xtxtbeforenotes{#1}%
2771   \noindent\csuse{Xbhooknote@#1}%
2772   \csname #1footfmt\endcsname #2{#1}}%
2773 \setbox0=\hbox{\Xunvxh{0}{#1}}%
2774 \dp0=0pt
2775 \ht0=\csname #1footfudgefactor\endcsname\wd0
2776 %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

2777 \if@RTL\noindent \leavevmode\fi\box0%
2778 \penalty0
2779 \egroup}
2780
2781 %

```

The final penalty of 0 was added here at Wayne's suggestion to avoid a weird page-breaking problem, which occurs on those occasions when \TeX attempts to split foot paragraphs. After trying out such a split (see *The TeXbook*, p. 124), \TeX inserts a penalty of -10000 here, which nearly always forces the break at the end of the whole footnote paragraph (since individual notes can't be split) even when this leads to an overfull vbox. The change above results in a penalty of 0 instead which allows, but does not force, such breaks. This penalty of 0 is later removed, after page breaks have been decided, by the `\unpenalty` macro in `\makehboxofhboxes`. So it does not affect how the footnote paragraphs are typeset (the notes still have a penalty of -10 between them, which is added by `\parafootfmt`).

`\mpparavfootnote` This version is for minipages.

```

2782 \newcommand*{\mpparavfootnote}[2]{%
2783   \global\setbox\@nameuse{mp#1footins}\vbox{%
2784     \unvbox\@nameuse{mp#1footins}%
2785     \csuse{Xnotefontsize@#1}
2786     \footsplitskips
2787     \setbox0=\vbox{\hsize=\maxdimen%
2788       \let\bid@RTL@everypar\@empty%
2789       \insert@Xtxtbeforenotes{#1}%
2790       \noindent\color@begingroup%
2791       \csuse{Xbhooknote@#1}%
2792       \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
2793     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2794     \dp0=\z@
2795     \ht0=\csname #1footfudgefactor\endcsname\wd0
2796     \box0
2797     \penalty0
2798   }}

```

2799

2800 %

\Xunvxh Here is (modified) Michael’s definition of `\unvxh`, used above. Michael’s macro also takes care to remove some unwanted penalties and glue that \TeX automatically attaches to the end of paragraphs. When \TeX finishes a paragraph, it throws away any remaining glue, and then tacks on the following items: a `\penalty` of 10000, a `\parfillskip` and a `\rightskip` (*The TeXbook*, pp. 99–100). `\unvxh` cancels these unwanted paragraph-final items using `\unskip` and `\unpenalty`.

2801 \newcommand*{\Xunvxh}[2]{%

2802 \setbox0=\vbox{\unvbox#1%

2803 \global\setbox1=\lastbox}%

2804 \unhbox1

2805 \unskip % remove \rightskip,

2806 \unskip % remove \parfillskip,

2807 \unpenalty % remove \penalty of 10000,

2808 \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes

2809

2810 %

\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).

2811 \newcommand*{\parafootfmt}[4]{%

2812 \Xinsertparafootsep{#4}%

2813 \ledsetnormalparstuff@common%

2814 \printlinefootnote{#1}{#4}%

2815 \print@lemma{#1}{#2}{#4}%

2816 \csuse{Xwrapcontent@#4}{#3}%

2817 \penalty-10 }

2818 %

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.

2819 \newcommand*{\parafootgroup}[1]{%

2820 \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%

2821 \unvbox\csname #1footins\endcsname

```

2822 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2823 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2824 \makehboxofhboxes
2825 \setbox0=\hbox{\unhbox0 \removehboxes}%
2826 \csuse{Xbhookgroup@#1}%
2827 \csuse{Xnotefontsize@#1}%
2828 \unhbox0\par%
2829 \global\hsize=\old@hsize%
2830 }%
2831
2832 %

```

`\mpparafootgroup` The minipage version.

```

2833 \newcommand*{\mpparafootgroup}[1]{%
2834   \setXnoteswidthliketwocolumns@{#1}%
2835   \vskip\skip\@nameuse{mp#1footins}
2836   \ifl@dpairing\ifparledgroup%
2837     \leavevmode\marks\parledgroup@{begin}%
2838     \marks\parledgroup@series{#1}%
2839     \marks\parledgroup@type{Xfootnote}%
2840   \fi\fi\normalcolor
2841   \ifparledgroup%
2842     \ifl@dpairing%
2843     \else%
2844       \setXnoteswidthliketwocolumns@{#1}%
2845       \setXnotespositionliketwocolumns@{#1}%
2846       \print@Xfootnoterule{#1}%
2847     \fi%
2848   \else%
2849     \setXnoteswidthliketwocolumns@{#1}%
2850     \setXnotespositionliketwocolumns@{#1}%
2851     \print@Xfootnoterule{#1}%
2852   \fi%
2853   \unvbox\csname mp#1footins\endcsname
2854   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2855   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2856   \makehboxofhboxes
2857   \setbox0=\hbox{\unhbox0 \removehboxes}%
2858   \csuse{Xbhookgroup@#1}%
2859   \csuse{Xnotefontsize@#1}%
2860   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2861   \unhbox0\par}}
2862
2863 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes\newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes\loop
2866   \unpenalty
2867   \setbox2=\lastbox
2868   \ifhbox2
2869     \setbox0=\hbox{\box2\unhbox0}%
2870   \repeat}
2871
2872 \newcommand*{\removehboxes}{\setbox0=\lastbox
2873   \ifhbox0{\removehboxes}\unhbox0 \fi}
2874
2875 %

```

Insertion of the footnotes separator The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num\newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep\ifledRcol{%
2878   \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
2879     {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2880     {\ifnumequal{\csuse{prevline#1}}{\line@numR}%
2881     {\ifcsempy{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}}%
2882     {\csuse{Xparafootsep@#1}}}%
2883   }%
2884   {\csuse{Xparafootsep@#1}}}%
2885 }%
2886 {}%
2887 \global\csname #1prevpage@numR\endcsname=\page@numR%
2888 \else%
2889   \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
2890     {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2891     {\ifnumequal{\csuse{prevline#1}}{\line@num}%
2892     {\ifcsempy{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}}%
2893     {\csuse{Xparafootsep@#1}}}%
2894   }%
2895   {\csuse{Xparafootsep@#1}}}%
2896 }%
2897 {}%
2898 \global\csname #1prevpage@num\endcsname=\page@num%
2899 \fi%
2900 }
2901 %

```

XII.6.4 Columnar footnotes

Common tools

`\rigidbalance` We will now define macros for three-column notes and two-column notes. Both sets of macros will use `\rigidbalance`, which splits a box (#1) into into a number (#2) of

`\rigidbalanceX`

`\Xrigidbalance`

`\dosplits`

`\splitoff`

`\@h`

`\@k`

columns, each with a space (#3) between the top baseline and the top of the \vbox. The \rigidbalance macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they do not depend on white space. Note also the extra unboxing in \splittoff, which allows the new \vbox to have its natural height as it goes into the alignment.

The \TeX \line macro has no relationship to the TeX \line. The \TeX equivalent is \@oline.

We do not call directly \rigidbalance, but we call \Xrigidbalance for critical notes and \rigidbalanceX for familiar notes. Both of them call \rigidbalance.

```

2902 \newcount\@k \newdimen\@h
2903 \newcommand*\Xrigidbalance}[3]{%
2904   \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
2905   \rigidbalance{#1}{#2}{#3}%
2906 }%
2907
2908 \newcommand*\rigidbalanceX}[3]{%
2909   \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
2910   \rigidbalance{#1}{#2}{#3}%
2911 }%
2912
2913 \newcommand*\rigidbalance}[3]{%
2914   \setbox0=\box#1 \@k=#2 \@h=#3%
2915   \@oline{\splittopskip=\@h \vbadness=\@M \hfilneg
2916   \valign{##\vfil\cr\dosplits}}}%
2917
2918 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splittoff
2919   \global\advance\@k-1\cr\dosplits\fi}
2920
2921 \newcommand*\splittoff{\dimen0=\ht0
2922   \divide\dimen0 by\@k \advance\dimen0 by\@h
2923   \setbox2 \vsplit0 to \dimen0
2924   \unvbox2 }
2925
2926 %

```

Three columns

```

\Xarrangement@threecol 2927 \newcommand*\Xarrangement@threecol}[1]{%
2928   \csgdef{series@display#1}{threecol}
2929   \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
2930   \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
2931   \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
2932   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2933   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2934   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2935   \threecolfootsetup{#1}
2936 %

```

The additional setup for minipages.

```

2937 \ifnoledgroup@else
2938   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2939   \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
2940   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2941   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2942   \mpthreecolfootsetup{#1}
2943 \fi
2944 }
2945
2946 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XII.6.2 p. 163 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.

```

2947 \newcommand*{\threecolfootsetup}[1]{%
2948   \count\csname #1footins\endcsname 333
2949   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
2950   \multiply\dimen\csname #1footins\endcsname \thr@@}
2951 %

```

`\mpthreecolfootsetup` The setup for minipages.

```

2952 \newcommand*{\mpthreecolfootsetup}[1]{%
2953   \count\csname mp#1footins\endcsname 333
2954   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
2955
2956 %

```

`\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).

```

2957 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootnote}[2]{%
2958 \iftoggle{Xgroupbyline@#1}{%
2959 \prepare@Xgroupbyline{#1}{#2}{\threecolfootnote@inserted}%
2960 }%
2961 {%
2962 \X@beforeinsertion{#1}%
2963 \insert\csname #1footins\endcsname{%
2964 \threecolfootnote@inserted{#1}{#2}%
2965 }%
2966 }%
2967 }%
2968 %

```

```

\threecolfootnote@inserted \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
threecolfootnote@inserted}[2]{%
2970 \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2971 \noindent\csuse{Xhooknote@#1}%
2972 \csuse{Xnotefontsize@#1}%
2973 \footssplitsskip%
2974 \csname #1footfmt\endcsname #2{#1}%
2975 }%
2976 %

```

`\threecolfootfmt` `\threecolfootfmt` is the command that formats one note. The arguments are #1 the line numbers, #2 the lemma and #4 the text of the `-footnote` command #4 optional (for backward compatibility): the series.

```

2977 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
2978 \threecol@begin@insert{#4}%
2979 \hspace{\parindent}%
2980 \printlinefootnote{#1}{#4}%
2981 \print@lemma{#1}{#2}{#4}%
2982 \csuse{Xwrapcontent@#4}{#3}%
2983 \nottoggle{Xgroupbyline@#4}%
2984 {\strut\par\allowbreak}%
2985 {}%
2986 }%
2987 %

```

`\threecol@begin@insert` The `\threecol@begin@insert` contains code used at the beginning of any `\insert` for critical footnotes in three columns. It is used both by `\threecolfootfmt` and by `\insert@Xtxtbeforenotes`.

```

2988 \newcommand{\threecol@begin@insert}[1]{%
2989   \normal@pars%
2990   \nottoggle{Xgroupbyline@#1}%
2991   {\hspace \csuse{Xhsizethreecol@#1}}%
2992   {}%
2993   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2994   \tolerance=5000%
2995   \hangindent=\csuse{Xhangindent@#1}%
2996   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
2997   \@tempdima=\parindent%
2998   \csuse{Xcolalign@#1}%
2999   \parindent=\@tempdima%
3000   \strut%
3001 }%
3002 %

```

\threecolfootgroup And here is the footgroup macro that is called within the output routine to regroup the notes into three columns. Once again, the call to `\Xnotefontsize@{s}` is there to ensure that it is the right `\splittopskip`—the one used in footnotes—which is used to provide the third argument for `\rigidbalance`. This third argument (`\@h`) is the topskip for the box containing the text of the footnotes, and does the job of making sure the top lines of the columns line up horizontally. In *The TeXbook*, p. 398, Donald Knuth suggests retrieving the output of `\rigidbalance`, putting it back into the insertion box, and then printing the box. Here, we just print the `\line` which comes out of `\rigidbalance` directly, without any re-boxing.

```

3003 \newcommand*\threecolfootgroup[1]{%
3004   \csuse{Xhookgroup@#1}\par%
3005   \splittopskip=\ht\strutbox
3006   \expandafter
3007   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
3008 %

```

\mpthreecolfootgroup The setup for minipages.

```

3009 \newcommand*\mpthreecolfootgroup[1]{%
3010   \vskip\skip\@nameuse{mp#1footins}
3011   \ifl@dpairing\ifparledgroup%
3012     \leavevmode\marks\parledgroup@{begin}%
3013     \marks\parledgroup@series{#1}%
3014     \marks\parledgroup@type{Xfootnote}%
3015   \fi\fi\normalcolor
3016   \ifparledgroup%
3017     \ifl@dpairing%
3018     \else%
3019       \setXnoteswidthliketwocolumns@{#1}%
3020       \setXnotespositionliketwocolumns@{#1}%
3021       \print@Xfootnoterule{#1}%
3022     \fi%

```



```

3023 \else%
3024 \setXnoteswidthliketwocolumns@{#1}%
3025 \setXnotespositionliketwocolumns@{#1}%
3026 \print@Xfootnoterule{#1}%
3027 \fi%
3028 \csuse{Xbhookgroup@#1}\par%
3029 \splittopskip=\ht\strutbox
3030 \expandafter
3031 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3032
3033 %

```

Two columns

```

\Xarrangement@twocol 3034 \newcommand*{\Xarrangement@twocol}[1]{%
3035 \csgdef{series@display#1}{twocol}
3036 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3037 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3038 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3039 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3040 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3041 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3042 \twocolfootsetup{#1}
3043 %

```

The additional setup for minipages.

```

3044 \ifnoledgroup@else
3045 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3046 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
3047 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3048 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3049 \mptwocolfootsetup{#1}
3050 \fi
3051 }
3052
3053 %

```

`\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
3054 \newcommand*{\twocolfootsetup}[1]{%
3055 \count\csname #1footins\endcsname 500
3056 \csxdef{default@#1footins}{500}%
3057 \multiply\dimen\csname #1footins\endcsname \tw@}
3058 %

```

```

3059 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote}[2]{%
3060   \iftoggle{Xgroupbyline@#1}{%
3061     \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3062   }{%
3063     \X@beforeinsertion{#1}%
3064     \insert\csname #1footins\endcsname{%
3065       \twocolvfootnote@inserted{#1}{#2}%
3066     }%
3067   }%
3068 }%
3069 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote@inserted}[2]{%
3070   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3071   \noindent\csuse{Xhooknote@#1}%
3072   \csuse{Xnotefontsize@#1}%
3073   \footsplitskips%
3074   \csname #1footfmt\endcsname #2{#1}%
3075 }%
3076 %

3077 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmt}[4]{% 4th
   arg is optional, for backward compatibility
3078   \twocol@begin@insert{#4}%
3079   \hspace{\parindent}%
3080   \printlinefootnote{#1}{#4}%
3081   \print@lemma{#1}{#2}{#4}%
3082   \csuse{Xwrapcontent@#4}{#3}%
3083   \nottoggle{Xgroupbyline@#4}%
3084     {\strut\par\allowbreak}%
3085     {}%
3086 }%
3087 %

3088 \newcommand{\twocol@begin@insert}[1]{%
3089   \normal@pars%
3090   \hsize \csuse{Xhsizetwocol@#1}%
3091   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3092   \tolerance=5000%
3093   \hangindent=\csuse{Xhangindent@#1}%
3094   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3095   \@tempdima=\parindent%
3096   \csuse{Xcolalign@#1}%
3097   \parindent=\@tempdima%
3098   \strut%
3099 }%
3100
3101 \newcommand*{\twocolfootgroup}[1]{%
3102   \csuse{Xhookgroup@#1}\par%
3103   \splittopskip=\ht\strutbox
3104   \expandafter

```

```

3105 \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
3106
3107 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3108 \newcommand*{\mptwocolfootsetup}[1]{%
3109   \count\csname mp#1footins\endcsname 500
3110   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3111 %

3112 \newcommand*{\mptwocolfootgroup}[1]{%
3113   \vskip\skip\@nameuse{mp#1footins}
3114   \ifl@dpairing\ifparledgroup%
3115     \leavevmode\marks\parledgroup@{begin}%
3116     \marks\parledgroup@series{#1}%
3117     \marks\parledgroup@type{Xfootnote}%
3118     \fi\fi\normalcolor
3119   \ifparledgroup%
3120     \ifl@dpairing%
3121     \else%
3122       \setXnoteswidthliketwocolumns@{#1}%
3123       \setXnotespositionliketwocolumns@{#1}%
3124       \print@Xfootnoterule{#1}%
3125     \fi%
3126   \else%
3127     \setXnoteswidthliketwocolumns@{#1}%
3128     \setXnotespositionliketwocolumns@{#1}%
3129     \print@Xfootnoterule{#1}%
3130   \fi%
3131   \csuse{Xbhookgroup@#1}\par%
3132   \splittopskip=\ht\strutbox
3133   \expandafter
3134   \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3135
3136 %

```

XII.7 Critical notes presentation

Here, we define some commons macro which are used in order to print a critical notes, that is a note with 1) line number 2) lemma 3) lemma separator 4) text associated to the lemma.

XII.7.1 Font tools

`\endashchar` The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals does not contain an en-dash or square brackets, and its period and comma are in odd locations.

`\fullstop`

`\rbracket`

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 `\Xlinangeseparator`, `\Xendlinangeseparator`, `\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.

```

3137 \def\endashchar{\textnormal{--}}
3138
3139 \newcommand*\fullstop{\textnormal{.}}
3140 \def\Xsublinesep@side{\fullstop}
3141
3142 \newcommand*\rbracket{\textnormal{%
3143   \csuse{text\csuse{footnote@lang}}{%
3144     \ifluatex%
3145       \ifdefstring{footnote@luatextextdir}{TRT}{\thinspace[]\thinspace
3146     }}%
3147     \else%
3148     \thinspace}%
3149     \fi}%
3150   }}%
3151 }
3152 %

```

XII.7.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

3153 \newcommand{\printpstart}[0]{%
3154   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3155     l@dprintingcolumns}}{%
3156     \ifledRcol%
3157       \thepstartR%
3158     \else%
3159       \thepstartL%
3160     \fi%
3161   }{%
3162     \thepstart%
3163   }}%
3164 %

```

XII.7.3 Lemma printing

\print@lemma \print@lemma is called inside critical footnotes to print the lemma and the lemma separator (#1: line number and font information, #2: lemma, #3: series).

```

3165 %
3166 \newcommand{\print@lemma}[3]{%
3167   \bgroup%
3168   \nottoggle{Xlemmadisablefontselection@#3}%
3169   {\select@lemmafnt#1|}%
3170   }%
3171   \bgroup%
3172   \csuse{Xlemmafnt@#3}%Deprecated
3173   \csuse{Xwraplemma@#3}{#2}%
3174   \egroup%
3175   \egroup%
3176   \iftoggle{nosep@}{%
3177     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3178     \relax%
3179   }%
3180   {\ifcsemt{Xlemmaseparator@#3}%
3181     {%
3182       \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3183       \relax%
3184     }%
3185     {%
3186       \nobreak%
3187       \hskip\csuse{Xbeforelemmaseparator@#3}%
3188       \csuse{Xlemmaseparator@#3}%
3189       \hskip\csuse{Xafterlemmaseparator@#3}%
3190       \relax%
3191     }%
3192   }%
3193 }%
3194 %

```

XII.7.4 Line number printing

\printlinefootnote The \printlinefootnote macro is called in each \<type>footfmt command. It controls whether the line number is printed or not, according to the series options. Its first argument is the information about lines; its second is the series of the footnote. The printing of the line number is shared in \printlinefootnotenumbers.

```

3195 \newcommand{\printlinefootnote}[2]{%
3196   \l@dp@rsefootspec#1|}%
3197   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3198     \edef\lineinfo@{\l@dparsestartline - \l@dparsestartsub - \
3199     \l@dparseendline - \l@dparseendsub}%
3200   }%
3201   {%

```

```

3201 \edef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3202 }%
3203 \iftoggle{nonum@}{%Try if the line number must printed for this specific
not (by default, yes)
3204 \hspace{\csuse{Xinplaceofnumber@#2}}%
3205 }%
3206 {%
3207 {%
3208 \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
default, yes)
3209 {%
3210 \hspace{\csuse{Xinplaceofnumber@#2}}%
3211 }%
3212 {%
3213 {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
3214 {%
3215 \ifcsdef{prevline#2}%
3216 {%Be sure the \prevline exists.
3217 \ifcsequal{prevline#2}{\lineinfo@}%Try it
3218 {%
3219 \ifcseempty{Xsymlinenum@#2}%
3220 {%
3221 \hspace{\csuse{Xinplaceofnumber@#2}}%
3222 }%
3223 {\printsymlinefootnotearea{#2}}%
3224 }%
3225 {%
3226 \printlinefootnotearea{#1}{#2}%
3227 }%
3228 }%
3229 {%
3230 \printlinefootnotearea{#1}{#2}%
3231 }%
3232 }%
3233 {%
3234 \printlinefootnotearea{#1}{#2}%
3235 }%
3236 \csxdef{prevline#2}{\lineinfo@}%
3237 }%
3238 }%
3239 }%
3240 }%
3241 }
3242 %

```

\printsymlinefootnotearea This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

3243 \newcommand{\printsymlinefootnotearea}[1]{%

```

```

3244 \hspace{\csuse{Xbeforesymlinewidth@#1}}%
3245 \csuse{Xnotenumfont@#1}%
3246 \ifdimequal{\csuse{Xboxsymlinewidth@#1}}{\z@}%
3247   {\csuse{Xsymlinewidth@#1}}%
3248   {\hbox to \csuse{Xboxsymlinewidth@#1}%
3249     {\csuse{Xsymlinewidth@#1}\hfill}%
3250   }%
3251 \hspace{\csuse{Xaftersymlinewidth@#1}}%
3252 }%
3253 %

```

\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.)

```

3254 \newcommand{\printlinefootnotearea}[2]{%
3255   \printXbeforenumber{#2}%
3256   \csuse{Xnotenumfont@#2}%
3257   \boxfootnotenumbers{#1}{#2}%
3258   \printXafternumber{#2}%
3259 }%
3260 %

```

\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.

```

3261 \newcommand{\boxfootnotenumbers}[2]{%
3262   \ifdimequal{\csuse{Xboxlinewidth@#2}}{\Opt}{%
3263     \printlinefootnotenumbers{#1}{#2}%
3264   }%
3265   {%
3266     \hbox to \csuse{Xboxlinewidth@#2}%
3267     {%
3268       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3269       \printlinefootnotenumbers{#1}{#2}%
3270       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3271     }%
3272   }%
3273 }%
3274 %

```

\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.

```

3275 \newcommand{\printlinefootnotenumbers}[2]{%
3276   \xdef\@currentseries{#2}%

```

```

3277 \ifboolexpr{%
3278   (togl{Xpstart@#2} and bool{numberpstart})%
3279   or togl{Xpstarteverytime@#2}}%
3280 {\printpstart}{}%
3281 \iftoggle{Xstanza@#2}{%
3282   \ifnumberstanza%
3283   \printstanza%
3284   \csuse{Xstanzaseparator@#2}%
3285   \fi%
3286 }{%
3287 \iftoggle{Xonlypstart@#2}{%
3288   \csuse{Xtxtbeforenumber@#2}%
3289   \printlines#1|\ifledRcol@{\@Rlineflag\fi}}%
3290 }%
3291 %

```

\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.)

```

3292 \newcommand{\printXbeforenumber}[1]{%
3293   \hspace{\csuse{Xbeforenumber@#1}}%
3294 }%
3295 %

```

\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

```

3296 \newcommand{\printXafternumber}[1]{%
3297   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3298   \hspace{\csuse{Xafternumber@#1}}%
3299 }%
3300 %

```

If we have decided to print the line number in a specific notes, the \printlines macro prints the line numbers for a note—which, in the general case, is a rather complicated task. The seven parameters of the argument are the line numbers as stored in \l@d@nums, in the form described on V.9 p. 95: the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma.

edmac' creator have defined six boolean in order to know which component of line number description we have to print:

- \ifl@d@pnum for page numbers;
- \ifl@d@ssub for starting sub-line;
- \ifl@d@elin for ending line;
- \ifl@d@esl for ending sub-line; and

- `\ifl@d@dash` for the dash between the starting and ending groups.

There is no boolean for the line number because it is always printed.

Maïeul Rouquette has added `\ifl@d@Xtwolines` and `\ifl@d@Xmorethantwolines` to print a symbol which stands for “and subsequent” when there are two, three or more lines.

```

\ifl@d@pnum01 \newif\ifl@d@pnum
\ifl@d@ssub02 \newif\ifl@d@ssub
\ifl@d@elin03 \newif\ifl@d@elin
\ifl@d@esl04 \newif\ifl@d@esl
\ifl@d@dash05 \newif\ifl@d@dash
\ifl@d@Xtwolines3306 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines3307 \newif\ifl@d@Xmorethantwolines%
3308 %

```

```

\l@dp@rsefootspec \l@dp@rsefootspec{<spec>}{<lemma>}{<text>} parses a footnote specification. <lemma>
\l@dp@rsefootspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@dp@rsefootspec ber and lemma font specifier in \l@d@nums style format. The real work is done by
\l@dp@rsefootspec \l@dp@rsefootspec which defines macros holding the numeric values. In many cases,
\l@dp@rsefootspec this last command is called directly. Just a reminder of the arguments:
\l@dp@rsefootspec \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dp@rsefootspec \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@dp@rsefootspec
3309 \newcommand*{\l@dp@rsefootspec}[3]{\l@dp@rsefootspec#1|}
3310 \def\l@dp@rsefootspec#1|#2|#3|#4|#5|#6|#7|{%
3311 \gdef\l@dp@rsefootspec#1|{%
3312 \gdef\l@dp@rsefootspec#2|{%
3313 \gdef\l@dp@rsefootspec#3|{%
3314 \gdef\l@dp@rsefootspec#4|{%
3315 \gdef\l@dp@rsefootspec#5|{%
3316 \gdef\l@dp@rsefootspec#6|{%
3317 }
3318 %

```

Initialise the several number value macros.

```

3319 \def\l@dp@rsefootspec{0}%
3320 \def\l@dp@rsefootspec{0}%
3321 \def\l@dp@rsefootspec{0}%
3322 \def\l@dp@rsefootspec{0}%
3323 \def\l@dp@rsefootspec{0}%
3324 \def\l@dp@rsefootspec{0}%
3325
3326 %

```

`\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`.

```

3327 \newcommand*{\setprintlines}[6]{%
3328   \l@d@pnumfalse \l@d@dashfalse
3329   %

```

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

```

3330   \ifbypage@
3331     \ifnum#4=#1 \else
3332       \l@d@pnumtrue
3333       \l@d@dashtrue
3334     \fi
3335   \fi
3336   %

```

We print the ending line number if: (1) we are printing the ending page number, or (2) it is different from the starting line number.

```

3337   \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
3338   \ifnum#2=#5 \else
3339     \l@d@elintrue
3340     \l@d@dashtrue
3341   \fi
3342   %

```

We print the starting sub-line if it is nonzero.

```

3343   \l@d@ssubfalse
3344   \ifnum#3=0 \else
3345     \l@d@ssubtrue
3346   \fi
3347   %

```

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.

```

3348   \l@d@eslfalse
3349   \ifnum#6=0 \else
3350     \ifnum#6=#3
3351       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3352     \else
3353       \l@d@esltrue
3354       \l@d@dashtrue
3355     \fi
3356   \fi%
3357   %

```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

3358   \ifl@d@dash%
3359   \ifboolexpr{togl{fulllines@} or test{\ifcempty{Xtwolines@}\
@currentseries}}}%
3360   {}%

```

```

3361 {%
3362 \setistwofollowinglines{#1}{#2}{#4}{#5}%
3363 \ifboolexpr{%
3364   (%
3365     togl {Xtwolinesbutnotmore@\@currentseries}%
3366     and not%
3367     (%
3368       bool {istwofollowinglines@}%
3369     )%
3370   )%
3371 or%
3372 (%
3373   (not test{ifnumequal{#1}{#4}})%
3374   and togl{Xtwolinesonlyinsamepage@\@currentseries}%
3375 )%
3376 }%
3377 {}%
3378 {%
3379 \l@d@dashfalse%
3380 \l@d@Xtwolinestrue%
3381 \l@d@elinfalse%
3382 \l@d@eslfalse%
3383 \ifcempty{Xmorethantwolines@\@currentseries}%
3384 {}%
3385 {\ifistwofollowinglines@\else%
3386   \l@d@Xmorethantwolinestrue%
3387   \fi%
3388 }%
3389 }%
3390 }%
3391 \fi%
3392 %

```

End of \setprintlines.

```

3393 }%
3394 %

```

\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.

```

3395 \newif\ifistwofollowinglines%
3396 \newcommand{\setistwofollowinglines}[4]{%
3397     \ifcsdef{lastlinenumberon\#1}%
3398         {\numdef{\tmp}{\csuse{lastlinenumberon\#1}}}%
3399         {\numdef{\tmp}{0}}%
3400     \istwofollowinglines@false%
3401     \ifnumequal{\#4-\#2}{1}%
3402     {\istwofollowinglines@true}%
3403     {\ifbypage%
3404         \ifnumequal{\#3-\#1}{1}%
3405         {%
3406             \ifnumequal{\#2}{\tmp}%
3407             {\ifnumequal{\#4}{1}{\istwofollowinglines@true}{}}%
3408             {}%
3409         }%
3410         {}%
3411     \fi%
3412 }%
3413 }%
3414 %

```

`\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

```

3415 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3416     \begingroup%
3417     %

```

If we use Lua_T_EX, ensure we use good text's direction.

```

3418     \ifluatex%
3419         \edef\@tmp{\the\textdir}%
3420         \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
3421         \fi%
3422     %

```

Decide which part of line number components we will print.

```

3423 \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3424 %

```

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.

```

3425 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3426 {\bgroup}%
3427 {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
3428 \ifl@d@pnum%
3429 \wrap@edcrossref{\@this@crossref@start}{#1}%
3430 \csuse{Xpagelinesep@\@currentseries}%
3431 \fi%
3432 \wrap@edcrossref{\@this@crossref@start}{%
3433 \linenumrep{#2}%
3434 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3435 }%
3436 \ifl@d@ssub%
3437 \csuse{Xsublinesep@\@currentseries}%
3438 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3439 \fi
3440 \egroup%
3441 %

```

Then print the dash + end line number, or the range symbol.

```

3442 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3443 {\bgroup}%
3444 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
3445 \ifl@d@Xtwolines%
3446 \ifl@d@Xmorethantwolines%
3447 \csuse{Xmorethantwolines@\@currentseries}%
3448 \else%
3449 \csuse{Xtwolines@\@currentseries}%
3450 \fi%
3451 \else%
3452 \ifl@d@dash%
3453 \ifdefined\linangesep%
3454 \linangesep%
3455 \else%
3456 \csuse{Xlinangeseparator@\@currentseries}%
3457 \fi%
3458 \fi%
3459 \ifl@d@pnum%
3460 \wrap@edcrossref{\@this@crossref@end}{#4}%
3461 \csuse{Xpagelinesep@\@currentseries}%
3462 \fi%
3463 \ifl@d@elin%

```

```

3464 \wrap@edcrossref{\@this@crossref@end}{%
3465 \linenumrep{#5}%
3466 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3467 }%
3468 \fi%
3469 \ifl@d@esl%
3470 \ifl@d@elin%
3471 \csuse{Xsublinesep@\@currentseries}%
3472 \fi%
3473 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3474 \fi%
3475 \fi%
3476 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3477 {}%
3478 {\hfill}%Prevent underfull hbox
3479 \egroup%
3480 \endgroup%
3481 }%
3482 %

```

XII.7.5 Footnote grouped by line

`\prepare@Xgroupbyline` `\prepare@Xgroupbyline` is a macro called on on the `\<XXX>\vfootnote` if `\Xgroupbyline` is set to true, instead of calling directly the `\insert`.

```

3483 \newcommand{\prepare@Xgroupbyline}[3]{%
3484 \iftoggle{Xgroupbylineseparetwolines@#1}{%
3485 \l@dparsfootspec#2%
3486 \ifcsdef{#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3487 {%
3488 \csgappto%
3489 {#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3490 {%
3491 \ifcsemtyp{Xsymlinenum@#1}%
3492 {\csuse{Xparafootsep@#1}}%
3493 {}%
3494 #3{#1}{#2}%
3495 \hskip\csuse{Xafternote@#1}\relax%
3496 }%
3497 }%
3498 {%
3499 \csdef%
3500 {#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3501 {%
3502 #3{#1}{#2}%
3503 \hskip\csuse{Xafternote@#1}\relax%
3504 }%

```

```

3505 }%
3506 \listcsxadd{#1@forinserting}{\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3507 }{%
3508 \ifcsdef{#1@forinserting@all}{%
3509 \csgappto%
3510 {#1@forinserting@all}%
3511 {%
3512 \ifcsemt{Xsymlinenum@#1}%
3513 {\csuse{Xparafootsep@#1}}%
3514 }%
3515 #3{#1}{#2}%
3516 \hskip\csuse{Xafternote@#1}\relax%
3517 }%
3518 }%
3519 {%
3520 \csdef%
3521 {#1@forinserting@all}%
3522 {%
3523 #3{#1}{#2}%
3524 \hskip\csuse{Xafternote@#1}\relax%
3525 }%
3526 }%
3527 \listcsgadd{#1@forinserting}{all}%
3528 }%
3529 }%
3530 %

```

XIII Familiar footnotes

XIII.1 Adjacent footnotes

The original edmac provided users with five series of critical footnotes (`\Afootnote` `\Bfootnote` `\Cfootnote` `\Dfootnote` `\Efootnote`), and \TeX provides a single numbered footnote. The `reledmac` package uses the edmac mechanism to provide six series of numbered footnotes.

First, though, the `footmisc` package has an option whereby two or more consecutive footnotes have their marks separated by commas. This seemed to Peter Wilson such a useful ability that it was provided automatically by `eledmac`.

Maïeul Rouquette has maintained this feature in `reledmac`, despite he thought that is not directly in relationship with the aim of `reledmac`.

`\multiplefootnotemarker` These macros may have been defined by the `memoir` class, are provided by the `footmisc` package and perhaps by other footnote packages. That is why we use `\providecommand` and not `\newcommand`.

```

3531 \providecommand*\multiplefootnotemarker{\footnote}
3532 \providecommand*\multfootsep{\textsuperscript{\normalfont,}}

```

```
3533
3534 %
```

\m@mmf@prepare A pair of self-cancelling kerns. This may have been defined in the memoir class.

```
3535 \providecommand*\m@mmf@prepare{%
3536   \kern-\multiplefootnotemarker
3537   \kern\multiplefootnotemarker\relax}
3538 %
```

\m@mmf@check This may have been defined in the memoir class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```
3539 \providecommand*\m@mmf@check{%
3540   \ifdim\lastkern=\multiplefootnotemarker\relax
3541     \edef\x@sf{\the\spacefactor}%
3542     \unkern
3543     \multfootsep
3544     \spacefactor\x@sf\relax
3545   \fi}
3546
3547 %
```

We have to modify `\@footnotetext` and `\@footnotemark`. However, if memoir is used the modifications have already been made.

```
3548 \@ifclassloaded{memoir}{%
3549 %
```

\@footnotetext Add `\m@mmf@prepare` at the end of `\@footnotetext`.

```
3550 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
3551 %
```

\@footnotemark Modify `\@footnotemark` to cater for adjacent footnotes.

```
3552
3553 \patchcmd{\@footnotemark}
3554   {\nobreak}
3555   {\m@mmf@check
3556    \nobreak
3557   }
3558   {}{}
3559 \patchcmd{\@footnotemark}
3560   {\@makefnmark}
3561   {\@makefnmark
3562    \m@mmf@prepare
3563   }
3564   {}{}
3565 %
```


Finished the modifications for the non-memoir case.

```
3566 }
3567
3568 %
```

XIII.2 Regular footnotes for numbered texts

`\l@doldold@footnotetext` In order to enable the regular `\footnotes` in numbered text we have to play around with its `\@footnotetext`, using different forms for when in numbered or regular text.

```
3569 \pretocmd{\@footnotetext}{%
3570   \ifnumberedpar@
3571   \edtext{}{\l@dbfnote{#1}}}%
3572 \else
3573   {}{}
3574 \apptocmd{\@footnotetext}{\fi}{}{}%
3575 %
```

`\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.

```
\l@dbfnote
\vl@dbfnote
\vl@dbfnote
\footnote
\get@fnmark
\get@thisfootnote
3576
3577 \patchcmd%
3578   {\footnote}%
3579   {\stepcounter\@mpfn}%
3580   {%
3581     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3582       l@dprintingcolumns}}{%
3583       \global\advance\footnote@reading by \@ne%
3584       \get@thisfootnote%
3585       \get@fnmark{\thisc@footnote}%
3586       \ifcsdef{footnotereading\the\footnote@reading=typeset}%
3587         {\setcounter{\@mpfn}{\footnote@reading}}%
3588       }{%
3589         \stepcounter\@mpfn%
3590       }%
3591     }%
3592   }%
3593   {}
3594
3595 \newcommand{\get@thisfootnote}{%
3596   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3597     l@dprintingcolumns}}{%
3598     \protected@xdef\thisc@footnote{\the\footnote@reading}%
3599   }%
```

```

3598 }{%
3599 \protected@xdef\thisc@footnote{\the\c@footnote}%
3600 }%
3601 }%
3602
3603 \newcommand{\l@dbfnote}[1]{%
3604 \get@thisfootnote%
3605 \gdef\@tag{#1\relax}%
3606 \ifledRcol%
3607 \xright@appenditem{%
3608 \ifdefined\Hy@footnote@currentHref%
3609 \noexpand\def\noexpand\Hy@footnote@currentHref{
3610 \fi%
3611 \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
3612 }%
3613 \to\inserts@listR
3614 \global\advance\insert@countR \@ne%
3615 \else%
3616 \xright@appenditem{%
3617 \ifdefined\Hy@footnote@currentHref%
3618 \noexpand\def\noexpand\Hy@footnote@currentHref{
3619 \fi%
3620 \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
3621 }%
3622 \to\inserts@list
3623 \global\advance\insert@count \@ne%
3624 \fi
3625 \ignorespaces%
3626 }%
3627
3628 \newcommand{\get@fnmark}[1]{%
3629 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3630 l@dprintingcolumns}}%
3631 {%
3632 \stepcounter{footnote@typeset}%
3633 \setcounter{footnote}{\c@footnote@typeset}%
3634 \immediate\write\@mainaux{%
3635 \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
3636 }%
3637 \def\@thefnmark{\thefootnote}%
3638 }%
3639 {%
3640 \setcounter{footnote}{#1}%
3641 \def\@thefnmark{\thefootnote}%
3642 }%
3643 }%
3644 \newcommand{\vl@dbfnote}[2]{%

```

```

3645 \get@fnmark{#2}%
3646 \@footnotetext{#1}%
3647 }%
3648 %

```

XIII.3 Footnote formats

Some of the code for the various formats is remarkably similar to that in section ??.

The following macros generally set things up for the ‘standard’ footnote format.

`\prebodyfootmark` Two convenience macros for use by `\...@footnotemark...` macros.
`\postbodyfootmark`

```

3649 \newcommand*\prebodyfootmark{%
3650 \leavevmode
3651 \ifhmode
3652 \edef\x@sf{\the\spacefactor}%
3653 \m@mmf@check
3654 \nobreak
3655 \fi}
3656 \newcommand*\postbodyfootmark{%
3657 \m@mmf@prepare
3658 \ifhmode\spacefactor\x@sf\fi\relax}
3659 %
3660 %

```

XIII.4 Footnote arrangement

XIII.4.1 User level macro

`\arrangementX` `\arrangementX[⟨s⟩]{⟨arrangement⟩}` command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

3661 \newcommandx{\arrangementX}[2][1,usedefault]{%
3662 \def\do##1{%
3663 \csname arrangementX@#2\endcsname{##1}%
3664 }%
3665 \ifstrepty{#1}%
3666 {%
3667 \dolistloop{\@series}%
3668 }%
3669 {
3670 \docsvlist{#1}%
3671 }%
3672 }%
3673 %

```

XIII.4.2 Normal footnotes

`\normal@footnotemarkX` `\normal@footnotemarkX{<series>}` sets up the typesetting of the marker at the point where the footnote is called for.

```

3674 \newcommand*{\normal@footnotemarkX}[1]{%
3675   \prebodyfootmark
3676   \wrapped@bodyfootmarkX{#1}%
3677   \postbodyfootmark}
3678
3679 %

```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{<series>}` *really* typesets the in-text marker. The style is the normal superscript.

```

3680 \newcommand*{\normalbodyfootmarkX}[1]{%
3681   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}%
3682 %

```

`\normalvfootnoteX` `\normalvfootnoteX{<series>}{<text>}` does the `\insert` for the `<series>` and calls the series' `\footfmt...` to format the `<text>`.

```

3683 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalvfootnoteX}[2]{%
3684   \csuse{beforeinsertingX@#1}%
3685   \insert\@nameuse{footins#1}\bgroup
3686     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3687     \noindent\csuse{bhooknoteX@#1}%
3688     \csuse{notefontsizeX@#1}%
3689     \footsplitskips
3690     \ifl@dpairing\ifl@dpaging\else%
3691       \setnoteswidthliketwocolumnsX@{#1}%
3692     \fi\fi%
3693     \setnotesXpositionliketwocolumns@{#1}%
3694     \spaceskip=\z@skip \xspaceskip=\z@skip
3695     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
3696
3697 %

```

`\mpnormalvfootnoteX` The minipage version.

```

3698 \newcommand*{\mpnormalvfootnoteX}[2]{%
3699   \get@thisfootnoteX{#1}%
3700   \get@fnmarkX{#1}{\thisc@footnote}%
3701   \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3702   \global\setbox\@nameuse{mpfootins#1}\vbox{%
3703     \unvbox\@nameuse{mpfootins#1}
3704     \noindent\csuse{bhooknoteX@#1}%
3705     \csuse{notefontsizeX@#1}%
3706     \hsize\columnwidth
3707     \@parboxrestore

```

```

3708 \color@begingroup
3709 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
3710
3711 %

```

\normalfootfmtX `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

3712 \notbool{parapparatus@}{\newcommand*{\normalfootfmtX}[2]{%
3713 \ifluatex%
3714 \texdir\footnote@luatexttexdir%
3715 \pardir\footnote@luatexpardir%
3716 \par%
3717 \fi%
3718 \protected@edef\currentlabel{%
3719 \@nameuse{@thefnmark#1}%
3720 }%
3721 \ledsetnormalparstuffX{#1}%
3722 \hangindent=\csuse{hangindentX@#1}%
3723 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3724 \rule{z@}{\splittopskip}%
3725 {{\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}}}%
3726 \csuse{wrapcontentX@#1}{#2}%
3727 \strut\par}}
3728
3729 %

```

\normalfootfootmarkX `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

3730 \newcommand*{\normalfootfootmarkX}[1]{%
3731 \textsuperscript{\@nameuse{@thefnmark#1}}}
3732
3733 %

```

\normalfootstartX `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

3734 \newcommand*{\normalfootstartX}[1]{%
3735 \ifdimequal{Opt}{\prenotesX@}{}%
3736 {%
3737 \iftoggle{prenotesX@}{%
3738 \togglefalse{prenotesX@}%
3739 \skip\csname footins#1\endcsname=%
3740 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3741 }%
3742 }%
3743 }%
3744 \vskip\skip\csname footins#1\endcsname%
3745 \leftskip=\z@
3746 \rightskip=\z@

```

```

3747 \ifl@dpairing\else%
3748   \hsize=\old@hsize%
3749 \fi%
3750 \setnoteswidthliketwocolumnsX@{#1}%
3751 \setnotesXpositionliketwocolumns@{#1}%
3752 \print@footnoteXrule{#1}%
3753 }%
3754
3755 %

```

\normalfootnoteruleX The rule drawn before the footnote series group.

```

3756 \let\normalfootnoteruleX=\footnoterule
3757
3758 %

```

\normalfootgroupX `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

3759 \newcommand*{\normalfootgroupX}[1]{%
3760   \csuse{bhookgroupX@#1}%
3761   \unvbox\@nameuse{footins#1}%
3762   \hsize=\old@hsize%
3763 }%
3764
3765 %

```

\mpnormalfootgroupX The minipage version.

```

3766 \newcommand*{\mpnormalfootgroupX}[1]{%
3767   \vskip\skip\@nameuse{mpfootins#1}
3768   \ifl@dpairing\ifparledgroup%
3769     \leavevmode\marks\parledgroup@{begin}%
3770     \marks\parledgroup@series{#1}%
3771     \marks\parledgroup@type{footnoteX}%
3772   \fi\fi\normalcolor
3773   \ifparledgroup%
3774     \ifl@dpairing%
3775     \else%
3776       \setnoteswidthliketwocolumnsX@{#1}%
3777       \setnotesXpositionliketwocolumns@{#1}%
3778       \print@footnoteXrule{#1}%
3779     \fi%
3780   \else%
3781     \setnoteswidthliketwocolumnsX@{#1}%
3782     \setnotesXpositionliketwocolumns@{#1}%
3783     \print@footnoteXrule{#1}%
3784   \fi%
3785   \csuse{bhookgroupX@#1}%
3786   \unvbox\@nameuse{mpfootins#1}}

```

```

3787
3788 %
\normalbfnoteX
3790 \newcommand{\normalbfnoteX}[2]{%
3791   \get@thisfootnoteX{#1}%
3792   \ifledRcol%
3793     \ifluatex
3794       \footnotelang@lua[R]%
3795     \fi
3796     \@ifundefined{xpg@main@language}%if polyglossia
3797       {}%
3798       {\footnotelang@poly[R]}%
3799     \xright@appenditem{%
3800       \noexpand\led@set@index@fornote{#1}%
3801       \noexpand\prepare@edindex@fornote{\led@nums}%
3802       \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3803       \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
3804       \noexpand\led@reinit@index@fornote%
3805       \unexpanded{\advance\@edindex@fornote@m@ne}%
3806     }%
3807       \to\inserts@listR
3808     \global\advance\insert@countR \@ne%
3809   \else%
3810     \ifluatex
3811       \footnotelang@lua%
3812     \fi
3813     \@ifundefined{xpg@main@language}%if polyglossia
3814       {}%
3815       {\footnotelang@poly}%
3816     \xright@appenditem{%
3817       \noexpand\led@set@index@fornote{#1}%
3818       \noexpand\prepare@edindex@fornote{\led@nums}%
3819       \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3820       \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
3821       \noexpand\led@reinit@index@fornote%
3822       \unexpanded{\advance\@edindex@fornote@m@ne}%
3823     }%
3824       \to\inserts@list
3825     \global\advance\insert@count \@ne%
3826   \fi
3827   \ignorespaces}
3828
3829 %

```

`\get@thisfootnoteX` The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

3830 \newcommand{\get@thisfootnoteX}[1]{%
3831   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3832     \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
endcsname}%
3833   }{%
3834     \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}%
3835   }%
3836 }%
3837 %

```

\vbfnoteX This command calls the correct footnote-inserting commands.

```

3838 \newcommand{\vbfnoteX}[3]{%
3839   \get@fnmarkX{#1}{#3}%
3840   \@nameuse{regvfootnote#1}{#1}{#2}%
3841 }%
3842 %
3843 %

```

\get@fnmarkX This command gets the correct footnote number when typesetting parallel texts.

```

3844 \newcommand{\get@fnmarkX}[2]{%
3845   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3846     {%
3847       \stepcounter{footnote#1@typeset}%
3848       \setcounter{footnote#1}{\value{footnote#1@typeset}}%
3849       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3850       \immediate\write\@mainaux{%
3851         \csgdef{footnote#1@reading#2=typeset}{\the\csname c@footnote#1
@typeset\endcsname}%
3852       }%
3853     }%
3854     {%
3855       \setcounter{footnote#1}{#2}%
3856       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3857     }%
3858   }
3859   %
3860   %

```

```

\numfootnoteX61 \newcommand{\numfootnoteX}[2]{%
3862   \ifnumberedpar@
3863     \edtext{}{\normalbfnoteX{#1}{#2}}%
3864   \else
3865     \def\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3866     \get@thisfootnoteX{#1}%
3867     \get@fnmarkX{#1}{\expandonce\thisc@footnote}%

```



```

3868 \@nameuse{regvfootnote#1}{#1}{#2}%
3869 \fi}
3870
3871 %

```

`\arrangementX@normal` `\arrangementX@normal{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```

3872 \newcommand*{\arrangementX@normal}[1]{%
3873 \csgdef{series@displayX#1}{normal}
3874 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
3875 \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
3876 \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
3877 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
3878 \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
3879 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
3880 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
3881 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
3882 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3883 \count\csname footins#1\endcsname=1000
3884 \csxdef{default@footins#1}{1000}%Use to have note only for one side
3885 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3886 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3887 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3888 %

```

Additions for minipages.

```

3889 \ifnoledgroup@else%
3890 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3891 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
3892 \count\csname mpfootins#1\endcsname=1000
3893 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
3894 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3895 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3896 \fi
3897 }
3898
3899 %

```

XIII.4.3 Two columns footnotes

The following macros set footnotes in two columns. It is assumed that the length of each footnote is less than the column width.

```

\arrangementX@twocol\newcommand*{\arrangementX@twocol}[1]{%
3901 \csgdef{series@displayX#1}{twocol}
3902 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
3903 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
3904 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX

```

```

3905 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3906 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3907 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3908 \twocolfootsetupX{#1}
3909 \ifnoledgroup@else%
3910 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3911 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
3912 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3913 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
3914 \mptwocolfootsetupX{#1}
3915 \fi%
3916 }
3917
3918 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
3919 \newcommand*{\twocolfootsetupX}[1]{%
3920 \count\csname footins#1\endcsname 500
3921 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
3922 \multiply\dimen\csname footins#1\endcsname by \tw@}
3923 \newcommand*{\mptwocolfootsetupX}[1]{%
3924 \count\csname mpfootins#1\endcsname 500
3925 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
3926
3927 %

```

```

\twocolvfootnoteX \twocolvfootnoteX{<series>}
3928 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
3929 \csuse{beforeinsertingX@#1}%
3930 \insert\csname footins#1\endcsname\bgroup%
3931 \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3932 \noindent\csuse{bhooknoteX@#1}%
3933 \csuse{notefontsizeX@#1}%
3934 \footsplitskips%
3935 \spaceskip=\z@skip \xspaceskip=\z@skip%
3936 \@nameuse{footfmt#1}{#1}{#2}\egroup}
3937
3938 %

```

```

\twocolfootfmtX \twocolfootfmtX{<series>}
3939 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
3940 \protected@edef\@currentlabel{%
3941 \@nameuse{@thefnmark#1}%
3942 }%
3943 \normal@pars%
3944 \hangindent=\csuse{hangindentX@#1}%

```

```

3945 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3946 \hsize \csuse{hsizetwocolX@#1}%
3947 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3948 \tolerance=5000\relax%
3949 \par%
3950 \@tempdima=\parindent%
3951 \csuse{colalignX@#1}%
3952 \parindent=\@tempdima%
3953 {\hspace{\parindent}%
3954 \csuse{notenumentfontX@#1}\wrapped@footfootmarkX{#1}\strut%
3955 \csuse{wrapcontentX@#1}{#2}%
3956 \strut\par}%
3957 \allowbreak%
3958 }%
3959
3960 %

```

```

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX
3961 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
3962 \splittopskip=\ht\strutbox
3963 \expandafter
3964 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
3965
3966 \newcommand*{\mptwocolfootgroupX}[1]{%
3967 \vskip\skip\@nameuse{mpfootins#1}
3968 \ifl@dpairing\ifparledgroup%
3969 \leavevmode\marks\parledgroup@{begin}%
3970 \marks\parledgroup@series{#1}%
3971 \marks\parledgroup@type{footnoteX}%
3972 \fi\fi\normalcolor
3973 \ifparledgroup%
3974 \ifl@dpairing%
3975 \else%
3976 \setnoteswidthliketwocolumnsX@{#1}%
3977 \setnotesXpositionliketwocolumns@{#1}%
3978 \print@footnoteXrule{#1}%
3979 \fi%
3980 \else%
3981 \setnoteswidthliketwocolumnsX@{#1}%
3982 \setnotesXpositionliketwocolumns@{#1}%
3983 \print@footnoteXrule{#1}%
3984 \fi%
3985 \csuse{bhookgroupX@#1}%
3986 \splittopskip=\ht\strutbox
3987 \expandafter
3988 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
3989
3990 %

```

XIII.4.4 Three columns footnotes

The following macros set footnotes in three columns. It is assumed that the length of each footnote is less than the column width.

```

\arrangementX@threecol \newcommand*\arrangementX@threecol}[1]{%
3992   \csgdef{series@displayX#1}{threecol}
3993   \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
3994   \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
3995   \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
3996   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3997   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3998   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3999   \threecolfootsetupX{#1}
4000   \ifnoledgroup@ \else%
4001     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4002     \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4003     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4004     \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4005     \mpthreecolfootsetupX{#1}
4006   \fi%
4007 }
4008
4009 %

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX
4010 \newcommand*\threecolfootsetupX}[1]{%
4011   \count\csname footins#1\endcsname 333
4012   \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4013   \multiply\dimen\csname footins#1\endcsname by \thr@@
4014   \newcommand*\mpthreecolfootsetupX}[1]{%
4015     \count\csname mpfootins#1\endcsname 333
4016     \multiply\dimen\csname mpfootins#1\endcsname by \thr@@
4017
4018   %

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
4019 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{
%
4020   \csuse{beforeinsertingX@#1}%
4021   \insert\csname footins#1\endcsname\bgroup%
4022     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4023     \noindent\csuse{bhooknoteX@#1}%
4024     \csuse{notefontsizeX@#1}%
4025     \footsplitskips%
4026     \@nameuse{footfmt#1}{#1}{#2}\egroup}
4027

```

```

4028 %
\threecolfootfmtX \threecolfootfmtX{<series>}
4029 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4030   \protected@edef\@currentlabel{%
4031     \@nameuse{@thefnmark#1}%
4032   }%
4033   \hangindent=\csuse{hangindentX@#1}%
4034   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4035   \normal@pars%
4036   \hsize \csuse{hsizethreecolX@#1}%
4037   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4038   \tolerance=5000\relax%
4039   \@tempdima=\parindent%
4040   \csuse{colalignX@#1}%
4041   \parindent=\@tempdima%
4042   {\hspace{\parindent}%
4043     \csuse{notenunfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4044     \csuse{wrapcontentX@#1}{#2}%
4045     \strut\par}\allowbreak}
4046
4047 %

```

```

\threecolfootgroupX \threecolfootgroupX{<series>}
\mpthreecolfootgroupX
4048 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4049   \splittopskip=\ht\strutbox
4050   \expandafter
4051   \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4052
4053 \newcommand*{\mpthreecolfootgroupX}[1]{%
4054   \vskip\skip\@nameuse{mpfootins#1}
4055   \ifl@dpairing\ifparledgroup
4056     \leavevmode\marks\parledgroup@{begin}%
4057     \marks\parledgroup@series{#1}%
4058     \marks\parledgroup@type{footnoteX}%
4059     \fi\fi\normalcolor
4060     \ifparledgroup%
4061       \ifl@dpairing%
4062       \else%
4063         \setnoteswidthliketwocolumnsX@{#1}%
4064         \setnotesXpositionliketwocolumns@{#1}%
4065         \print@footnoteXrule{#1}%
4066       \fi%
4067     \else%
4068       \setnoteswidthliketwocolumnsX@{#1}%
4069       \setnotesXpositionliketwocolumns@{#1}%
4070       \print@footnoteXrule{#1}%

```

```

4071 \fi%
4072 \csuse{bhookgroupX@#1}%
4073 \splittopskip=\ht\strutbox
4074 \expandafter
4075 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4076
4077 %

```

XIII.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@threecol` `\footparagraphX{<series>}`

```

4078 \newcommand*{\arrangementX@paragraph}[1]{%
4079 \csgdef{series@displayX#1}{paragraph}%
4080 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4081 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4082 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4083 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4084 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4085 \count\csname footins#1\endcsname=1000
4086 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
4087 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4088 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4089 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4090 \para@footsetupX{#1}
4091 \ifnoledgroup@else
4092 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4093 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4094 \count\csname mpfootins#1\endcsname=1000
4095 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4096 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4097 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4098 \fi
4099 }
4100
4101 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

4102 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4103 \setnoteswidthliketwocolumnsX@{#1}%
4104 \ifcsempy{widthX@#1}%
4105 {}%
4106 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4107 \dimen0=\baselineskip
4108 \multiply\dimen0 by 1024

```

```

4109 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
4110 %
4111 \expandafter
4112 \xdef\csname footfudgefactor#1\endcsname{%
4113   \expandafter\strip@pt\dimen0 }}
4114 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

4115 \newcommand*\parafootstartX}[1]{%
4116   \ifdimequal{0pt}{\prenotesX@}{}%
4117   {%
4118     \iftoggle{prenotesX@}{%
4119       \togglefalse{prenotesX@}%
4120       \skip\csname footins#1\endcsname=%
4121       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4122     }%
4123   }%
4124 }%
4125 \leftskip=\z@
4126 \rightskip=\z@
4127 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4128 \vskip\skip\@nameuse{footins#1}%
4129 \setnoteswidthliketwocolumnsX@{#1}%
4130 \setnotesXpositionliketwocolumns@{#1}%
4131 \print@footnoteXrule{#1}%
4132 }
4133 %
4134 %

```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`

`\mppara@vfootnoteX`

```

4135 \newcommand*\para@vfootnoteX}[2]{%
4136   \csuse{beforeinsertingX@#1}%
4137   \insert\csname footins#1\endcsname%
4138   \bgroup
4139     \csuse{notefontsizeX@#1}
4140     \footsplitskips
4141     \setbox0=\vbox{\hsize=\maxdimen%
4142       \let\bidir@RTL@everypar\@empty%
4143       \noindent\csuse{bhooknoteX@#1}%
4144       \@nameuse{footfmt#1}{#1}{#2}}%
4145     \setbox0=\hbox{\unvXH{0}{#1}}%
4146     \dp0=\z@
4147     \ht0=\csname footfudgefactor#1\endcsname\wd0
4148     \box0
4149     \penalty0
4150   \egroup}
4151 \newcommand*\mppara@vfootnoteX}[2]{%

```

```

4152 \get@thisfootnoteX{#1}%
4153 \get@fnmarkX{#1}{\thisc@footnote}%
4154 \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4155 \global\setbox\@nameuse{mpfootins#1}\vbox{%
4156   \unvbox\@nameuse{mpfootins#1}
4157   \csuse{notefontsizeX@#1}
4158   \footsplitskips
4159   \setbox0=\vbox{\hsize=\maxdimen%
4160     \let\bidirTL@everypar\@empty%
4161     \noindent\color@begingroup%
4162     \csuse{bhooknoteX@#1}%
4163     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
4164   \setbox0=\hbox{\unvxhX{0}{#1}}%
4165   \dp0=\z@
4166   \ht0=\csname footfudgefactor#1\endcsname\wd0
4167   \box0
4168   \penalty0}}
4169
4170 %

```

```

\unvxhX71 \newcommand*{\unvxhX}[2]{% 2th is optional for retro-compatibility
4172   \setbox0=\vbox{\unvbox#1%
4173     \global\setbox1=\lastbox}%
4174   \unhbox1
4175   \unskip           % remove \rightskip,
4176   \unskip           % remove \parfillskip,
4177   \unpenalty        % remove \penalty of 10000,
4178   \hskip\csuse{afternoteX@#2}%
4179   \relax}% but add the glue to go between the notes
4180
4181 %

```

\parafootfmtX \parafootfmtX{<series>}

```

4182 \newcommand*{\parafootfmtX}[2]{%
4183   \protected@edef\@currentlabel{%
4184     \@nameuse{@thefnmark#1}%
4185   }%
4186   \insertparafootsepX{#1}%
4187   \ledsetnormalparstuff@common%
4188   {\csuse{notenumfontX@#1}%
4189     \csuse{notenumfontX@#1}%
4190     \wrapped@footfootmarkX{#1}%
4191     \strut%
4192     \csuse{wrapcontentX@#1}{#2}%
4193     \penalty-10}}
4194
4195 %

```



```

\para@footgroupX \para@footgroupX{<series>}
\mppara@footgroupX
4196 \newcommand*{\para@footgroupX}[1]{%
4197   \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4198   \unvbox\csname footins#1\endcsname
4199   \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4200   \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4201   \makehboxofhboxes
4202   \setbox0=\hbox{\unhbox0 \removehboxes}%
4203   \csuse{bhookgroupX@#1}
4204   \csuse{notefontsizeX@#1}
4205   \unhbox0\par}
4206
4207 \newcommand*{\mppara@footgroupX}[1]{%
4208   \setnoteswidthliketwocolumnsX@{#1}%
4209   \vskip\skip\@nameuse{mpfootins#1}
4210   \ifl@dpairing\ifparledgroup
4211     \leavevmode%
4212     \leavevmode\marks\parledgroup@{begin}%
4213     \marks\parledgroup@series{#1}%
4214     \marks\parledgroup@type{footnoteX}%
4215   \fi\fi\normalcolor
4216   \ifparledgroup%
4217     \ifl@dpairing%
4218     \else%
4219       \setnoteswidthliketwocolumnsX@{#1}%
4220       \setnotesXpositionliketwocolumns@{#1}%
4221       \print@footnoteXrule{#1}%
4222     \fi%
4223   \else%
4224     \setnoteswidthliketwocolumnsX@{#1}%
4225     \setnotesXpositionliketwocolumns@{#1}%
4226     \print@footnoteXrule{#1}%
4227   \fi%
4228   \unvbox\csname mpfootins#1\endcsname
4229   \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4230   \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4231   \makehboxofhboxes
4232   \setbox0=\hbox{\unhbox0 \removehboxes}%
4233   \csuse{bhookgroupX@#1}%
4234   \csuse{notefontsizeX@#1}%
4235   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4236   \unhbox0\par}}
4237
4238 %

```

Insertion of the footnotes separator The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX39 \newcommand{\insertparafootsepX}[1]{%
4240   \ifledRcol{%
4241     \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
4242     {\csuse{Xparafootsep@#1}}%
4243     {}%
4244     \global\csname prevpage#1@numR\endcsname=\page@numR%
4245   \else%
4246     \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
4247     {\csuse{Xparafootsep@#1}}%
4248     {}%
4249     \global\csname prevpage#1@num\endcsname=\page@num%
4250   \fi%
4251 }
4252 %

```

XIII.5 Wrapping footnote marks in hyperlink

`\wrapped@footfootmarkX` prints the footnote mark of the footpage, wrapped in `hyperref` package's commands, if needed.

```

4253 \newcommand{\wrapped@footfootmarkX}[1]{%
4254   \ifdefined\hypertarget%
4255     \hyperlink%
4256       {@bodyfootmark#1@\this@footnoteX@reading}%
4257       {@nameuse{footfootmark#1}}%
4258     \Hy@raisedlink{%
4259       \hypertarget%
4260       {@footnotemark#1@\this@footnoteX@reading}%
4261       {}%
4262     }%
4263   \else%
4264     \@nameuse{footfootmark#1}%
4265   \fi%
4266 }%
4267 %

```

`\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package's commands, if needed.

```

4268 \newcommand{\wrapped@bodyfootmarkX}[1]{%
4269   \ifdefined\hypertarget%
4270     \hyperlink%
4271       {@footnotemark#1\expandafter\the\csname footnote#1@reading\endcsname}%
4272       {@nameuse{bodyfootmark#1}}%
4273     \Hy@raisedlink{%
4274       \hypertarget%
4275       {@bodyfootmark#1\expandafter\the\csname footnote#1@reading\endcsname}%

```

```

4276     {}%
4277   }%
4278   \else%
4279     \@nameuse{bodyfootmark#1}%
4280   \fi%
4281 }%
4282 %

```

XIV Code common to both critical and familiar footnote in normal arrangement

`\par` should always be redefined to `\endgraf` within the format macro (this is what `\normal@pars` does), to override tricky material in the main text to get the lines numbered automatically (as set up by `\autopar`, for example).

In the case of footnote arranged in a “normal” way, we also must set some setting for paragraph indent and text direction when using Lua¹TeX.

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% \textdir\footnote@luatextextdir%
4286 \pardir\footnote@luatexpardir%
4287 \fi%
4288 \csuse{\csuse{footnote@dir}}%
4289 \normal@pars%
4290 \parfillskip \z@ \@plus 1fil}%
4291
4292 \newcommand*{\Xledsetnormalparstuff}[1]{%
4293   \ledsetnormalparstuff@common%
4294   \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
4295 }%
4296
4297 \newcommand*{\ledsetnormalparstuffX}[1]{%
4298   \ledsetnormalparstuff@common%
4299   \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}%
4300 }%
4301 %

```

XV Footnotes’ width for two columns

We define here some commands which make sense only with `reledpar`, but must be called when defining notes parameters. These commands change the width of block notes to allow them to have the same size than two parallel columns.

`\old@hsize` These two commands are called at the beginning of critical or familiar notes groups.
`\setXnoteswidthliketwocolumns@` They set, if the option is enabled, the `\hsize`. They are also called at the on the setup
`\setnoteswidthliketwocolumnsX@` for paragraphed notes.

```

4302
4303 \newdimen\old@hsize%
4304 \AtBeginDocument{\old@hsize=\hsize}%
4305
4306 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
4307   \global\let\hsize@fornote=\hsize%
4308   \global\old@hsize=\hsize%
4309   \let\old@columnwidth=\columnwidth%
4310   \iftoggle{Xnoteswidthliketwocolumns@#1}%
4311     {%
4312       \csuse{setwidthliketwocolumns@\columns@position}%
4313       \global\let\hsize@fornote=\hsize%
4314     }%
4315     {}%
4316   \let\hsize=\hsize@fornote%
4317   \let\columnwidth=\old@columnwidth%
4318 }%
4319
4320 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4321   \global\let\hsize@fornote=\hsize%
4322   \global\old@hsize=\hsize%
4323   \let\old@columnwidth=\columnwidth%
4324   \iftoggle{noteswidthliketwocolumnsX@#1}%
4325     {%
4326       \csuse{setwidthliketwocolumns@\columns@position}%
4327       \global\let\hsize@fornote=\hsize%
4328     }%
4329     {}%
4330   \let\hsize=\hsize@fornote%
4331   \let\columnwidth=\old@columnwidth%
4332 }%
4333
4334 %

```

`\setXnotespositionliketwocolumns@` These two commands set the position of the critical / familiar footnotes, depending on
`\setnotesXpositionliketwocolumns@` the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They
 call commands which are defined only in `reledpar`, because this feature has no sens
 without `reledpar`.

```

4335 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
4336   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4337     \csuse{setnotespositionliketwocolumns@\columns@position}%
4338   }{}%
4339 }%
4340
4341 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%

```

```

4342 \iftoggle{noteswidthliketwocolumnsX@#1}{%
4343 \cuse{setnotespositionliketwocolumns@\columns@position}%
4344 }{}%
4345 }%
4346
4347 %

```

XVI Footnotes' order

`\fnpos` and `\mpfnpos` simply place their arguments in `\@fnpos` and `\@mpfnpos`, which will be used later in the output routine.

```

\fnpos
\mpfnpos
\@fnpos
\@mpfnpos
4348 \def\@fnpos{familiar-critical}
4349 \def\@mpfnpos{critical-familiar}
4350 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
4351 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
4352 %

```

XVII Footnotes' rule

Because the footnotes' rules can be shifted to the right when footnotes are set like two columns, we do not print them directly, but we put them in a `\vbox`.

```

\print@Xfootnoterule 53 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoterule 54 \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
4355 \nointerlineskip%
4356 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
4357 \nointerlineskip%
4358 \vskip\csuse{Xafterrule@#1}%
4359 }%
4360
4361 \newcommand{\print@footnoterule}[1]{%
4362 \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
4363 \nointerlineskip%
4364 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
4365 \nointerlineskip%
4366 \vskip\csuse{afterruleX@#1}%
4367 }%
4368
4369 %

```

XVIII Specific skip for first series of footnotes

XVIII.0.1 Overview

`\Xbeforenotes` inserts a specific skip for the first series of notes in a page. As we can't know in advance which series will be the first, we call `\prepare@Xprenotes` before inserting any critical notes, in order to prevent page number overlapping.

1. If it is the first note of the current page, it changes the footnote skip for the series to the value specified to `\Xbeforenotes`. It also keeps the series of the note as the first one of the current page.
2. If it is not the first note of the current page:
 - If the current series is printed after the series kept as the first of the current page, then nothing happens.
 - If the current series is printed before the series kept as the first of the current page, then it changes the footnote skip of the current series to the value normally used by the series which was marked as the first of the page. It also keeps the current series as the new first one of the current page.

For example, suppose the series order is A,B. We call first a `\Bfootnote` and a `\Afootnote`. The only skips used are, finally, the skip specific to the first series of the page, and the skip for the B series. If we have not called `\Afootnote`, the only skip used is the skip specific to the first series of the page.

That is perfect.

The series skip and the first series of the current page are reset before the footnotes are printed. Then, the footstart macros manage the problem of the first series of the page.

After the rule, the space which is defined by `\Xafterrule` does not depend on whether the series is the first one of the page or not. So we use its normal value for each series.

And now, implementation !

XVIII.0.2 User level command

`\Xprenotes@` If user redefines `\Xprenotes@`, via `\Xprenotes` to a value greater than 0 pt, this skip will be added before first series notes instead of the notes skip.

```

4370 \newtoggle{Xprenotes@}%
4371 \toggletrue{Xprenotes@}%
4372 \newcommand{\Xprenotes@}{0pt}%
4373 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
4374 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
4375 %

```

The same, but for familiar footnotes.

```

\Xprenotes76 \newtoggle{prenotesX@}
\Xprenotes77 \toggletrue{prenotesX@}
4378 \newcommand{\prenotesX@}{Opt}
4379 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
4380 %

```

XVIII.0.3 Internal commands

```

firstXseries@81 \gdef\firstXseries@{}
prepare@Xprenotes82 \newcommand{\prepare@Xprenotes}[1]{%
4383 \ifdimequal{Opt}{\Xprenotes@}%
4384 }%
4385 {%
4386 \IfStrEq{\firstXseries@}{}%
4387 \global\skip\csuse{#1footins}=\Xprenotes@%
4388 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4389 \gdef\firstXseries@{#1}%
4390 }%
4391 {%
4392 \ifseriesbefore{#1}{\firstXseries@}%
4393 {%
4394 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
4395 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4396 \gdef\firstXseries@{#1}%
4397 }%
4398 }%
4399 }%
4400 }%
4401 }
4402 %

```

The same thing is required for familiar notes and \prenotesX.

```

firstseriesX@83 \gdef\firstseriesX@{}
prepare@prenotesX84 \newcommand{\prepare@prenotesX}[1]{%
4405 \ifdimequal{Opt}{\prenotesX@}%
4406 }%
4407 {%
4408 \IfStrEq{\firstseriesX@}{}%
4409 \global\skip\csuse{footins#1}=\prenotesX@%
4410 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4411 \gdef\firstseriesX@{#1}%
4412 }%
4413 {%
4414 \ifseriesbefore{#1}{\firstseriesX@}%

```

```

4415     {%
4416     \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
4417     \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4418     \gdef\firstXseries@{#1}%
4419     }%
4420     {}%
4421   }%
4422 }%
4423 }
4424 %

```

XIX Endnotes

First, check the noend option.

```

4425 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
4426 %

```

XIX.1 Internal commands

\l@dend@open \l@dend@open and \l@dend@close are the macros that are used to open and close the endnote file. Note that all our writing to this file is \immediate: all page and line numbers for the endnotes are generated by the same mechanism we use for the footnotes, so that there is no need to defer any writing to catch information from the output routine. The argument of these two command is the series letter.

```

4427 \newcommand{\l@dend@open}[1]{%
4428   \global\booltrue{l@dend@#1}%
4429   \expandafter\immediate%
4430   \expandafter\openout%
4431   \csname l@d@#1end\endcsname%
4432   =\l@auxdir\jobname.#1end\relax%
4433 }%
4434 \newcommand{\l@dend@close}[1]{%
4435   \global\boolfalse{l@dend@#1}%
4436   \expandafter\immediate%
4437   \expandafter\closeout\csname l@d@#1end\endcsname%
4438 }%
4439
4440 %

```

\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.

```

4441 \newcommand{\l@dend@stuff}{%
4442   \def\do##1{%

```



```

4443 \ifbool{l@dend@##1}{}%
4444   {\l@dend@open{##1}}%
4445   \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
4446   }%
4447   \dolistloop{@series}%
4448 }%
4449
4450 %

```

`\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.

```

4451 \global\newbool{parapparatus@}{\long}\def\endprint#1#2#3#4#5#6#7{%
4452   \csuse{Xendbhooknote@#4}%
4453   \csuse{Xendnotefontsize@#4}%
4454   \hangindent=\csuse{Xendhangindent@#4}%
4455   \ifXendinsertsep@%
4456     \hskip\csuse{Xendafternote@#4}\relax%
4457     \csuse{Xendsep@#4}%
4458   \else%
4459     \iftoggle{Xendparagraph@#4}%
4460       {\global\Xendinsertsep@true}%
4461     }%
4462   \fi%
4463   \xdef\@currentseries{#4}%
4464   \def\do##1{%
4465     \setkeys[mac]{truefootnoteoption}{##1}%
4466   }%
4467   \notblank{#5}{\docsvlist{#5}}{}%
4468   \IfStrEq{#6}{R}{\ledRcol@true}{}%
4469   \def\@this@crossref@start{#7:start}%
4470   \def\@this@crossref@end{#7:end}%

```

```

4471 \printlineendnote{#1}{#4}%
4472 \IfStrEq{#6}{R}{\ledRcol@false}{}%
4473 \undef\@this@crossref@start%
4474 \undef\@this@crossref@end%
4475 \nottoggle{Xendlemmadisablefontselection@#4}%
4476 {\select@lemmafont#1|}%
4477 {}%
4478 \bgroup%
4479 \csuse{Xendlemmafont@#4}%
4480 \csuse{Xendwraplemma@#4}{#2}%
4481 \egroup%
4482 \ifboolexpr{%
4483   togl {nosep@}%
4484   or test{\ifcsemtty{Xendlemmaseparator@#4}}%
4485 }%
4486 {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
4487 {\nobreak%
4488   \hskip\csuse{Xendbeforelemmaseparator@#4}%
4489   \csuse{Xendlemmaseparator@#4}%
4490   \hskip\csuse{Xendafterlemmaseparator@#4}%
4491   \relax%
4492 }%
4493 \csuse{Xendwrapcontent@#4}{#3}%
4494 \nottoggle{Xendparagraph@#4}{\par}{}%
4495 \def\do##1{%
4496   \setkeys[mac]{falsefootnoteoption}{##1}%
4497 }%
4498 \notblank{#5}{\docsvlist{#5}}{}%
4499 }}%
4500
4501 \let\l@d@section=\@gobble
4502
4503 %

```

\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.

```

4504 \newcommand{\printlineendnote}[2]{%
4505   \l@dp@rsefootspec#1%
4506   \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
4507     \edef\lineinfo@{\l@d@p@rsestartpage - \l@d@p@rsestartline - \
4508       \l@d@p@rsestartsub - \l@d@p@rseendpage - \l@d@p@rseendline - \
4509       \l@d@p@rseendsub}%
4510     }%
4511     {\
4512       \edef\lineinfo@{\l@d@p@rsestartpage - \l@d@p@rsestartline - \
4513         \l@d@p@rsestartsub}%
4514       }%
4515     \ifboolexpr{%

```

```

4513     togl {nonum@}%
4514     or togl {Xendnonumber@#2}%
4515     }%
4516     {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4517     {%
4518     \iftoggle{Xendnumberonlyfirstinline@#2}%
4519     {\ifcsdef{prevendline#2}%
4520     {\ifcsequal{prevendline#2}{lineinfo@}%
4521     {%
4522     \csuse{Xendbhookinplaceofnumber@#2}%
4523     \ifcsequal{Xendsymmlinenumber@#2}%
4524     {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4525     {\printsymmlineendnotearea{#2}}}%
4526     \csuse{Xendahookinplaceofnumber@#2}%
4527     }%
4528     {\printlineendnotearea{#1}{#2}}}%
4529     {\printlineendnotearea{#1}{#2}}}%
4530     }%
4531     {\printlineendnotearea{#1}{#2}}%We keep every time line
4532     \csxdef{prevendline#2}{\lineinfo@}%
4533     }%
4534 }%
4535 %

```

```

\printsymmlineendnotearea% \newcommand{\printsymmlineendnotearea}[1]{%
4537 \hspace{\csuse{Xendbeforemlinenumber@#1}}%
4538 \csuse{Xendnotenumfont@#1}%
4539 \ifdimequal{\csuse{Xendboxsymmlinenumber@#1}}{\z@}%
4540 {\csuse{Xendsymmlinenumber@#1}}%
4541 {\hbox to \csuse{Xendboxsymmlinenumber@#1}%
4542 {\csuse{Xendsymmlinenumber@#1}\hfill}%
4543 }%
4544 \hspace{\csuse{Xendaftersymmlinenumber@#1}}%
4545 }%
4546 %

```

\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.)

```

4547 \newcommand{\printlineendnotearea}[2]{%
4548 \csuse{Xendbhooklinenumber@#2}%
4549 \hspace{\csuse{Xendbeforenumber@#2}}%
4550 \bgroup%
4551 \csuse{Xendnotenumfont@#2}%
4552 \ifdimequal{\csuse{Xendboxlinenumber@#2}}{0pt}%
4553 {\printendlines#1||\ifledRcol@{\Rlineflag\fi}%
4554 {\leavevmode%

```

```

4555 \hbox to \csuse{Xendboxlinenum@#2}%
4556 {%
4557 \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4558 \printendlines#1||\ifledRcol@\@Rlineflag\fi%
4559 \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4560 }%
4561 \egroup%
4562 \hspace{\csuse{Xendafternumber@#2}}%
4563 \csuse{Xendahooklinenum@#2}%
4564 }%
4565 %

```

XIX.2 User level commands

XIX.2.1 Inserting contents to endnotes

The `\Xendnotes` commands are defined upper, when defining apparatus commands by series. Here, we define. only `\toendnotes` command not specific to a series, in order to insert arbitrary code. The regular version writes an unexpanded argument, while the regular version writes a once-expanded argument.

```

\toendnotes*66 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes*67 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4568 \def\do##1{%
4569 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4570 {\unexpanded{#2}\@percentchar}%
4571 }%
4572 \ifstrempy{#1}%
4573 {\dolistloop{\@series}}%
4574 {\docsvlist{#1}}%
4575 }{\led@err@toendnotes@outsidenumbering}%
4576 }%
\WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
4577 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4578 \def\do##1{%
4579 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4580 {#2\@percentchar}%
4581 }%
4582 \ifstrempy{#1}%
4583 {\dolistloop{\@series}}%
4584 {\docsvlist{#1}}%
4585 }{\led@err@toendnotes@outsidenumbering}%
4586 }%
4587 }%
4588 %

```

XIX.2.2 Printing endnotes

`\doendnotes` `\doendnotes` is the command you use to print one series of endnotes; it takes one argument: the series letter of the note series you want to print. `\Xendinsertsep@` is

set to true at the first note of the series, and to false at the last one.

```

4589 \newif\ifXendinsertsep%
4590 \newcommand*{\doendnotes}[1]{%
4591   \l@dend@close{#1}%
4592   \begingroup
4593     \csxdef{prevpagenum@#1}{}%
4594     \csxdef{prevpagerange@#1}{}%
4595     \makeatletter
4596     \expandafter\let\csname #1end\endcsname=\endprint
4597     \input\l@auxdir\jobname.#1end%
4598     \global\Xendinsertsep@false%
4599   \endgroup}
4600 %

```

\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.

```

4601 \newcommand*{\doendnotesbysection}[1]{%
4602   \l@dend@close{#1}%
4603   \csxdef{prevpagenum@#1}{}%
4604   \csxdef{prevpagerange@#1}{}%
4605   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
4606   \begingroup%
4607     \makeatletter%
4608     \def\l@d@section##1{%
4609       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
4610         {\cslet{#1end}{\endprint}}%
4611         {\cslet{#1end}{\@gobblefive}}%
4612       }%
4613       \input\l@auxdir\jobname.#1end%
4614       \global\Xendinsertsep@false%
4615     \endgroup%
4616   }%
4617 %

```

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.

```

4618 }%
4619 %

```

\setprintendlines The \printendlines macro is similar to \printlines but is for printing endnotes rather than footnotes.

The principal difference between foot- and endnotes is that footnotes are printed on the page where they are specified but endnotes are printed at a different point in the

document. We need an indication of the source of an endnote; `\setprintendlines` provides this by always printing the page number. The coding is slightly simpler than `\setprintlines`.

First of all, we print the second page number only if the ending page number is different from the starting page number.

```
4620 \newcommand*{\setprintendlines}[6]{%
4621   \l@dpnumfalse \l@ddashfalse
4622   \ifnum#4=#1 \else
4623     \l@dpnumtrue
4624     \l@ddashtrue
4625   \fi
4626   %
```

We print the ending line number if: (1) we are printing the ending page number, or (2) it is different from the starting line number.

```
4627   \ifl@dpnum \l@d@elintrue \else \l@d@elinfalse \fi
4628   \ifnum#2=#5 \else
4629     \l@d@elintrue
4630     \l@ddashtrue
4631   \fi
4632   %
```

We print the starting sub-line if it is nonzero.

```
4633   \l@d@ssubfalse
4634   \ifnum#3=0 \else
4635     \l@d@ssubtrue
4636   \fi
4637   %
```

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.

```
4638   \l@d@eslfalse
4639   \ifnum#6=0 \else
4640     \ifnum#6=#3
4641       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4642     \else
4643       \l@d@esltrue
4644       \l@ddashtrue
4645     \fi
4646   \fi%
4647   %
```

```
4648   \ifl@d@dash%
4649     \ifboolexpr{togl{fulllines@} or test{\ifcempty{Xendtwolines@}\
@currentseries}}}%
4650     {}%
4651     {%
4652     \setistwofollowinglines{#1}{#2}{#4}{#5}%
```

```

4653 \ifboolexpr{%
4654   (%
4655     togl {Xendtwolinesbutnotmore@\@currentseries}%
4656     and not%
4657     (%
4658       bool {istwofollowinglines@}%
4659     )%
4660   )%
4661   or%
4662   (%
4663     (not test{\ifnumequal{#1}{#4}})%
4664     and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
4665   )%
4666   }%
4667   {}%
4668   {%
4669     \l@d@dashfalse%
4670     \l@d@Xtwolinestrue%
4671     \l@d@elinfalse%
4672     \l@d@eslfalse%
4673     \ifcempty{Xendmorethantwolines@\@currentseries}%
4674     {}%
4675     {\ifistwofollowinglines@\else%
4676       \l@d@Xmorethantwolinestrue%
4677       \fi%
4678     }%
4679   }%
4680 }%
4681 \fi%
4682 %

End of \setprintendlines.

4683 }%
4684 %

```

\printendlines Now we are ready to print it all.

```

4685 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4686   \begingroup
4687   \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
4688   %

```

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.

```

4689 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
4690   {\bgroup}%

```

```

4691 {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}%
4692 %

```

Then, print the starting page number-

```

4693 \ifboolexpr{%
4694   (%
4695     test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
4696     and not%
4697     (togl{Xendpagenumberonlyfirstifsingle@\@currentseries} and bool{
l@d@pnum}))%
4698   )%
4699   or%
4700   (%
4701     test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
4702   )%
4703 }%
4704 {%
4705   \ifcseempty{Xendsympagenum@\@currentseries}%
4706     {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
4707     {\csuse{Xendsympagenum@\@currentseries}}%
4708   }%
4709   {%
4710     \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
4711   }%
4712 %

```

Then, determine what must be printed before the start line.

```

4713 \ifl@d@dash%
4714   \ifl@d@pnum%
4715     \csuse{Xendlineprefixsingle@\@currentseries}%
4716   \else%
4717     \ifcseempty{Xendlineprefixmore@\@currentseries}%
4718       {\csuse{Xendlineprefixsingle@\@currentseries}}%
4719       {\csuse{Xendlineprefixmore@\@currentseries}}%
4720   \fi%
4721 \else%
4722   \csuse{Xendlineprefixsingle@\@currentseries}%
4723 \fi%
4724 %

```

The print the starting line, followed, if needed, by the side flag and the starting sub line number.

```

4725 \wrap@edcrossref{\@this@crossref@start}{%
4726   \ifledRcol@%
4727     \linenumrepR{#2}%
4728   \else%
4729     \linenumrep{#2}%
4730   \fi%

```



```

4731 }%
4732 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi}{}%
4733 \ifl@d@ssub%
4734 \csuse{Xendsublinesep@ \@currentseries}%
4735 \wrap@edcrossref{\@this@crossref@start}{%
4736 \ifledRcol@%
4737 \sublinenumrepR{#3}%
4738 \else%
4739 \sublinenumrep{#3}%
4740 \fi%
4741 }%
4742 \fi%
4743 %

```

Close the box.

```

4744 \egroup%
4745 %

```

Open the box for the ending line number.

```

4746 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
4747 {\bgroup}%
4748 {\hbox to \csuse{Xendboxendlinenum@ \@currentseries}\bgroup}%
4749 %

```

Print the dash + the ending line number, or the line number range symbol.

```

4750 \ifl@d@xtwolines%
4751 \ifl@d@morethantwolines%
4752 \csuse{Xendmorethantwolines@ \@currentseries}%
4753 \else%
4754 \csuse{Xendtwolines@ \@currentseries}%
4755 \fi%
4756 \else%
4757 \ifl@d@dash%
4758 \ifdefined\linerangesep@%
4759 \linerangesep@%
4760 \else%
4761 \csuse{Xendlinerangeseparator@ \@currentseries}%
4762 \fi%
4763 \fi%
4764 %

```

Print the ending page number.

```

4765 \ifl@d@pnum%
4766 \ifcsstring{prevpagerange@ \@currentseries}{#1-#4}%
4767 {%
4768 \ifcsempy{Xendsympagenum@ \@currentseries}%
4769 {\hspace{\csuse{Xendinplaceofpagenumber@ \@currentseries}}}%
4770 {\csuse{Xendsympagenum@ \@currentseries}}%
4771 }%

```

```

4772      {%
4773      \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
4774      }%
4775      \fi%
4776 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag and the subline number.

```

4777      \ifl@d@elin%
4778      \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
4779      \wrap@edcrossref{\@this@crossref@end}{%
4780      \ifledRcol%
4781      \linenumrepR{#5}%
4782      \else%
4783      \linenumrep{#5}%
4784      \fi%
4785      }%
4786      \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
4787      \fi%
4788      \ifl@d@esl%
4789      \ifl@d@elin%
4790      \csuse{Xendsublinesep@\@currentseries}%
4791      \fi%
4792      \wrap@edcrossref{\@this@crossref@end}{%
4793      \ifledRcol%
4794      \sublinenumrepR{#6}%
4795      \else%
4796      \sublinenumrep{#6}%
4797      \fi%
4798      }%
4799      \fi%
4800      \fi%
4801 %

```

Close the ending line box.

```

4802      \ifdimequal{\csuse{Xendboxendlinenumber@\@currentseries}}{0pt}%
4803      {}%
4804      {\hfill}%Prevent underfull hbox
4805      \egroup%
4806 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

4807      \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
4808      {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%
4809      {\csxdef{prevpagerange@\@currentseries}{#1-#4}}%
4810      {\csxdef{prevpagenum@\@currentseries}{#4}}%
4811      }%
4812      {}%
4813 %

```

Now, the end of `\printendlines` macro.

```
4814 \endgroup%
4815 }%
4816
4817 %
```

`\printnpnum` A macro to print a page number in an endnote. Should not be override anymore

```
4818 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@\@currentseries
4819 }#1\csuse{Xendafterpagenumber@\@currentseries}}
4820 %
```

XX Generate series of notes

In this section, X means the name of the series (A, B etc.)

`\series` `\series\series` creates one more new series. It is a public command, which just loops on the private command `\newseries@`.

```
4821 \newcommand{\newseries}[1]{%
4822   \def\do##1{\newseries@{##1}}%
4823   \docsvlist{#1}
4824 }
4825 %
```

`\@series` The `\series@` macro is an etoolbox list, which contains the name of all series.

```
4826 \newcommand{\@series}{%
4827 %
```

The command `\newseries@\series` creates a new series of the footnote.

`\newseries@` `\newcommand{\newseries@}[1]{`

```
4828 %
4829 %
```

XX.1 Test if series is still existing

```
4830 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
4831 {%
4832 %
```

XX.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```

4833 \ifdefined\newseries@par%
4834 \newseries@par{#1}%
4835 \fi%
4836 %

```

XX.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```

4837 \unless\ifnocritical@
4838 %

```

XX.3.1 Options

```

4839 \newtoggle{Xlineflag@#1}
4840 \newtoggle{Xparindent@#1}
4841 \newtoggle{Xlemmadisablefontselection@#1}
4842 \csgdef{Xwrapcontent@#1}{}%
4843 \csgdef{Xbeforeinserting@#1}{}%
4844 \csgdef{Xhangindent@#1}{Opt}%
4845 \csgdef{Xragged@#1}{}%
4846 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
4847 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
4848 \csgdef{Xcolalign@#1}{\raggedright}%
4849 \csgdef{Xnotenunfont@#1}{\normalfont}%
4850 \csgdef{Xnotefontsize@#1}{\footnotesize}%
4851 \csgdef{Xbhooknote@#1}{}%
4852 \csgdef{Xbhookgroup@#1}{}%
4853
4854 \csgdef{Xboxlinenum@#1}{Opt}%
4855 \csgdef{Xboxlinenumalign@#1}{L}%
4856
4857 \csgdef{Xboxstartlinenum@#1}{Opt}%
4858 \csgdef{Xboxendlinenum@#1}{Opt}%
4859
4860 \csgdef{Xboxsymlinenum@#1}{Opt}%
4861 \newtoggle{Xnumberonlyfirstinline@#1}%
4862 \newtoggle{Xgroupbyline@#1}%
4863 \newtoggle{Xgroupbylineseparetwolines@#1}%
4864 \newtoggle{Xnumberonlyfirstintwolines@#1}%
4865 \csgdef{Xtwolines@#1}{}%
4866 \csgdef{Xmorethantwolines@#1}{}%
4867 \csgdef{Xsublinesep@#1}{\fullstop}%
4868 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
4869 \newtoggle{Xtwolinesbutnotmore@#1}%
4870 \newtoggle{Xtwolinesonlyinsamepage@#1}%

```

```

4871 \newtoggle{Xonlypstart@#1}%
4872 \newtoggle{Xpstarteverytime@#1}%
4873 \newtoggle{Xpstart@#1}%
4874 \newtoggle{Xstanza@#1}%
4875 \csgdef{Xstanzaseparator@#1}{}%
4876 \csgdef{Xsymlinenum@#1}{}%
4877 \newtoggle{Xnonumber@#1}%
4878 \csgdef{Xbeforenumber@#1}{Opt}%
4879 \csgdef{Xtxtbeforenumber@#1}{}%
4880 \csgdef{Xafternumber@#1}{0.5em}%
4881 \newtoggle{Xnonbreakableafternumber@#1}%
4882 \csgdef{Xbeforenymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
4883 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
4884 \csgdef{Xinplaceofnumber@#1}{1em}%
4885 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
4886 \csgdef{Xbeforelemmaseparator@#1}{0em}%
4887 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
4888 \csgdef{Xinplaceofflemmaseparator@#1}{1em}%
4889 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
4890 \csgdef{Xafterrule@#1}{Opt}%
4891
4892 \csgdef{Xtxtbeforenotes@#1}{%
4893 \newtoggle{Xtxtbeforenotes@#1@typeset}%Not directly used by user,
but internal
4894
4895 \csgdef{Xmaxhnotes@#1}{0.8\vsizel}
4896 \newtoggle{Xnoteswidthliketwocolumns@#1}%
4897 \csgdef{Xparafootsep@#1}{}%
4898 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%
4899 \csgdef{Xlinerrangeseparator@#1}{\endashchar}%
4900
4901 \csgdef{Xlemmafont@#1}{}%
4902 \csgdef{Xwraplemma@#1}{%
4903 \csgdef{Xwidth@#1}{\hsizel}%
4904 %

```

XX.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

4905 \expandafter\newinsert\csname #1footins\endcsname%
4906 \unless\ifnoledgroup%
4907 \expandafter\newinsert\csname mp#1footins\endcsname%
4908 \fi%
4909 %

```

XX.3.3 Create commands for critical apparatus, \Afootnote, \Bfootnote etc.

Note the double # in command: it is because command it is made inside another command.

```

4910 \global\newbool{parapparatus@}\expandafter\newcommand\expandafter
*}{\expandafter\newcommand\csname #1footnote\endcsname[2] [] {%
4911 \if@edtext@secondarg%
4912 \ifledRcol%
4913 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}}%
4914 \else%
4915 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}}%
4916 \fi%
4917 \begingroup%
4918 \newcommand{\content}{##2}%
4919 \ifnumberedpar%
4920 \ifledRcol%
4921 \ifluatex%
4922 \footnotelang@lua[R]%
4923 \fi%
4924 \@ifundefined{xpg@main@language}%if polyglossia
4925 {}%
4926 {\footnotelang@poly[R]}%
4927 \footnoteoptions@{R}{#1}{true}%
4928 \xright@appenditem{%
4929 \ifbool{indtl@innote}%
4930 {\unexpanded{\let\index\nindex}}%
4931 {}%
4932 \ifbool{indtl@notenumber}%
4933 {\unexpanded{\let\index\nindex}}%There is no note
...number so
4934 {}%
4935 \noexpand\Xnote@true%
4936 \noexpand\prepare@Xprenotes{#1}%
4937 \noexpand\prepare@edindex@fornote{\l@d@nums}%
4938 \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.
4939 \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
4940 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
4941 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
4942 \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
4943 \noexpand\csuse{v#1footnote}{#1}%
4944 {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
4945 \noexpand\Xnote@false%
4946 \unexpanded{\advance\@edindex@fornote@m@ne}%
4947 \unexpanded{\undef\@this@crossref@start}%
4948 \unexpanded{\undef\@this@crossref@end}%

```

```

4949         \ifbool{indtl@innote}%
4950             {\unexpanded{\let\index\orig@@index}}%
4951             {}%
4952         \ifbool{indtl@notenumber}%
4953             {\unexpanded{\let\index\orig@@index}}%
4954             {}%
4955         }\to\inserts@listR
4956         \footnoteoptions@{R}{##1}{false}%
4957         \global\advance\insert@countR \@ne%
4958     \else%
4959         \ifluatex%
4960             \footnotelang@lua%
4961         \fi%
4962         \@ifundefined{xpg@main@language}%if polyglossia
4963             {}%
4964             {\footnotelang@poly}%
4965         \footnoteoptions@{L}{##1}{true}%
4966         \xright@appenditem{%
4967             \ifbool{indtl@innote}%
4968                 {\unexpanded{\let\index\nindex}}%
4969                 {}%
4970             \ifbool{indtl@notenumber}%
4971                 {\unexpanded{\let\index\nindex}}%There is no note
...number so
4972             {}%
4973             \noexpand\Xnote@true%
4974             \noexpand\prepare@Xprenotes{##1}%
4975             \noexpand\prepare@edindex@fornote{\l@d@nums}%
4976             \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.
4977             \ifl@pairing%
4978                 \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
4979             \fi%
4980             \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
4981             \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
4982             \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
4983             \noexpand\csuse{v#1footnote}%
4984             {##1}%
4985             {\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
4986             \unexpanded{\undef\@this@crossref@start}%
4987             \unexpanded{\undef\@this@crossref@end}%
4988             \noexpand\Xnote@false%
4989             \unexpanded{\advance\@edindex@fornote@m@ne}%
4990             \ifbool{indtl@innote}%

```

```

4991         {\unexpanded{\let\index\orig@@index}}}%
4992     }%
4993     \ifbool{indtl@notenumber}%
4994     {\unexpanded{\let\index\orig@@index}}}%
4995     }%
4996     }\to\inserts@list
4997     \global\advance\insert@count \@ne%
4998     \footnoteoptions@{L}{#1}{false}%
4999     \fi
5000     \else
5001         \csuse{v#1footnote}{#1}{0|0|0|0|0|0|0|0}{#1}%
5002     \fi%
5003     \endgroup%
5004     \else%
5005         \led@err@FootnoteNotInSecondArgEdtext{#1}%
5006     \fi%
5007     \ignorespaces%
5008 }
5009 %

```

Create counter used to determine on which page the previous note was called.

```

5010     \expandafter\newcount\csname #1prevpage@num\endcsname%
5011     \expandafter\newcount\csname #1prevpage@numR\endcsname%
5012 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

5013     \global\csletcs{#1@@footnote}{#1footnote}
5014 %

```

XX.3.4 Set standard display

```

5015     \Xarrangement@normal{#1}%
5016 %

```

End of for critical footnotes.

```

5017     \fi
5018 %

```

XX.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```

5019     \unless\ifnofamiliar@
5020 %

```


XX.4.1 Options

```

5021 \newtoggle{parindentX@#1}
5022 \csgdef{wrapcontentX@#1}{}%
5023 \csgdef{hangindentX@#1}{Opt}%
5024 \csgdef{beforeinsertingX@#1}{}%
5025 \csgdef{raggedX@#1}{}%
5026 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
5027 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
5028 \csgdef{colalignX@#1}{\raggedright}%
5029 \csgdef{notenumfontX@#1}{\normalfont}%
5030 \csgdef{notefontsizeX@#1}{\footnotesize}%
5031 \csgdef{bhooknoteX@#1}{}%
5032 \csgdef{bhookgroupX@#1}{}%
5033 \csgdef{afterruleX@#1}{Opt}
5034 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
5035 \csgdef{maxhnotesX@#1}{0.8\vsize}%
5036 \newtoggle{noteswidthliketwocolumnsX@#1}%
5037 \csgdef{parafootsepX@#1}{}%
5038 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
5039 \csgdef{widthX@#1}{\hsize}%
5040 % End of for familiar footnotes.
5041 % \subsubsection{Create inserts, needed to add notes in foot}
5042 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
5043 % \begin{macrocode}
5044 \expandafter\newinsert\csname footins#1\endcsname%
5045 \unless\ifnoledgroup%
5046 \expandafter\newinsert\csname mpfootins#1\endcsname%
5047 \fi%
5048 %

```

XX.4.2 Create tools for familiar footnotes (\footnoteX)

First, create the \footnoteX command. Note the double # in command: it is because a command is called inside another command.

```

5049
5050 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
5051 \begingroup%
5052 \prepare@prenotesX{#1}%
5053 \newcommand{\content}{##1}%
5054 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.1.2 p. 50).

```

5055 \global\expandafter\advance\csname footnote#1@reading\
5056 \endcsname by \@ne%
5057 \ifbool{bool{1@dpairing}} or bool{1@dprintingpages} or
5058 bool{1@dprintingcolumns}}{%
5059 \ifcsdef{footnote#1@reading\the\csname footnote#1@reading\
5060 \endcsname=typeset}%

```

```

5058         {\setcounter{footnote#1}{\csuse{footnote#1reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
5059         {\setcounter{footnote#1}{\the\csname footnote#1@reading
\endcsname}}}%
5060     }{%
5061         \stepcounter{footnote#1}%
5062     }%
5063 %

```

We also have to check consistency with `\onlysideX` setting.

```

5064         \ifledRcol%
5065         \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}}%
5066         \else%
5067         \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}}%
5068         \fi%
5069 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

5070         \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5071         \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
5072         {\csuse{@footnotemark#1}}%
5073         {}%
5074         \ifluatex%
5075         \xdef\footnote@luatextextdir{\the\textdir}%
5076         \xdef\footnote@luatexpardir{\the\pardir}%
5077         \fi%
5078         \if@ledgroup%
5079         \led@set@index@for@note{#1}%
5080         \fi%
5081         \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
5082         \ifbool{indtl@innote}%
5083         {\let\index\orig@@index}%
5084         {}%
5085         \ifbool{indtl@notenumber}%
5086         {\let\index\orig@@index}%
5087         {}%
5088         \endgroup%
5089     }
5090 %

```

Then define the counters. The \LaTeX counter `footnoteX` is the only one manipulated by the user. This is the one which is printed. The \TeX counter `\footnoteX@reading` is increased at each footnote. It is used for hyperlinks, for using `hyperlink` package, and for getting the correct footnote number when using parallel typesetting (V.1.2 p. 50).

```

5091         \newcounter{footnote#1}
5092         \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}

```

```

5093 \expandafter\newcount\csname footnote#1@reading\endcsname%
5094 %

```

Create counter used to determine on which page the previous note was called.

```

5095 \expandafter\newcount\csname prevpage#1@num\endcsname%
5096 \expandafter\newcount\csname prevpage#1@numR\endcsname%
5097 %

```

Add `\let\footnoteX\@gobble` to `\no@expands`.

```

5098 \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
5099 %

```

Do not forget to initialize the series

```

5100 \arrangementX@normal{#1}%
5101 \fi
5102 %

```

XX.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

5103 \unless\ifnoend@
5104 %

```

XX.5.1 The auxiliary file

`\l@d@Xend` Endnotes of all varieties are saved up in a file, one by series, typically named `<jobname>.Xend`.
`\ifl@dend@X` `\l@d@end` is the output stream number for this file, and `\ifl@dend@X` is a flag that is
`\l@dend@Xtrue` true when the file is open.
`\l@dend@Xfalse`

```

5105 \expandafter\newwrite\csname l@d@#1end\endcsname%
5106 \expandafter\newif\csname ifl@dend@#1\endcsname%
5107 %

```

XX.5.2 The main macro

The `\Xendnote` macro functions to write one endnote to the `.Xend` file. We change `\newlinechar` so that in the file every space becomes the start of a new line; this generally ensures that a long note does not exceed restrictions on the length of lines in files.

```

5108
5109 \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,
usedefault]{%
5110 \bgroup%
5111 \newlinechar='40%

```

```

5112 \global\@noneed@Footnotetrue%
5113 \newcommand{\content}{##2}%
5114 \stepcounter{labidx}%
5115 \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
5116 \unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\
expandafter{\sw@inthisedtext}}\@percentchar\space%Explicit space, to add a
linebreak in the output file
5117 \expandafter\string\csname #1end\endcsname%
5118 {\ifnumberedpar@l@d@nums\fi}%
5119 {\ifnumberedpar@expandonce\@tag\fi}%
5120 {\expandonce\content}%
5121 {#1}%
5122 {\unexpanded{##1}}%
5123 {\ifledRcol R\else L\fi}%
5124 {\theedtext}%
5125 \@percentchar%
5126 }%
5127 \egroup%
5128 \ignorespaces%
5129 }%
5130 %

```

XX.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoennotes`

```

5131 \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
5132 \toendnotes[#1]{##1}%
5133 }%
5134
5135 \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
endcsname*[1]{%
5136 \toendnotes*[#1]{##1}%
5137 }%
5138
5139 %
5140 %

```

XX.5.4 Internal commands

`\Xendnote` commands called `\Xend` commands on to the endnote file; these are analogous to the various `footfmt` commands above, and they take the same arguments. When we process this file, we want to pick out the notes of one series and ignore all the rest. To do that, we equate the end command for the series we want to `\endprint`, and leave the rest equated to `\@gobblefive`, which just skips over its five arguments.

```

5141

```

```

5142 \global\cslet{#1end}{\@gobblefive}
5143 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

5144 \global\expandafter\newcount\csname #1endbysection\endcsname%
5145 %

```

XX.5.5 The options

```

5146 \csgdef{Xendwraplemma@#1}{%
5147 \csgdef{Xendwrapcontent@#1}{}%
5148 \csgdef{Xendtwolines@#1}{}%
5149 \csgdef{Xendmoreethantwolines@#1}{}%
5150 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
5151 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
5152 \newtoggle{Xendlemmadisablefontselection@#1}%
5153 \csgdef{Xendnotenumfont@#1}{\normalfont}%
5154 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
5155 \csgdef{Xendbhooknote@#1}{}%
5156
5157 \csgdef{Xendsublinesep@#1}{\fullstop}%
5158
5159 \csgdef{Xendbeforenumber@#1}{Opt}
5160 \csgdef{Xendafternumber@#1}{0.5em}
5161
5162 \csgdef{Xendboxlinenum@#1}{Opt}%
5163 \csgdef{Xendboxlinenumalign@#1}{L}%
5164
5165 \csgdef{Xendboxstartlinenum@#1}{Opt}%
5166 \csgdef{Xendboxendlinenum@#1}{Opt}%
5167
5168 \csgdef{Xendlemmaseparator@#1}{}%
5169 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
5170 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
5171 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
5172
5173 \newtoggle{Xendparagraph@#1}%
5174 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
5175 \csgdef{Xendsep@#1}{}%
5176
5177 \csgdef{Xendinplaceofnumber@#1}{Opt}%
5178 \newtoggle{Xendnonumber@#1}%
5179
5180 \csgdef{Xendhangindent@#1}{Opt}%
5181 \newtoggle{Xendnumberonlyfirstinline@#1}%
5182 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
5183
5184 \csgdef{Xendbeforesymlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
5185 \csgdef{Xendaftersymlinenum@#1}{\csuse{Xendafternumber@#1}}%

```

```

5186 \csgdef{Xendsymmlinenumber@#1}{}%
5187 \csgdef{Xendboxsymmlinenumber@#1}{Opt}%
5188
5189 \csgdef{Xendbhooklinenumber@#1}{}%
5190 \csgdef{Xendehooklinenumber@#1}{}%
5191 \csgdef{Xendbhookinplaceofnumber@#1}{}%
5192 \csgdef{Xendehookinplaceofnumber@#1}{}%
5193
5194 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%
5195
5196 \csgdef{Xendbeforepagenumber@#1}{p.}%
5197 \csgdef{Xendafterpagenumber@#1}{) }%
5198 \csgdef{Xendlineprefixsingle@#1}{}%
5199 \csgdef{Xendlineprefixmore@#1}{}%
5200
5201 \newtoggle{Xendlineflag@#1}
5202
5203 \csgdef{Xendlemmafont@#1}{}%
5204
5205 \newtoggle{Xendpagenumberonlyfirst@#1}%
5206 \newtoggle{Xendpagenumberonlyfirstifsingle@#1}%
5207 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
5208 \csgdef{Xendsympagenum@#1}{}%
5209 \csgdef{Xendinplaceofpagenumber@#1}{Opt}%
5210
5211 %

```

End of endnotes declaration

```

5212 \fi%
5213 %

```

Dump series in \@series

```

5214 \listxadd{\@series}{#1}
5215 }
5216 }% End of \newseries
5217 %

```

XX.6 Init standards series (A,B,C,D,E)

```

5218 \expandafter\newseries\expandafter{\default@series}
5219 %

```

XXI Setting series display

XXI.1 Change series order

`\seriesatbegin` `\seriesatbegin{⟨s⟩}` changes the order of series, to put the series `⟨s⟩` at the beginning of the list. The series can be the result of a command.

```

5220 \newcommand{\seriesatbegin}[1]{%
5221   \StrDel{\@series}{#1}[\@series]%
5222   \edef\@new{}%
5223   \listadd{\@new}{#1}%
5224   \listadd{\@new}{\@series}%
5225   \xdef\@series{\@new}%
5226 }
5227 %

```

\seriesatend And `\seriesatend` moves the series to the end of the list.

```

5228 \newcommand{\seriesatend}[1]{%
5229   \StrDel{\@series}{#1}[\@series]%
5230   \edef\@new{}%
5231   \listadd{\@new}{\@series}%
5232   \listadd{\@new}{#1}%
5233   \xdef\@series{\@new}%
5234 }
5235 %

```

XXI.2 Test series order

\ifseriesbefore `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands `<true>` if `<seriesA>` is printed before `<seriesB>`, expands `<false>` otherwise.

```

5236 \newcommand{\ifseriesbefore}[4]{%
5237   \StrPosition{\@series}{#1}[\@first]%
5238   \StrPosition{\@series}{#2}[\@second]%
5239   \ifnumgreater{\@second}{\@first}{#3}{#4}%
5240 }
5241 %

```

XXI.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries42 \newcommand{\@getfirstseries}{%
5243   \ifdefempty{\@series}%
5244   {\xdef\@firstseries{}}%
5245   {\StrChar{\@series}{1}[\@firstseries]}%
5246 }%
5247 %

```

XXI.3 Series setting

XXI.3.1 General way of working

The setting's command (like `\numberonlyfirstinline`), also called “hooks” can be divided in two categories: those which require a string values and those which require

a boolean value. The first category includes those which require a length value, because we store the length's expression send by user and we evaluate it only in the commands which requires to know the setting. The second category require boolean value only when it is set to FALSE. Otherwise, we understand the insinuated value is TRUE.

For each “hook” command, we store the value in commands (first category) or a etoolbox's toggle (second category) which names are in the form `\<hook>@<series>`. For example when calling `\twolines{<sq.>}`, we store `sq.` in commands `\twolines@A`, `\twolines@B`, `\twolines@C`...for each series defined for use with `reledmac`, or, if the `[<series>]` optional argument was send, for each series of this argument.

These values are tested in some specific places, scattered in all the code, depending of their effects. The default values are defined by the `\newseries@` command.

In order to prevent code duplication, we have created some generic commands. Some of them change the value of any hook send as argument. Some other, getting a hook name, generate the user level commands.

XXI.3.2 Tools to set options

`\settoggle@series` `\settoggle@series{<series>}{<toggle>}{<value>}` is a generic command to switch toggles for some series. The arguments are:

- #1 (mandatory): the series for which the hooks should be set. If empty, all the series will be affected.
- #2 (mandatory): the name of the hook.
- #3 (mandatory): the new value of toggle (true or false).
- #4 (optional): if equal to `reload`, reload the footnote setting (call again `\Xarrangement` or `\arrangementX` or ... depending of the footnote display).
- #5 (optional): if not empty, and if #1 is empty, change the hook setting for pseudo-series, as `appref`.

```

5248 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
5249   \def\do##1{%
5250     \global\settoggle{#2@##1}{#3}%
5251     \ifstrequal{#4}{critical}{
5252       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
5253     }{}
5254     \ifstrequal{#4}{familiar}{
5255       \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
5256     }{}
5257   }%
5258   \ifstreempty{#1}{%
5259     \dolistloop{@series}%
5260     \ifstreempty{#5}{%
5261       \docsvlist{#5}%
5262     }
5263   }%

```



```

5264      {%
5265      \docsvlist{#1}%
5266      }%
5267  }
5268  %

```

`\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`.

```

5269 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
5270   \def\do##1{
5271     \csgdef{#2@##1}{#3}
5272     \ifstrequal{#4}{critical}{%
5273       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
5274     }{
5275       \ifstrequal{#4}{familiar}{%
5276         \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
5277       }{}%
5278   }%
5279   \ifstreempty{#1}{%
5280     \dolistloop{\@series}%
5281     \ifstreempty{#5}{}{%
5282       \docsvlist{#5}
5283     }
5284   }%
5285   {%
5286     \docsvlist{#1}%
5287   }%
5288 }%
5289 %

```

XXI.3.3 Tools to generate options commands

`\newhookcommand@series` `\newhookcommand@series\command` names is a generic command to add new commands for hooks, like `\Xhsizetwocol`. The first argument is the name of the hook, the second a comma-separated list of pseudo-series where the hook can be used, like

appref in the case of `\Xtwolines`. The second argument is also used to create commands named `\<hookname><pseudoseris>`, like `\Xtwolinesappref`.

```

5290 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
5291   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1]{%
5292     \setcommand@series{##1}{#1}{##2}[1][#2]%
5293   }%
5294   \ifstrempy{#2}{}{%
5295     \def\do##1{%
5296       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
5297       [1]{%
5298         \csuse{#1}[##1]{####1}%
5299       }%
5300     \docsvlist{#2}%
5301   }%
5302 }
5303 %

```

`\newhooktoggle@series` `\newhooktoggle@series\command` names is a generic command to add new commands for a new toggle hook, like `\Xnumberonlyfirstinline`. The second argument is also used to create commands named `\<hookname><pseudoseris>`, like `\Xtwolinesbutnotmoreappref`.

```

5304 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
5305   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
5306     true},usedefault]{%
5307     \settoggle@series{##1}{#1}{##2}[1][#2]%
5308   }%
5309   \ifstrempy{#2}{}{%
5310     \def\do##1{%
5311       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
5312         \csuse{#1}[##1]%
5313       }%
5314     \docsvlist{#2}%
5315   }%
5316 }
5317 %

```

`\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

```

5318 \newcommand{\newhooktoggle@series@reload}[2]{%
5319   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
5320     true},usedefault]{%
5321     \settoggle@series{##1}{#1}{##2}[#2]%
5322   }%
5323 %

```

`\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.

```

5324 \newcommand{\newhookcommand@series@reload}[2]{%
5325   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][\{%
5326     \setcommand@series{##1}{#1}{##2}[#2]%
5327   }%
5328 }
5329 %

```

XXI.3.4 Options for critical notes

Before generating the commands that are used to set the critical notes, such as `\Xnumberonlyfirstinline`, `\Xlemmaseparator` and the like, we check the `nocritical` option.

```

5330 \unless\ifnocritical@
5331   \newhookcommand@series{Xwrapcontent}%
5332   \newhookcommand@series{Xbeforeinserting}%
5333   \newhookcommand@series{Xlemmafont}%
5334   \newhookcommand@series{Xwraplemma}%
5335   \newhooktoggle@series{Xparindent}
5336   \newhookcommand@series{Xhangindent}
5337   \newhookcommand@series{Xragged}
5338   \newhookcommand@series{Xhsizetwocol}
5339   \newhookcommand@series{Xhsizethreecol}
5340   \newhookcommand@series{Xcolalign}%
5341   \newhookcommand@series{Xnotenumfont}
5342   \newhookcommand@series{Xbhooknote}
5343   \newhookcommand@series@reload{Xbhookgroup}{critical}
5344   \newhookcommand@series{Xboxsymlinenum}%
5345   \newhookcommand@series{Xsymlinenum}
5346   \newhookcommand@series{Xbeforenumber}
5347   \newhookcommand@series{Xtxtbeforenumber}
5348   \newhookcommand@series{Xafternumber}
5349   \newhookcommand@series{Xbeforesymlinenum}
5350   \newhookcommand@series{Xaftersymlinenum}
5351   \newhookcommand@series{Xinplaceofnumber}
5352   \newhookcommand@series{Xlemmaseparator}
5353   \newhookcommand@series{Xbeforelemmaseparator}
5354   \newhookcommand@series{Xafterlemmaseparator}
5355   \newhookcommand@series{Xinplaceoflemmaseparator}
5356   \newhookcommand@series{Xtxtbeforenotes}
5357   \newhookcommand@series@reload{Xafterrule}{critical}
5358   \newhooktoggle@series{Xnumberonlyfirstinline}
5359   \newhooktoggle@series{Xnumberonlyfirstintwolines}
5360   \newhooktoggle@series{Xgroupbyline}%
5361   \newhooktoggle@series{Xgroupbylineseparatetwolines}%
5362   \newhooktoggle@series{Xnonumber}
5363   \newhooktoggle@series{Xpstart}

```

```

5364 \newhooktoggle@series{Xpstarteverytime}%
5365
5366 \newhooktoggle@series{Xstanza}%
5367 \newhookcommand@series{Xstanzaseparator}%
5368
5369 \newhooktoggle@series{Xonlypstart}
5370 \newhooktoggle@series{Xnonbreakableafternumber}
5371 \newhooktoggle@series{Xlemmadisablefontselection}
5372 \newhookcommand@series@reload{Xmaxhnotes}{critical}
5373 \newhookcommand@series@reload{Xbeforenotes}{critical}
5374 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
5375 \newhookcommand@series@reload{Xnotefontsize}{critical}
5376
5377 \newhookcommand@series{Xboxlinenum}%
5378 \newhookcommand@series{Xboxlinenumalign}%
5379
5380 \newhookcommand@series{Xboxstartlinenum}%
5381 \newhookcommand@series{Xboxendlinenum}%
5382
5383 \newhookcommand@series{Xafternote}%
5384 \newhookcommand@series{Xparafootsep}
5385
5386 \newhookcommand@series@reload{Xwidth}{critical}%
5387
5388 \ifundef{\Xhsize}%
5389 {%
5390   \newcommandx{\Xhsize}[2][1,usedefault]{%
5391     \led@warning@Xhsize@deprecated%
5392     \Xwidth[#1]{#2}%
5393   }%
5394 }%
5395 {}%
5396 \fi
5397 \newhooktoggle@series{Xlineflag}[appref,SEref]
5398 \newhookcommand@series{Xtwolines}[appref,SEref]
5399 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
5400 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
5401 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
5402 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
5403 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
5404 \newhookcommand@series{Xlinerrangeseparator}[appref,SEref]
5405 %

```

XXI.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

5406 \unless\ifnofamiliar@
5407   \newhookcommand@series{wrapcontentX}%

```

```

5408 \newhookcommand@series{beforeinsertingX}%
5409 \newhooktoggle@series{parindentX}
5410 \newhookcommand@series{hangindentX}
5411 \newhookcommand@series{raggedX}
5412 \newhookcommand@series{hsizetwocolX}
5413 \newhookcommand@series{hsizethreecolX}
5414 \newhookcommand@series{colalignX}%
5415 \newhookcommand@series{notenumfontX}
5416 \newhookcommand@series{bhooknoteX}
5417 \newhookcommand@series@reload{bhookgroupX}{familiar}
5418 \newhookcommand@series@reload{beforenotesX}{familiar}
5419 \newhookcommand@series@reload{maxhnotesX}{familiar}
5420 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
5421 \newhookcommand@series@reload{afterruleX}{familiar}
5422 \newhookcommand@series@reload{notefontsizeX}{familiar}
5423 \newhookcommand@series{afternoteX}
5424 \newhookcommand@series{parafootsepX}
5425 \newhookcommand@series@reload{widthX}{familiar}%
5426 \ifundef{\hsizeX}%
5427   {%
5428     \newcommandx{\hsizeX}[2][1,usedefault]{%
5429       \led@warning@hsizeX@deprecated%
5430       \widthX[#1]{#2}%
5431     }%
5432   }%
5433   {}%
5434 \fi
5435 %

```

XXI.3.6 Options for endnotes

Before generating the commands that are used to set the endnotes, such as `\Xnumberonlyfirstinline`, `\Xlemmaseparator+` and the like, we check the `noend` option.

```

5436 \unless\ifnoend@
5437   \newhookcommand@series{Xendwraplemma}
5438   \newhookcommand@series{Xendwrapcontent}
5439   \newhookcommand@series{Xendnotenumfont}
5440   \newhookcommand@series{Xendlemmafnt}%
5441   \newhookcommand@series{Xendbhooknote}
5442
5443   \newhookcommand@series{Xendboxlinenum}%
5444   \newhookcommand@series{Xendboxlinenumalign}%
5445
5446   \newhookcommand@series{Xendboxstartlinenum}%
5447   \newhookcommand@series{Xendboxendlinenum}%
5448
5449   \newhookcommand@series{Xendnotefontsize}
5450   \newhooktoggle@series{Xendlemmadisablefontselection}
5451   \newhookcommand@series{Xendlemmaseparator}

```

```

5452 \newhookcommand@series{Xendbeforelemmaseparator}
5453 \newhookcommand@series{Xendafterlemmaseparator}
5454 \newhookcommand@series{Xendinplaceoflemmaseparator}
5455
5456 \newhookcommand@series{Xendbeforenumber}%
5457 \newhookcommand@series{Xendafternumber}%
5458
5459 \newhooktoggle@series{Xendparagraph}
5460 \newhookcommand@series{Xendafternote}
5461 \newhookcommand@series{Xendsep}
5462
5463 \newhookcommand@series{Xendinplaceofnumber}%
5464 \newhooktoggle@series{Xendnonumber}%
5465
5466 \newhooktoggle@series{Xendnumberonlyfirstinline}%
5467 \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
5468
5469 \newhookcommand@series{Xendsymlinenum}%
5470 \newhookcommand@series{Xendbeforesymlinenum}%
5471 \newhookcommand@series{Xendaftersymlinenum}%
5472 \newhookcommand@series{Xendboxsymlinenum}%
5473
5474 \newhookcommand@series{Xendbhooklinenumber}%
5475 \newhookcommand@series{Xendahooklinenumber}%
5476 \newhookcommand@series{Xendbhookinplaceofnumber}%
5477 \newhookcommand@series{Xendahookinplaceofnumber}%
5478
5479 \newhookcommand@series{Xendhangindent}%
5480
5481 \newhooktoggle@series{Xendpagenumberonlyfirst}%
5482 \newhooktoggle@series{Xendpagenumberonlyfirstifsingle}%
5483 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
5484 \newhookcommand@series{Xendsympagenum}%
5485 \newhookcommand@series{Xendinplaceofpagenumber}%
5486
5487 \fi
5488 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
5489 \newhookcommand@series{Xendtwolines}[apprefwithpage,SErefwithpage]
5490 \newhookcommand@series{Xendmoreethantwolines}[apprefwithpage,SErefwithpage]
5491 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,SErefwithpage]
5492 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
SErefwithpage]
5493 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage
]
5494 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
SErefonlypage]
5495 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
5496 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
5497 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
5498 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]

```

5499
5500 %

XXI.4 Hooks for a particular footnote

\newhooktoggle@specific `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```
5501 \newcommand{\newhooktoggle@specific}[1]{%
5502   \newtoggle{#1}%
5503   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
   When enabling footnote option
5504   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false}
   }}
5505 }
5506 %
```

\newhookarg@specific `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```
5507 \newcommand{\newhookarg@specific}[1]{%
5508   \define@key[mac]{truefootnoteoption}{#1}{\global\def\linrangesep@{##1}}%
   When enabling footnote option
5509   \define@key[mac]{falsefootnoteoption}{#1}{\global\undef\linrangesep@}%
   When
5510 }
5511 %
```

And now, we define some hooks specific to a note.

```
5512 \newhooktoggle@specific{fulllines}%
5513 \newhooktoggle@specific{nonum}
5514 \newhooktoggle@specific{nosep}
5515 \newhookarg@specific{linrangesep}
5516 %
```

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.

5517 %

\nomk@ `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```
5518 \newtoggle{nomk}%
5519 %
```

XXI.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```
5520 \newcommand*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]{}}
5521 %
```

XXII Output routine

Now we begin the output routine and associated things.

XXII.1 Extra footnotes output

With luck we might only have to change `\@makecol` and `\@reinserts` of the \TeX 's kernel. Since `reledmac`, we use `etoolbox`'s patching commands instead of overriding. It should provides better compatibility with other package which modify these commands

`\doxtrafeet` `\doxtrafeet` is the code extending `\@makecol` to cater for the extra `reledmac` feet. We have two categories of extra footnotes. By default, we order the footnote inserts so that the regular footnotes of \TeX are first, then familiar familiar footnotes and finally the critical footnotes.

```
5522 \newcommand*{\l@ddoxtrafeet}{%
5523   \IfStrEq{familiar-critical}{\@fnpos}
5524     {\do@feetX\do@Xfeet}%
5525   {%
5526     \IfStrEq{critical-familiar}{\@fnpos}%
5527       {\do@Xfeet\do@feetX}%
5528     {%
5529       \setbox\@outputbox \vbox{%
5530         \unvbox\@outputbox%
5531         \do@feet@custom@order}{\@fnpos}%
5532       }%
5533     }%
5534   }%
5535 }%
5536
5537 %
```

`\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.

```
5538 \newcommand{\do@feet@custom@order}[2]{%
5539   \def\do##1{%
5540     \edef\@notesseries{\@firstoftwo##1}%
5541     \edef\@notetype{\@secondoftwo##1}%
5542     \ifdefstring{\@notetype}{critical}%
```



```

5543     {\csuse{#1append@Xnotes}{\@@notesseries}}}%
5544     {\ifdefstring{\@@notetype}{familiar}%
5545       {\csuse{#1append@notesX}{\@@notesseries}}}%
5546       {}}%
5547   }%
5548 }%
5549 \expandafter\docsvlist\expandafter{#2}%
5550 }%
5551 %

```

`\do@Xfeet` `\do@Xfeet` is the code extending `\@makecol` to cater to the extra critical feet.

```

5552 \newcommand*{\do@Xfeet}{%
5553   \setbox\@outputbox \vbox{%
5554     \unvbox\@outputbox
5555     \@opXfeet}}
5556 %

```

`\@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`

```

5557 \newcommand{\append@Xnotes}[1]{%
5558   \ifvoid\csuse{#1footins}\else%
5559     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
5560     \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
5561     \print@Xnotes{#1}%
5562   \fi%
5563 }%
5564 %

```

The normal way to add one series, `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

5565 \newcommand\print@Xnotes[1]{%
5566   \xdef\@currentseries{#1}%
5567   \csuse{#1footstart}{#1}%
5568   \csuse{#1footgroup}{#1}%
5569 }%
5570 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

5571 \newcommand*{\@opXfeet}{%
5572   \unless\ifnocritical@%
5573     \gdef\firstXseries@{}%
5574     \def\do##1{%
5575       \append@Xnotes{##1}%
5576     }%
5577     \dolistloop{\@series}%
5578   \fi%

```

```
5579 }%
5580 %
```

\l@ddodoreintrafeet \l@ddodoreintrafeet is the code for catering for the extra footnotes within \@reinserts. We use the same category and ordering as in \l@ddoxtrafeet.

```
5581 \newcommand*\l@ddodoreintrafeet{%
5582   \IfStrEq{familiar-critical}{\@fnpos}
5583     {\@doreinfeetX\X@doreinfeet}%
5584     {%
5585       \IfStrEq{critical-familiar}{\@fnpos}%
5586       {\X@doreinfeet\@doreinfeetX}%
5587       {\@doreinfeetX\X@doreinfeet}%
5588     }%
5589 }
5590
5591 %
```

\X@doreinfeet \X@doreinfeet is the code for catering for the extra critical footnotes within \@reinserts.

```
5592 \newcommand*\X@doreinfeet{%
5593   \unless\ifnocritical@%
5594     \def\do##1{%
5595       \ifvoid\csuse{##1footins}\else%
5596         \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
5597       \fi}%
5598     \dolistloop{\@series}
5599   \fi%
5600 }
5601
5602 %
```

\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

```
5603 \newcommand{\append@notesX}[1]{%
5604   \ifvoid\csuse{footins#1}\else%
5605     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
5606     \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
5607     \print@notesX{#1}%
5608   \fi%
5609 }%
5610 %
```

The normal way to print one series of notes. \print@Xnotes is replaced by reledpar when using \Pages.

```
5611 \newcommand\print@notesX[1]{%
5612   \xdef\@currentseries{#1}%
```

```

5613 \csuse{footstart#1}{#1}%
5614 \csuse{footgroup#1}{#1}%
5615 }%
5616 %

```

We print all the series of notes by looping on them. We check before printing them that they are not voided.

```

5617 \newcommand*{\do@feetX}{%
5618   \unless\ifnofamiliar%
5619   \gdef\firstseriesX@{}%
5620   \setbox\@outputbox \vbox{%
5621     \unvbox\@outputbox%
5622     \def\do##1{%
5623       \append@notesX{##1}%
5624     }%
5625     \dolistloop{\@series}}%
5626   \fi%
5627 }%
5628
5629 \newcommand{\@doreinfeetX}{%
5630   \unless\ifnofamiliar%
5631   \def\do##1{%
5632     \ifvoid\csuse{footins##1}\else
5633     \insert%
5634       \csuse{footins##1}
5635       {\unvbox\csuse{footins##1}}%
5636     \fi%
5637   }%
5638   \dolistloop{\@series}%
5639   \fi%
5640 }%
5641
5642 %

```

XXII.2 Patching standard output's commands

The memoir class does not use the ‘standard’ versions of `\@makecol` and `\@reinserts`, due to its sidebar insert. We had better add that code if memoir is used. (It can be awkward dealing with `\if` code within `\if` code, so don't use `\ifl@dmemoir` here.)

```

5643 \@ifclassloaded{memoir}{%
5644   %

```

memoir is loaded so we use memoir's built in hooks.

```

5645   \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
5646   \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
5647 }{%
5648 %

```

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`.

```

5649 \ifboolexpr{%
5650   test{\@ifpackageloaded{fancyhdr}}%
5651   and test {\ifdef{\latex@makecol}}%
5652 }{%
5653   \patchcmd%
5654     {\latex@makecol}%
5655     {\xdef\@freelist{\@freelist\@midlist}}%
5656     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5657     {}%
5658     {\led@error@fail@patch@@@makecol}%
5659   }{%
5660     \patchcmd%
5661       {\@makecol}%
5662       {\xdef\@freelist{\@freelist\@midlist}}%
5663       {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5664       {}%
5665       {\led@error@fail@patch@@@makecol}%
5666     }%
5667
5668   \patchcmd%
5669     {\@reinserts}%
5670     {\ifvbox}%
5671     {\l@ddodoreinextrafeet\ifvbox}%
5672     {}%
5673     {\led@error@fail@patch@@@reinserts}%
5674   }
5675
5676   %

```

It turns out that `\@docclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

5677 \newif\if@led@nofoot
5678
5679 %

```

```

5680 \@ifclassloaded{memoir}{%
5681   %

```

If the `memoir` class is loaded we hook into its modified `\@docclearpage`.

`\@mem@extranofeet`⁸² `\g@addto@macro{\@mem@extranofeet}{%%`

```

5683 \def\do#1{%
5684   \unless\ifnocritical@%
5685     \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
5686   \fi%
5687   \unless\ifnofamiliar@%
5688     \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
5689   \fi%
5690 }
5691 \dolistloop{\@series}%
5692 }%
5693 }{%
5694 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot%5 \newcommand*{\@led@testifnofoot}{%
\@doclearpage%6   \@led@nofoottrue%
5697   \ifvoid\footins\else%
5698     \@led@nofootfalse%
5699   \fi%
5700   \def\do##1{%
5701     \unless\ifnocritical@%
5702       \ifvoid\csuse{##1footins}\else%
5703         \@led@nofootfalse%
5704       \fi%
5705     \fi%
5706     \unless\ifnofamiliar@%
5707       \ifvoid\csuse{footins##1}\else%
5708         \@led@nofootfalse%
5709       \fi%
5710     \fi%
5711   }%
5712   \dolistloop{\@series}%
5713 }%
5714
5715 \pretocmd%
5716 {\@doclearpage}%
5717 {\@led@testifnofoot}%
5718 {}%
5719 {\led@error@fail@patch@@doclearpage}%
5720
5721 \patchcmd%
5722 {\@doclearpage}%
5723 {\ifvoid\footins}%
5724 {\if@led@nofoot}%
5725 {}%
5726 {\led@error@fail@patch@@doclearpage}%
5727
5728 }

```

```
5729
5730 %
```

XXIII Cross referencing

You can mark a place in the text using a command of the form `\edlabel{<foo>}`, and later refer to it using the label `<foo>` by typing `\edpageref{<foo>}`, or `\lineref{<foo>}` or `\sublineref{<foo>}` or `\pstartref`. These reference commands will produce, respectively, the page, line sub-line and pstart on which the `\edlabel{<foo>}` command occurred.

The reference macros warn you if a reference is made to an undefined label. If `{<foo>}` has been used as a label before, the `\edlabel{<foo>}` command will issue a complaint; subsequent `\edpageref` and `\edlineref` commands will refer to the latest occurrence of `\edlabel{<foo>}`.

\labelref@list Set up a new list, `\labelref@list`, to hold the page, line and sub-line numbers for each label.

```
5731 \list@create{\labelref@list}
5732 %
```

\zz@@@ A convenience macro to zero two labeling counters in one go.

```
5733 \newcommand*{\zz@@@}{000|000} % set two counters to zero in one go
5734
5735 %
```

\edlabel The `\edlabel` command first writes a `\@lab` macro to the `\linenum@out` file. It then checks to see that the `\labelref@list` actually has something in it (if not, it creates a dummy entry), and pops the next value for the current label, storing it in `\label@refs`. Finally it defines the label to be `\empty` so that any future check will turn up the fact that it has been used.³²

This version of the original edmac `\label` uses `\@bsphack` and `\@esphack` to eliminate extra space problems and also use the \TeX write methods for the `.aux` file.

Jesse Billett³³ found that the original code could be off by several pages. This version, hopefully cures that, and also allows for non-arabic page numbering.

```
5736 \newcommand*{\edlabel}[1]{%
5737   \ifl@dpairing\ifautopar%
5738     \strut%
5739   \fi\fi%
5740   \@bsphack%
5741   \ifbool{bool{ledRcol} or bool{ledRcol@}}{%
5742     \ifXnote%
```

³²The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

³³(jdb43@cam.ac.uk) via the ctt thread ‘ledmac cross referencing’, 25 August 2003.

```

5743 \protected@write\@auxout{}\%
5744 {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\the\c@pstartR|{#1}}}%
5745 \ifdef{\hypertarget}\%
5746 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5747 {}%
5748 \else%
5749 \write\linenum@outR{\string\@lab}%
5750 \ifx\labelref@listR\empty%
5751 \xdef\label@refs{\zz@@@}%
5752 \else%
5753 \gl@p\labelref@listR\to\label@refs%
5754 \fi%
5755 \ifvmode%
5756 \advancelabel@refs%
5757 \fi%
5758 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

5759 \protected@write\@auxout{}\%
5760 {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}}%
5761 \ifdef{\hypertarget}\%
5762 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5763 {}%
5764 \fi%
5765 }{%
5766 \ifXnote@%
5767 \protected@write\@auxout{}\%
5768 {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub|\the\c@pstartR|{#1}}}%
5769 \ifdef{\hypertarget}\%
5770 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5771 {}%
5772 \else%
5773 \write\linenum@outR{\string\@lab}%
5774 \ifx\labelref@listR\empty%
5775 \xdef\label@refs{\zz@@@}%
5776 \else%
5777 \gl@p\labelref@listR\to\label@refs%
5778 \fi%
5779 \ifvmode%
5780 \advancelabel@refs%
5781 \fi%
5782 \protected@write\@auxout{}\%
5783 {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstart
R|{#1}}}%
5784 \ifdef{\hypertarget}\%
5785 {\Hy@raisedlink{\hypertarget{#1}{}}}%

```

```

5786     {}%
5787     \fi%
5788   }%
5789 \@esphack}%
5790
5791 %

```

`\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.

```

5792 \newcounter{line}%
5793 \newcounter{subline}%
5794 \newcommand{\advancelabel@refs}{%
5795   \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
5796   \stepcounter{line}%
5797   \ifsublines@%
5798     \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
5799   %
5800   \stepcounter{subline}{1}%
5801   \def\label@refs{\theline|\thesubline}%
5802   \else%
5803     \def\label@refs{\theline|0}%
5804   \fi%
5805 }
5806 \def\labelrefsparseline#1|#2{#1}
5807 \def\labelrefsparsesubline#1|#2{#2}
5808 %

```

`\l@dmake@labels` The `\l@dmake@labels` macro gets executed when the labels file is read. For each label it defines a macro, whose name is made up partly from the label you supplied, that contains the page, line and sub-line numbers. But first it checks to see whether the label has already been used (and complains if it has).

The initial use of `\newcommand` is to catch if `\l@dmake@labels` has been previously defined (by a class or package).

#1 page number, #2 line number, #3 sub-line number, #4 pstart number, #5 label.

```

5808 \newcommand*{\l@dmake@labels}{%
5809 \def\l@dmake@labels#1|#2|#3|#4|#5{%
5810   \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
5811   else
5812     \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
5813     \fi
5814   \global\providetoggle{label@#5\ledRcol}%False is the default value of
5815   this toggle, which tell us if a label is linked to a right or a left side
5816   \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
5817   {#1|#2|#3|#4|\relax}%

```



```

5815 \ignorespaces}
5816
5817 %

```

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.

```

5818 \AtBeginDocument{%
5819 \def\l@dmake@labels#1|#2|#3|#4|#5{%
5820 }
5821
5822 %

```

\@lab The \@lab command, which appears in the \linenum@out file, appends the current values of page, line and sub-line to the \labelref@list. These values are defined by the earlier \@page, \@nl, and the \sub@on and \sub@off commands appearing in the \linenum@out file.

TeX uses the page counter for page numbers. However, it appears that this is not the right place to grab the page number. That task is now done in the \edlabel macro. This version of \@lab appends just the current line and sub-line numbers to \labelref@list.

```

5823
5824 \newcommand*{\@lab}{%
5825 \ifledRcol
5826 \xright@appenditem{\linenumr@p{\line@numR}}|{%
5827 \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi}%
5828 \to\labelref@listR
5829 \else
5830 \xright@appenditem{\linenumr@p{\line@num}}|{%
5831 \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi}%
5832 \to\labelref@list
5833 \fi}
5834 %

```

\applabel \applabel, if called in \edtext will insert automatically both a start and an end label for the current edtext lines.

```

5835 \newcommand*{\applabel}[1]{%
5836 \if@edtext@secondarg@%
5837 %

```

Label should not be already defined.

```

5838 \ifcsundef{the@label#1}{%
5839 \csdef{the@label#1}{\applabel}%
5840 }%
5841 {%
5842 \led@warn@DuplicateLabel{#1 (\applabel)}%
5843 }%
5844 %

```

Parse the `\edtext` line numbers.

```
5845 \expandafter\l@dp@rsefootsspec\l@d@nums|%
5846 %
```

Use the \TeX standard hack for label.

```
5847 \@bsphack%
5848 %
```

And now, write the data in the auxiliary file.

```
5849 \ifledRcol%
5850 \protected@write\@auxout{}%
5851 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstartR|{#1:start}}%
5852 \ifdef{\hypertarget}%
5853 {\Hy@raisedlink{\hypertarget{#1:start}}}%
5854 {}%
5855 \protected@write\@auxout{}%
5856 {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|\the\c@pstartR|{#1:end}}%
5857 \else%
5858 \protected@write\@auxout{}%
5859 {\string\l@dmake@labels\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstart|{#1:start}}%
5860 \ifdef{\hypertarget}%
5861 {\Hy@raisedlink{\hypertarget{#1:start}}}%
5862 {}%
5863 \protected@write\@auxout{}%
5864 {\string\l@dmake@labels\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|\the\c@pstart|{#1:end}}%
5865 \fi%
5866 %
```

Use the \TeX standard hack for label.

```
5867 \@esphack%
5868 %
```

Warning if `\applabel` is called outside of `\edtext`.

```
5869 \else%
5870 \led@warn@AppLabelOutSecondArgEdtext{#1}%
5871 \fi%
5872 %
```

End of `\applabel`

```
5873 }%
5874 %
```

`\edlabels` `\edlabelS` and `\edlabelE` are just used to mark the beginning and the end of a passage.
`\edlabelE`
`\edlabelSE`

```

5875 \newcommand{\edlabelS}[1]{%
5876   \edlabel{#1:start}%
5877 }
5878 \newcommand{\edlabelE}[1]{%
5879   \edlabel{#1:end}%
5880 }
5881 \newcommand{\edlabelSE}[1]{%
5882   \edlabelS{#1}%
5883   \edlabelE{#1}%
5884 }
5885 %

```

\wrap@edcrossref \wrap@edcrossref is called around all reledmac crossref commands, except those which start with x. It adds the hyperlink.

```

5886 \newrobustcmd{\wrap@edcrossref}[2]{%
5887   \ifdef{\hyperlink}%
5888     {\hyperlink{#1}{#2}}%
5889     {#2}%
5890 }
5891 %

```

\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.

```

5892 \newcommand*{\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
5893 \newcommand*{\xpageref}[1]{\l@dgetref@num{1}{#1}}
5894
5895 %

```

\edlineref If the specified label exists, \lineref gives its line number.

\xlineref

```

5896 \newcommand*{\edlineref}[1]{%
5897   \l@dref@undefined{#1}%
5898   \wrap@edcrossref{#1}{%
5899     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
5900     \iftoggle{label@#1@ledRcol}%
5901       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
5902       {\linenumrep{\l@dgetref@num{2}{#1}}}%
5903     \xflagref{#1}%
5904   }%

```

```

5905 }%
5906 \newcommand*{\xlineref}[1]{\l@getref@num{2}{#1}}%
5907
5908 %

```

\sublineref If the specified label exists, \sublineref gives its sub-line number.

```

\sublineref
\newcommand*{\sublineref}[1]{%
5909   \l@dref@undefined{#1}%
5910   \wrap@edcrossref{#1}{%
5911     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
5912     \iftoggle{label@#1@ledRcol}%
5913       {\sublinenumrepR{\l@getref@num{3}{#1}}}%
5914       {\sublinenumrep{\l@getref@num{3}{#1}}}%
5915     }%
5916   }%
5917 }%
5918 \newcommand*{\xsublineref}[1]{\l@getref@num{3}{#1}}
5919
5920 %

```

\pstartref If the specified label exists, \pstartref gives its pstart number.

```

\pstartref
\newcommand*{\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@getref@num{4}{#1}}}
5921
5922 \newcommand*{\xpstartref}[1]{\l@getref@num{4}{#1}}
5923
5924 %

```

\xflagref \xflagref finds the side flag of any ref defined with \edlabel.

```

5925 \newcommand*{\xflagref}[1]{\l@getref@num{5}{#1}}
5926 %

```

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.

```

5927 \newcommand*{\l@dref@undefined}[1]{%
5928   \expandafter\ifx\csname the@label#1\endcsname\relax
5929     \led@warn@RefUndefined{#1}%
5930   \fi}
5931
5932 %

```

`\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.

```

5933 \newcommand*{\l@dgetref@num}[2]{%
5934   \expandafter
5935   \ifx\csname the@label#2\endcsname \relax
5936     000%
5937   \else
5938     \expandafter\expandafter\expandafter
5939     \l@dlabel@parse\csname the@label#2\endcsname|#1%
5940   \fi}
5941
5942 %

```

`\l@dlabel@parse` Notice that we slipped another `|` delimiter into the penultimate line of `\l@dgetref@num`, to keep the ‘switch-number’ separate from the reference numbers. This `|` is used as another parameter delimiter by `\l@dlabel@parse`, which extracts the appropriate number from its first arguments. The `|`-delimited arguments consist of the expanded label-macro (three reference numbers), followed by the switch-number (1, 2, 3 or 4) which defines which of the earlier five numbers to pick out. (It was earlier given as the first argument of `\l@dgetref@num`.)

```

5943 \newcommand*{\l@dlabel@parse}{%
5944   \def\l@dlabel@parse#1|#2|#3|#4|#5|#6{%
5945     \ifcase #6%
5946       \or #1%
5947       \or #2%
5948       \or #3%
5949       \or #4%
5950       \or #5%
5951     \fi}
5952 %

```

`\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.

```

5953 \newcommand*{\xxref}[2]{%
5954   {%
5955     \expandafter\ifx\csname the@label#1\endcsname \relax%
5956     \expandafter\let\csname the@@label#1\endcsname\zz@@@%

```

```

5957 \else%
5958 \expandafter\def\csname the@@label#1\endcsname{\l@getref@num
{1}{#1}|\l@getref@num{2}{#1}|\l@getref@num{3}{#1}}%
5959 \fi%
5960 \expandafter\ifx\csname the@label#2\endcsname \relax%
5961 \expandafter\let\csname the@@label#2\endcsname\zz@@@%
5962 \else%
5963 \expandafter\def\csname the@@label#2\endcsname{\l@getref@num
{1}{#2}|\l@getref@num{2}{#2}|\l@getref@num{3}{#2}}%
5964 \fi%
5965 \letcs{\@tempa}{the@@label#1}%
5966 \letcs{\@tempb}{the@@label#2}%
5967 \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@start{#1}}%
5968 \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@end{#2}}%
5969 \linenum{\@tempa|%
5970 \@tempb}}}%
5971
5972 %

```

`\appref` `\SEref`, `\apprefwithpage`, `\SErefwithpage` and `\SEonlypage` print cross-ref to some start / end lines defined by specific commands. It prints the lines as they should be printed in the apparatus (critical notes for not suffixed versions, endnotes for suffixed versions).

`\SErefwithpage` Here we define hooks similar to some those related to critical footnotes or endnotes. So, first declare the default value of the hooks for the pseudo-series. Also declare the internal toggle which are switch by `reledmac`.

```

5973 \def\Xtwolines@appref{}%
5974 \def\Xtwolines@SEref{}%
5975
5976 \def\Xmorethantwolines@appref{}%
5977 \def\Xmorethantwolines@SEref{}%
5978
5979 \def\Xlinerangeseparator@appref{\endashchar}%
5980 \def\Xlinerangeseparator@SEref{\endashchar}%
5981
5982 \def\Xsublinesep@appref{\fullstop}%
5983 \def\Xsublinesep@SEref{\fullstop}%
5984
5985 \def\Xpagelinesep@appref{\fullstop}%
5986 \def\Xpagelinesep@SEref{\fullstop}%
5987
5988
5989 \newtoggle{Xtwolinesbutnotmore@appref}%
5990 \newtoggle{Xtwolinesbutnotmore@SEref}%
5991
5992 \newtoggle{Xtwolinesonlyinsamepage@appref}%
5993
5994 \newtoggle{Xtwolinesonlyinsamepage@SEref}%

```

```

5995 \newtoggle{Xlineflag@appref}%
5996 \toggletrue{Xlineflag@appref}%Here exception
5997 \newtoggle{Xlineflag@SEref}%
5998 \toggletrue{Xlineflag@SEref}%Here exception
6000
6001 \def\Xendtwolines@apprefwithpage{}%
6002 \def\Xendtwolines@SErefwithpage{}%
6003
6004 \def\Xendmoreethantwolines@apprefwithpage{}%
6005 \def\Xendmoreethantwolines@SErefwithpage{}%
6006
6007 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
6008 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
6009 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
6010
6011 \def\Xendbeforepagenumber@apprefwithpage{p.}%
6012 \def\Xendbeforepagenumber@SErefwithpage{p.}%
6013 \def\Xendbeforepagenumber@SEonlypage{p.}%
6014
6015 \def\Xendafterpagenumber@apprefwithpage{}%
6016 \def\Xendafterpagenumber@SErefwithpage{}%
6017
6018
6019 \def\Xendlineprefixsingle@apprefwithpage{}%
6020 \def\Xendlineprefixsingle@SErefwithpage{}%
6021
6022 \def\Xendlineprefixmore@apprefwithpage{}%
6023 \def\Xendlineprefixmore@SErefwithpage{}%
6024
6025 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
6026 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
6027
6028 \def\Xendsublinesep@apprefwithpage{\fullstop}%
6029 \def\Xendsublinesep@SErefwithpage{\fullstop}%
6030
6031 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
6032 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
6033
6034 \newtoggle{Xendlineflag@apprefwithpage}
6035 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
6036 \newtoggle{Xendlineflag@SErefwithpage}
6037 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
6038
6039 %

```

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.

```

6040

```

```

6041 \gdef\Xboxstartlinenum@appref{Opt}
6042 \gdef\Xboxstartlinenum@SEref{Opt}
6043
6044 \gdef\Xboxendlinenum@appref{Opt}
6045 \gdef\Xboxendlinenum@SEref{Opt}
6046
6047 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}
6048 \gdef\Xendboxstartlinenum@SErefwithpage{Opt}
6049
6050 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
6051 \gdef\Xendboxendlinenum@SErefwithpage{Opt}
6052
6053 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
6054 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
6055
6056 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
6057 \newtoggle{Xendpagenumberonlyfirstifsingle@SErefwithpage}
6058
6059 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
6060 \newtoggle{Xendpagenumberonlyfirstintwo@SErefwithpage}
6061
6062 \gdef\Xendsympagenum@apprefwithpage{}
6063 \gdef\Xendsympagenum@SErefwithpage{}
6064
6065 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
6066 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
6067
6068 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

6069 \newcommand\@apprefprefixsingle{}%
6070 \newcommand\@SErefprefixsingle{}%
6071
6072 \newcommand\@apprefprefixmore{}%
6073 \newcommand\@SErefprefixmore{}%
6074
6075 \newcommand{\setapprefprefixsingle}[1]{%
6076   \gdef\@apprefprefixsingle{#1}%
6077 }
6078 \newcommand{\setSErefprefixsingle}[1]{%
6079   \gdef\@SErefprefixsingle{#1}%
6080 }
6081
6082 \newcommand{\setapprefprefixmore}[1]{%
6083   \gdef\@apprefprefixmore{#1}%
6084 }
6085 \newcommand{\setSErefprefixmore}[1]{%
6086   \gdef\@SErefprefixmore{#1}%
6087 }

```



```

6088 %
6089 %
And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.
6090 \let\setSErefonlypageprefixsingle\XendbeforepagenumberSErefonlypage%
6091 \newcommand{\setSErefonlypageprefixmore}[1]{%
6092   \gdef\SErefonlypage@prefixmore{#1}%
6093 }%
6094 %

```

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.

```

6095 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
6096 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
6097
6098 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
6099 {#1}{#2}{appref}}
6100 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
6101 {#1}{#2}{SEref}}
6102 \newcommandx{\SErefonlypage}[2][1,usedefault]{\reformattedonlypage@
6103 {#1}{#2}{SEref}}
6104
6105 \newcommand{\reformatted@}[3]{%
6106   \def\do##1{%
6107     \setkeys{mac}{truefootnoteoption}{##1}%
6108   }%
6109   \notblank{#1}{\docsvlist{#1}}}%
6110   \xdef\@currentseries{#3}%
6111   \ifcempty{@#3prefixmore}%
6112     {\@apprefprefixsingle}%
6113     {\IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}}%
6114     {\csuse{@#3prefixsingle}}%
6115     {\csuse{@#3prefixmore}}%
6116   }%
6117   \ifboolexpr{%
6118     test{\ifcsundef{the@label#2:start}}%
6119     or test{\ifcsundef{the@label#2:end}}%
6120   }%
6121   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6122   {%
6123     \def\@this@crossref@start{#2:start}%
6124     \def\@this@crossref@end{#2:end}%
6125     \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|\%

```

```

6126 \undef\@this@crossref@end%
6127 \undef\@this@crossref@start%
6128 }%
6129 \def\do##1{%
6130 \setkeys[mac]{falsefootnoteoption}{##1}%
6131 }%
6132 \notblank{#1}{\docsvlist{#1}}{ }%
6133 }%
6134
6135 \newcommand{\reformattedwithpage@}[3]{%
6136 \def\do##1{%
6137 \setkeys[mac]{truefootnoteoption}{##1}%
6138 }%
6139 \notblank{#1}{\docsvlist{#1}}{ }%
6140 \xdef\@currentseries{#3withpage}%
6141 \ifboolexpr{%
6142 test{\ifcsundef{the@label#2:start}}%
6143 or test{\ifcsundef{the@label#2:end}}%
6144 }%
6145 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6146 {%
6147 \def\@this@crossref@start{#2:start}%
6148 \def\@this@crossref@end{#2:end}%
6149 \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|}%
6150 \undef\@this@crossref@end%
6151 \undef\@this@crossref@start%
6152 }%
6153 \def\do##1{%
6154 \setkeys[mac]{falsefootnoteoption}{##1}%
6155 }%
6156 \notblank{#1}{\docsvlist{#1}}{ }%
6157 }%
6158
6159 \newcommand{\reformattedonlypage@}[3]{%
6160 \def\do##1{%
6161 \setkeys[mac]{truefootnoteoption}{##1}%
6162 }%
6163 \notblank{#1}{\docsvlist{#1}}{ }%
6164 \xdef\@currentseries{#3onlypage}%
6165 \ifboolexpr{%
6166 test{\ifcsundef{the@label#2:start}}%
6167 or test{\ifcsundef{the@label#2:end}}%
6168 }%
6169 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6170 {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
6171 {%
6172 \printnnum{%
6173 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%

```

```

6174 }%
6175 }%
6176 {%
6177 \ifcvoid{#3onlypage@prefixmore}%
6178 {}%
6179 {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
6180 \ifdefined\linangesep%
6181 \printnpnum{%
6182 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
6183 \linangesep%
6184 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
6185 }%
6186 \else%
6187 \printnpnum{%
6188 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
6189 \csuse{Xendlinangeseparator@{\@currentseries}%
6190 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
6191 }%
6192 \fi%
6193 }%
6194 }%
6195 \def\do##1{%
6196 \setkeys[mac]{falsefootnoteoption}{##1}%
6197 }%
6198 \notblank{#1}{\docsvlist{#1}}}%
6199 }%
6200 %

```

\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`.

```

6201 \newcommand*{\edmakelabel}[2]{\expandafter\edef\csname the@label#1\
6202 endcsname{#2}}
6203 %

```

(If you are only going to refer to such a label using `\xxref`, then you can omit entries in the same way as with `\linenum` (see VI.3 p. 128 and V.9 p. 95), since `\xxref` makes a call to `\linenum` in order to do its work.)

XXIII.1 Compatibility with xref

Here, we provide compatibility with the `xref` to enable `reledmac`’s cross-referencing to external documents. We assume that the user loads `xref` *before* `reledmac`, but uses `\externaldocument` *after* loading `reledmac`.

\XR@test First, we patch the `xr` macro `\XR@test`, which is called on every line of the external `.aux` file, in order to also call macros specific to `reledmac`.

```
6204 \pretocmd{\XR@test}%
6205   {\XR@test@mac+++#1#2#3#4+++}%
6206   {}%
6207   {}%
6208   %
```

\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`.

```
6209 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
6210 %
```

\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, note that the `\XR@prefix` is not called here, but it is integrated directly in `\l@dmake@labels` and `\l@dmake@labelsR`.

```
6211 \long\def\xR@test@mac@test#1#2...{%The triple dots (NOT \ldots) are because
6212   of the line 22 of xr.sty v5.02 1994/05/28
6213   \ifx#1\l@dmake@labels%
6214     \l@dmake@labels#2%
6215   \else
6216     \ifx#1\l@dmake@labelsR%
6217       \l@dmake@labelsR #2%
6218     \fi%
6219   \fi%
6220   }%
```

XXIV Side notes

Regular `\marginpars` do not work inside numbered text — they do not produce any note but do put an extra unnumbered blank line into the text.

\@xympar Changing `\@xympar` a little at least ensures that `\marginpars` in numbered text do not disturb the flow.

```
6221 \pretocmd{\@xympar}%
6222   {\ifnumberedpar@
6223     \led@warn@NoMarginpars
6224     \@esphack
6225   \else}%
6226   {}%
6227   {}%
6228   %
6229 \apptocmd{\@xympar}%
6230   {\fi}%
```

```

6231 {}
6232 {}
6233
6234 %

```

We provide side notes as replacement for `\marginpar` in numbered text.

`\sidenote@margin` These are the sidenote equivalents to `\line@margin` and `\linenummargin` for specifying which margin. The default is the right margin (opposite to the default for line numbers). `\l@dgetsidenote@margin` returns the number associated to side note margin:

left: 0

right: 1

outer: 2

inner: 3

```

6235 \newcount\sidenote@margin
6236 \newcommand*{\sidenotemargin}[1]{\{%
6237   \l@dgetsidenote@margin{#1}%
6238   \ifnum\@l@dttempcntb>\m@ne
6239     \ifledRcol
6240       \global\sidenote@marginR=\@l@dttempcntb
6241     \else
6242       \global\sidenote@margin=\@l@dttempcntb
6243     \fi
6244   \fi}}
6245 \newcommand*{\l@dgetsidenote@margin}[1]{\%
6246   \def\@tempa{#1}\def\@tempb{left}%
6247   \ifx\@tempa\@tempb
6248     \@l@dttempcntb \z@
6249   \else
6250     \def\@tempb{right}%
6251     \ifx\@tempa\@tempb
6252       \@l@dttempcntb \@ne
6253     \else
6254       \def\@tempb{outer}%
6255       \ifx\@tempa\@tempb
6256         \@l@dttempcntb \tw@
6257       \else
6258         \def\@tempb{inner}%
6259         \ifx\@tempa\@tempb
6260           \@l@dttempcntb \thr@@
6261         \else
6262           \led@warn@BadSidenotemargin
6263           \@l@dttempcntb \m@ne
6264         \fi

```

```

6265     \fi
6266     \fi
6267     \fi}
6268 \sidenotemargin{right}
6269
6270 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
6271 \newbox\l@dlp@rbox
6272 \newbox\l@drp@rbox
6273
6274 %

```

`\ledlsnotewidth` These specify the width of the left/right boxes (initialised to `\marginparwidth`), their
`\ledrsnotewidth` distance from the text (initialised to `\linenumsep`), and the fonts used.

```

\ledlsnotesep
6275 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
6276 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
6277 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup
6278 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
6279 \newcommand*{\ledlsnotefontsetup}{\raggedleft\footnotesize}
6280 \newcommand*{\ledrsnotefontsetup}{\raggedright\footnotesize}
6281
6282 %

```

`\ledleftnote` `\ledleftnote`, `\ledrightnote`, `\ledinnernote`, `\ledouternote` are the user com-
`\ledrightnote` mands for left, right, inner and outer sidenotes. The two last one are just alias for the
`\ledinnernote` two first one, depending of the page number. `\ledsidenote{<text>}` is the command
`\ledouternote` for a moveable sidenote.

```

\ledsidenote
6283 \newcommand*{\ledleftnote}[1]{\edtext{}{\l@dlsnote{#1}}}
6284 \newcommand*{\ledrightnote}[1]{\edtext{}{\l@drsnote{#1}}}
6285 \newcommand*{\ledsidenote}[1]{\edtext{}{\l@dcsnote{#1}}}%
6286 \newcommand*{\ledinnernote}[1]{\edtext{}{\l@disnote{#1}}}%
6287 \newcommand*{\ledouternote}[1]{\edtext{}{\l@dosnote{#1}}}%
6288 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent
`\l@drsnote` of the critical footnotes code.

```

\l@dcsnote
6289 \newif\ifrightrightnoteup
\l@desnote
6290 \rightrightnoteuptrue
\l@disnote
6291
6292 \newcommand*{\l@dlsnote}[1]{%
6293     \begingroup%
6294     \newcommand{\content}{#1}%
6295     \ifnumberedpar@
6296     \ifledRcol%
6297         \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%

```

```

6298         \to\inserts@listR
6299     \global\advance\insert@countR \@ne%
6300 \else%
6301     \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}}%
6302         \to\inserts@list
6303     \global\advance\insert@count \@ne%
6304 \fi
6305 \fi%
6306 \ignorespaces%
6307 \endgroup%
6308 }%
6309
6310 \newcommand*{\l@drsnote}[1]{%
6311     \begingroup%
6312     \newcommand{\content}{#1}%
6313     \ifnumberedpar@
6314         \ifledRcol%
6315             \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
6316                 \to\inserts@listR
6317             \global\advance\insert@countR \@ne%
6318         \else%
6319             \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
6320                 \to\inserts@list
6321             \global\advance\insert@count \@ne%
6322         \fi
6323     \fi\ignorespaces%
6324 \endgroup%
6325 }%
6326
6327 \newcommand*{\l@dcsnote}[1]{%
6328     \begingroup%
6329     \newcommand{\content}{#1}%
6330     \ifnumberedpar@
6331         \ifledRcol%
6332             \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
6333                 \to\inserts@listR
6334             \global\advance\insert@countR \@ne%
6335         \else%
6336             \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
6337                 \to\inserts@list
6338             \global\advance\insert@count \@ne%
6339         \fi
6340     \fi\ignorespaces%
6341 \endgroup%
6342 }%
6343
6344 \newcommand*{\l@disnote}[1]{%
6345     \begingroup%
6346     \newcommand{\content}{#1}%
6347     \ifnumberedpar@

```

```

6348 \ifledRcol%
6349 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
6350 \to\inserts@listR%
6351 \global\advance\insert@countR \@ne%
6352 \else%
6353 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
6354 \to\inserts@list%
6355 \global\advance\insert@count \@ne%
6356 \fi%
6357 \fi\ignorespaces%
6358 \endgroup%
6359 }%
6360
6361 \newcommand*{\l@dosnote}[1]{%
6362 \begingroup%
6363 \newcommand{\content}{#1}%
6364 \ifnumberedpar%
6365 \ifledRcol%
6366 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
6367 \to\inserts@listR%
6368 \global\advance\insert@countR \@ne%
6369 \else%
6370 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
6371 \to\inserts@list%
6372 \global\advance\insert@count \@ne%
6373 \fi%
6374 \fi\ignorespaces%
6375 \endgroup%
6376 }%
6377
6378 %

```

\vl@dlsnote Put the left/right text into boxes, but just save the moveable text. **\l@dcsnotetext**, **\vl@drsnote** **\l@dcsnotetext@l** and **\l@dcsnotetext@r** are etoolbox's lists which will store the content of side notes. We store the content in lists, because we need to loop later on them, in case many sidenote co-exist for the same line. That is there some special test **\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.

```

6379 \newcommand*{\vl@dlsnote}[1]{%
6380   \ifledRcol{%
6381     \@l@tempcntb=\sidenote@marginR%
6382     \ifnum\@l@tempcntb>\@ne%
6383       \advance\@l@tempcntb by\page@numR%
6384     \fi%
6385   \else%
6386     \@l@tempcntb=\sidenote@margin%
6387     \ifnum\@l@tempcntb>\@ne%
6388       \advance\@l@tempcntb by\page@num%
6389     \fi%
6390   \fi%
6391   \ifodd\@l@tempcntb%
6392     \listgadd{\l@dcsnotetext@l}{#1}%
6393   \else%
6394     \listgadd{\l@dcsnotetext}{#1}%
6395   \fi
6396 }
6397 \newcommand*{\vl@drsnote}[1]{%
6398   \ifledRcol{%
6399     \@l@tempcntb=\sidenote@marginR%
6400     \ifnum\@l@tempcntb>\@ne%
6401       \advance\@l@tempcntb by\page@numR%
6402     \fi%
6403   \else%
6404     \@l@tempcntb=\sidenote@margin%
6405     \ifnum\@l@tempcntb>\@ne%
6406       \advance\@l@tempcntb by\page@num%
6407     \fi%
6408   \fi%
6409   \ifodd\@l@tempcntb%
6410     \listgadd{\l@dcsnotetext}{#1}%
6411   \else%
6412     \listgadd{\l@dcsnotetext@r}{#1}%
6413   \fi%
6414 }
6415 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
6416
6417 \newcommand{\vl@disnote}[1]{%
6418   \ifledRcol{%
6419     \@tempcnta=\page@numR%
6420   \else%
6421     \@tempcnta=\page@num%
6422   \fi%
6423   \ifodd\@tempcnta% ODD => right page => inner side = left side
6424     \vl@dlsnote{#1}%
6425   \else%
```

```

6426 \vl@drsnote{#1}%
6427 \fi%
6428 }%
6429
6430 \newcommand{\vl@dosnote}[1]{%
6431 \ifledRcol%
6432 \@tempcnta=\page@numR%
6433 \else%
6434 \@tempcnta=\page@num%
6435 \fi%
6436 \ifodd\@tempcnta% ODD => right page => outer side = right side
6437 \vl@drsnote{#1}%
6438 \else%
6439 \vl@dlsnote{#1}%
6440 \fi%
6441 }%
6442
6443 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box.
`\setl@drpr@box` And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

6444 \newcommand*{\setl@dlp@rbox}[1]{%
6445 \begingroup%
6446 \parindent\z@\hsize=\ledlsnotewidth%
6447 \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
6448 \global\setbox\l@dlp@rbox%
6449 \ifleftnoteup%
6450 =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
6451 \else%
6452 =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}}%
6453 \fi%
6454 \endgroup%
6455 }
6456
6457 \newcommand*{\setl@drp@rbox}[1]{%
6458 \begingroup%
6459 \parindent\z@\hsize=\ledrsnotewidth%
6460 \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
6461 \global\setbox\l@drp@rbox%
6462 \ifrightrightnoteup%
6463 =\vbox to\z@{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
6464 \else%

```

```

6465     =\vbox to0.7\baselineskip{{\ledrsnotefontsetup\strut#1\vss}}}%
6466     \fi%
6467     \endgroup%
6468 }%
6469 \newif\ifleftnoteup
6470 \leftnoteuptrue
6471 %

```

\@sidenotesep This macro is used to separate sidenotes of the same line.

```

6472 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
6473 \newcommand{\@sidenotesep}{, }
6474 %

```

\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.

```

6475 \newcommand*{\affixside@note}{%
6476     \prepare@edindex@for@note{\the\page@num|\the\line@num|\the\subline@num|\
the\page@num|\the\line@num|\the\subline@num|}%
6477     \def\sidenotecontent@{}%
6478     \numgdef{\itemcount@}{0}%
6479     \def\do##1{%
6480         \ifnumequal{\itemcount@}{0}%
6481             {%
6482                 \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
6483             {\appto\sidenotecontent@{\@sidenotesep ##1}%
6484             }%
6485             \numgdef{\itemcount@}{\itemcount@+\@ne}%
6486         }%
6487     \dolistloop{\l@dcnotesetext}%
6488     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
6489 %

```

And we do the same for left and right notes (not movable).

```

6490 \gdef\@templ@d{%
6491 \gdef\@templ@n{\l@dcnotesetext\l@dcnotesetext@l\l@dcnotesetext@r}%
6492 \ifx\@templ@d\@templ@n \else%
6493     \if@twocolumn%
6494         \if@firstcolumn%
6495             \setl@dlp@rbox{##1}{\sidenotecontent@}%
6496         \else%
6497             \setl@drp@rbox{\sidenotecontent@}%
6498         \fi%
6499     \else%

```

```

6500 \l@dttempcntb=\sidenote@margin%
6501 \ifnum\l@dttempcntb>\@ne%
6502 \advance\l@dttempcntb by\page@num%
6503 \fi%
6504 \ifodd\l@dttempcntb%
6505 \setl@drp@rbox{\sidenotecontent@}%
6506 \gdef\sidenotecontent@{}%
6507 \numgdef{\itemcount@}{0}%
6508 \dolistloop{\l@dcsnotetext@l}%
6509 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
6510 \setl@dlp@rbox{\sidenotecontent@}%
6511 \else%
6512 \setl@dlp@rbox{\sidenotecontent@}%
6513 \gdef\sidenotecontent@{}%
6514 \numgdef{\itemcount@}{0}%
6515 \dolistloop{\l@dcsnotetext@r}%
6516 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
6517 \setl@drp@rbox{\sidenotecontent@}%
6518 \fi%
6519 \fi%
6520 \fi%
6521 \advance\@edindex@fornote@m@ne%
6522 }
6523 %

```

XXV Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfeetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage`.
`\l@dfeetendmini` They can be extended to handle other things if necessary.

```

6524 \ifnoledgroup@else%
6525 \newcommand*{\l@dfeetbeginmini}{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
6526 \newcommand*{\l@dfeetendmini}{%
6527 \IfStrEq{critical-familiar}{\@mpfnpos}%
6528 {\l@dedendmini\l@dfamendmini}%
6529 {%
6530 \IfStrEq{familiar-critical}{\@mpfnpos}%
6531 {\l@dfamendmini\l@dedendmini}%
6532 {\do@feet@custom@order{mp@}{\@mpfnpos}}%
6533 }%
6534 }%
6535 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

`\l@dedendmini`

`\mp@append@Xnotes`

```

6536 \newcommand*{\l@dedbeginmini}{%
6537   \unless\ifnocritical%
6538     \def\do##1{%
6539       \csletcs{v##1footnote}{mpv##1footnote}%
6540     }%
6541     \dolistloop{\@series}%
6542   \fi%
6543 }
6544 \newcommand*{\l@dedendmini}{%
6545   \unless\ifnocritical%
6546     \ifl@dpairing%
6547       \ifledRcol%
6548         \flush@notesR%
6549       \else%
6550         \flush@notes%
6551       \fi%
6552     \fi
6553     \def\do##1{%
6554       \mp@append@Xnotes{##1}%
6555     }%
6556     \dolistloop{\@series}%
6557   \fi%
6558 }%
6559 \newcommand{\mp@append@Xnotes}[1]{%
6560   \ifvoid\csuse{mp#1footins}\else%
6561     \ifl@dpairing%
6562       \ifparledgroup%
6563         \ifledRcol%
6564           \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
6565         \else%
6566           \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\skip
\@nameuse{mp#1footins}}%
6567         \fi%
6568       \fi%
6569     \fi%
6570     \ifcsstring{series@display#1}{paragraph}{}{%
6571       \setbox\@nameuse{mp#1footins}=\vbox{%
6572         \csuse{Xnotefontsize@#1}%
6573         \ifcsdef{Xhsize}\csuse{series@display#1}@#1{%
6574           \hsize\csuse{Xhsize}\csuse{series@display#1}@#1}%
6575         }{%
6576           \noindent\csuse{Xtxtbeforenotes@#1}%
6577           \unvbox\@nameuse{mp#1footins}%
6578           \@parboxrestore%
6579         }%
6580       }%
6581       \csuse{mp#1footgroup}{#1}%

```

```

6582 \fi%
6583 }%
6584 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
6585 \newcommand*\l@dfambeginmini{%
6586 \unless\ifnofamiliar%
6587 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
6588 \dolistloop{\@series}%
6589 \fi%
6590 }%
6591
6592 \newcommand*\l@dfamendmini{%
6593 \unless\ifnofamiliar%
6594 \def\do##1{%
6595 \mp@append@notesX{##1}%
6596 }%
6597 \dolistloop{\@series}%
6598 \fi%
6599 }%
6600 \newcommand{\mp@append@notesX}[1]{%
6601 \ifvoid\csuse{mpfootins#1}\else%
6602 \csuse{mpfootgroup#1}{#1}%
6603 \fi%
6604 }%
6605 %

```

`\@iiminipage` This is our extended form of the kernel `\@iiminipage` defined in `ltboxes.dtx`.

```

6606 \patchcmd%
6607 {\@iiminipage}%
6608 {\let\@footnotetext\@mpfootnotetext}%
6609 {\let\@footnotetext\@mpfootnotetext\l@dfeetbeginmini}%
6610 {}%
6611 {\led@error@fail@patch@\@iiminipage}%
6612 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

6613 \patchcmd%
6614 {\endminipage}%
6615 {\footnoterule}%
6616 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
6617 {}%
6618 {\led@error@fail@patch@endminipage}%
6619
6620 \patchcmd%
6621 {\endminipage}%
6622 {\@minipagefalse}%

```

```

6623 {\l@dfeetendmini\@minipagefalse}%
6624 {}%
6625 {\led@error@fail@patch@endminipage}
6626
6627 %

```

`\l@dunboxmpfoot` `\@ldunboxmpfoot` insert normal footnotes for `ledgroup`.

`edgroup@beforenormalnotes`

```

6628 \newcommand*{\l@dunboxmpfoot}{%
6629   \vskip\skip\@mpfootins
6630   \normalcolor
6631   \footnoterule
6632   \l@advance@parledgroup@beforenormalnotes
6633   \unvbox\@mpfootins%
6634 }
6635 %

```

When using parallel `ledgroup`, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

6636 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
6637   \ifparledgroup
6638     \ifl@dpairing
6639       \ifledRcol
6640         \dimdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
6641       \else
6642         \dimdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
6643       \fi
6644     \fi
6645   \fi
6646 }
6647 %

```

`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`.

```

6648
6649 \newenvironment{ledgroup}{%
6650   \resetprevpage@num%
6651   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
6652   \let\@footnotetext\@mpfootnotetext
6653   \l@dfeetbeginmini%
6654 }{%
6655   \par
6656   \unskip
6657   \ifvoid\@mpfootins\else
6658     \l@dunboxmpfoot

```

```

6659 \fi
6660 \l@dfeetendmini%
6661 \@ledgroupfalse%
6662 }
6663
6664
6665 %

```

```

\ledgroupsize \begin{ledgroupsize}[\langle pos \rangle]{\langle width \rangle}

```

This environment puts footnotes at the end, even if that happens to be in the middle of a page, or crossing a page boundary. It is a sort of unboxed, variable $\langle width \rangle$ minipage. The optional $\langle pos \rangle$ controls the sideways position of numbered text.

```

6666 \newenvironment{ledgroupsize}[2][1]{%
6667 %

```

Set the various text measures.

```

6668 \hsize #2\relax
6669 %

```

Initialize fills for centering.

```

6670 \let\ledllfill\hfil
6671 \let\ledrlfill\hfil
6672 \def\@tempa{#1}\def\@tempb{1}%
6673 %

```

Left adjusted numbered lines

```

6674 \ifx\@tempa\@tempb
6675 \let\ledllfill\relax
6676 \else
6677 \def\@tempb{r}%
6678 \ifx\@tempa\@tempb
6679 %

```

Right adjusted numbered lines

```

6680 \let\ledrlfill\relax
6681 \fi
6682 \fi
6683 %

```

Set up the footnoting.

```

6684 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
6685 \let\@footnotetext\@mpfootnotetext
6686 \l@dfeetbeginmini%
6687 }{%
6688 \par
6689 \unskip
6690 \ifvoid\@mpfootins\else
6691 \l@dunboxmpfoot
6692 \fi

```



```

6693 \l@dfetendmini%
6694 }
6695
6696 %

```

Close the \ifnoledgroup@\else.

```

6697 \fi%
6698 %

```

`\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`.

```

6699 \newif\ifledgroupnotesL@
6700 \newif\ifledgroupnotesR@
6701 %

```

XXVI Indexing

Here is some code for indexing using page and line numbers.

XXVI.1 Looking on package order

First, ensure that `imakeidx` or `indextools` is loaded *before* `eledmac`.

```

6702 \AtBeginDocument{%
6703   \unless\ifl@imakeidx%
6704     \@ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
6705   %
6706   \fi%
6707   \unless\ifl@indextools%
6708     \@ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}
6709   %
6710   \fi%
6711   \unless\ifl@footmisc%
6712     \@ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
6713   %
6714   \fi%
6715 }
6716 %

```

XXVI.2 Auxiliary macros for `\edindex`

`\pagelinesep` In order to get a correct line number we have to use the label/ref mechanism. These
`\edindexlab` macros are for that.

```

\c@labidx
6714 \newcommand{\pagelinesep}{-}
6715 \newcommand{\edindexlab}{${&}}
6716 \newcounter{labidx}

```

```

6717 \setcounter{labidx}{0}
6718
6719 %

```

\doedindexlabel This macro sets an \edlabel.

```

6720 \newcommand{\doedindexlabel}{%
6721   \stepcounter{labidx}%
6722   \edlabel{\edindexlab\thelabidx}%
6723 }
6724
6725 %

```

\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.

```

6726 \newcounter{pageline}%
6727 \renewcommand{\thepageline}{%
6728   \thepage%
6729   \pagelinesep%
6730   \xlineref{\edindexlab\thelabidx}%
6731 }
6732 %

```

\thestartpageline These macros make up the page/line start/end number when the \edindex command is called in critical notes.
\theendpageline

```

6733 \newcommand{\thestartpageline}{%
6734   \l@dparsedstartpage%
6735   \pagelinesep%
6736   \l@dparsedstartline%
6737 }
6738 \newcommand{\theendpageline}{%
6739   \l@dparsedendpage%
6740   \pagelinesep%
6741   \l@dparsedendline%
6742 }
6743 %

```

XXVI.3 Code specific to \edindex in critical footnotes

\@edindex@fornote@ This counter is incremented at the beginning of each note (either a footnote or a side-note), and decremented at the end of each note. If its value is greater than 0, that means we are inside a note.

```

6744 \newcount\@edindex@fornote@
6745 %

```

`\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 `|`.

```

6746 \newcommand{\prepare@edindex@fornote}[1]{%
6747   \l@dp@rsefootspec#1|}%
6748   \advance\@edindex@fornote@\@ne%
6749 }
6750 %

```

`\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.

```

6751 \newcommand{\get@edindex@ledinnote@command}{%
6752   \ifxindy%
6753     \gdef\@ledinnote@command{%
6754       ledinnote\thelabidx%
6755     }%
6756     \ifxindyhyperref%
6757       \immediate\write\eledmac@xindy@out{%
6758         (define-attributes ("ledinnote\thelabidx"))^^J
6759         \space\space(markup-locref^^J
6760           \eledmacmarkuplocdepth^^J
6761           :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
6762         }{"^^J
6763           :close "}"^^J
6764           :attr "ledinnote\thelabidx"^^J
6765         )
6766       }%
6767     \else%
6768       \immediate\write\eledmac@xindy@out{%
6769         (define-attributes ("ledinnote\thelabidx"))^^J
6770         \space\space(markup-locref^^J
6771           \eledmacmarkuplocdepth^^J
6772           :open "\string\ledinnote{\@index@command}{"}^^J
6773           :close "}"^^J
6774           :attr "ledinnote\thelabidx"^^J
6775         )
6776       }%
6777     \fi%
6778   }%

```

If we do not use `xindy` option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

6778   \else%
6779     \gdef\@ledinnote@command{%
6780       ledinnote[\edindexlab\thelabidx]{\@index@command}%

```

```

6781 }%
6782 \fi%
6783 }
6784 %

```

XXVI.4 Analysis of command in indexed text

`\get@index@command` This macro is used to analyze if a text to be indexed has a command after a |.

```

6785 \def\get@index@command#1|#2+{%
6786   \gdef\@index@txt{#1}%
6787   \gdef\@index@command{#2}%
6788   \xdef\@index@parenthesis{%
6789     \IfBeginWith{\@index@command}{(}{%
6790       \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
6791       \global\let\@index@command\@index@command@%
6792       \xdef\@index@parenthesis{(%
6793     )}%
6794     \IfBeginWith{\@index@command}{)}{%
6795       \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
6796       \global\let\@index@command\@index@command@%
6797       \xdef\@index@parenthesis{)%
6798     }%
6799   }
6800 %

```

XXVI.5 Code for the formatted index

`\ledinnote` These macros are used to specify that an index reference points to a note. Arguments of `\ledinnote` are: #1 (optional): the label for the hyperlink, #2: command applied to the number, #3: the number itself.

`\ledinnotehyperpage`

`\ledinnotemark`

```

6801 \newcommandx{\ledinnote}[3][1,usedefault]{%
6802   \ifboolexpr{%
6803     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
6804     or%
6805     bool {xindyhyperref}%
6806   }%
6807   {%
6808     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
6809   }%
6810   {%
6811     \csuse{#2}{\ledinnotemark{#3}}%
6812   }%
6813 }%
6814 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage{#2}}}}%
6815 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
6816 %

```

XXVI.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
6817 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
6818 \ifl@imakeidx%
6819 \ifnum\@edindex@fornote@>\z@%
6820 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6821 \get@edindex@ledinnote@command%
6822 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
6823 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%
6824 \else%
6825 \get@edindex@hyperref{#2}%
6826 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
6827 \fi%
6828 \else%
6829 \ifnum\@edindex@fornote@>\z@%
6830 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6831 \get@edindex@ledinnote@command%
6832 \expandafter\protected@write\@indexfile{}%
6833 {\string\indexentry{\@index@txt|(\@ledinnote@command}{\thestartpageline}
}{%
6834 }%
6835 \expandafter\protected@write\@indexfile{}%
6836 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
}{%
6837 }%
6838 \else%
6839 \protected@write\@indexfile{}%
6840 {\string\indexentry{#2}{\thepageline}
}{%
6841 }%
6842 \fi%
6843 \fi%
6844 \endgroup
6845 \@esphack%
6846 }
6847 %

```

Need to add the definition of \edindex to \makeindex, and initialise \edindex to do nothing.

```

6848 \pretocmd{\makeindex}{%
6849   \def\edindex{%
6850     \ifboolexpr{bool{numbering} or bool{numberingR} or bool{
6851       l@dprintingpages} or bool{l@dprintingcolumns}}{%
6852       \@bsphack%
6853       \doedindexlabel%
6854       \begingroup%
6855       \@sanitize%
6856       \@wredindex%
6857     }%
6858     {\%
6859       \led@warn@edinde@outsidenumbering%
6860       \index%
6861     }%
6862   }%
6863 }%
6864 {\led@error@fail@patch@makeindex}%
6865 \newcommand{\edindex}[1]{\@bsphack\@esphack}
6866 %

```

XXVI.7 Hyperlink

\hyperlinkformat \hyperlinkformat command is to be used to have both a internal hyperlink and a format, when indexing.

```

6867 \newcommand{\hyperlinkformat}[3]{%
6868   \ifstrempy{#1}%
6869   {\hyperlink{#2}{#3}}%
6870   {\csuse{#1}{\hyperlink{#2}{#3}}}%
6871 }%
6872 %

```

\hyperlinkR \hyperlinkR command is to be used to create a internal hyperlink and \ledRflag, when indexing.

```

6873 \newcommand{\hyperlinkR}[2]{%
6874   \hyperlink{#1}{#2\@Rlineflag}%
6875 }%
6876 %
6877 %

```

\hyperlinkformatR \hyperlinkformatR command is to be used to create a internal hyperlink, a format and a \@Rlineflag, when indexing.

```

6878 \newcommand{\hyperlinkformatR}[3]{%
6879   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
6880 }%
6881 %
6882 %

```

`\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`).

```
6883 \newcommand{\get@edindex@hyperref}[1]{%
6884 %
```

We have to disable temporary spaces to work through a xstring bug (or feature?)

```
6885 \edef\temp@{%
6886 \catcode\ =9 %space need for catcode
6887 \detokenize{#1}%For active character in unicode
6888 \catcode\ =10 % space need for catcode
6889 }%
6890 %
```

Now, we define `\@edindex@hyperref` if the hyperindex of `hyperref` is enabled.

```
6891 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
6892 \IfSubStr{\temp@}{|}%
6893 {\get@index@command#1+%
6894 \ifledRcol%
6895 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6896 hyperlinkformatR{\@index@command}%
6897 {\edindexlab\thelabidx}}%
6898 \else%
6899 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6900 hyperlinkformat{\@index@command}%
6901 {\edindexlab\thelabidx}}%
6902 \fi%
6903 }%
6904 {\get@index@command#1|+%
6905 \ifledRcol%
6906 \gdef\@edindex@hyperref{|\hyperlinkR{\edindexlab\thelabidx}}%
6907 \else%
6908 \gdef\@edindex@hyperref{|\hyperlink{\edindexlab\thelabidx}}%
6909 \fi%
6910 }%
6911 }%
6912 %
```

```
6913 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
6914 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
6915 % \begin{macrocode}
6916 {\ifxindyhyperref%
6917 \IfSubStr{\temp@}{|}%
6918 {\get@index@command#1+%
6919 {\get@index@command#1|+%
6920 \gdef\@edindex@hyperref{|\eledmac\thelabidx}%
6921 %
```

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`.

```

6922 \IfStrEq{\@index@parenthesis}{(}{%
6923   {%
6924     \csxdef{xindy@parenthesis@\@index@txt}{\thelabidx}%
6925     \gdef\@edindex@hyperref{|\@edmac\thelabidx}%
6926   }%
6927   {}%
6928 %

```

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`.

```

6929 \IfStrEq{\@index@parenthesis}{)}{%
6930   {%
6931     \xdef\@edindex@hyperref{|\@edmac\cuse{xindy@parenthesis@\@index@txt}}%
6932     \global\csundef{xindy@parenthesis@\@index@txt}%
6933   }%
6934 %

```

Write in the `.xdy` file the attributes of the location.

```

6935   {%
6936     \immediate\write\eledmac@xindy@out{%
6937       (define-attributes ("eledmac\thelabidx"))^^J
6938       \space\space(markup-locref^^J
6939       \eledmacmarkuplocdepth^^J
6940       :open "\string\hyperlink%
6941         \ifledRcol R\fi%
6942         {\edindexlab\thelabidx}%
6943         {\ifdefempty{\@index@command}%
6944           }%
6945           {\@backslashchar\@index@command}%
6946         {"^^J
6947       :close "}")^^J
6948       :attr "eledmac\thelabidx"^^J
6949     )
6950   }%
6951 }%
6952 %

```

And now, in any other case.

```

6953 \else%
6954   \gdef\@index@txt{#1}%
6955   \gdef\@edindex@hyperref{}%
6956 \fi%
6957 }%
6958 }
6959 %

```


XXVI.8 ‘innote’ and ‘notenumber’ option of indextols package

`\led@set@index@fornote` The `\led@set@index@fornote` is called when a familiar footnote is inserted — and not when it is read — and changes the `\index` command depending of the option of the `indextols` package. Its only argument is the note series.

```

6960 \newcommand{\led@set@index@fornote}[1]{%
6961   \ifbool{indtl@innote}%
6962     {\let\index\nindex}%
6963     {}%
6964   \ifbool{indtl@notenumber}%
6965     {%
6966       \renewcommand{\index}[2][\indtl@jobname]{%
6967         \orig@@index[##1]{%
6968           ##2|innotenumber{\this@footnoteX@reading}%
6969         }%
6970       }%
6971     }%
6972     {}%
6973   }%
6974   %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

6975 \newcommand{\led@reinit@index@fornote}{%
6976   \ifbool{indtl@innote}%
6977     {\let\index\orig@@index}%
6978     {}%
6979   \ifbool{indtl@notenumber}%
6980     {\let\index\orig@@index}%
6981     {}%
6982   }%
6983   %

```

XXVII Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

6984 \AtBeginDocument{%
6985   \@ifpackageloaded{glossaries}{%
6986     %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

6987   \renewcommand{\do}[1]{%
6988     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
6989       \doedindexlabel%
6990       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%

```

```

6991 }%
6992 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,3,usedefault]{%
6993 \doedindexlabel%
6994 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
6995 }%
6996 }%
6997 \docsvlist{%
6998 gls,%
6999 GLs,%
7000 GLS,%
7001 glspl,%
7002 Glspl,%
7003 GLSpl,%
7004 glstext,%
7005 Glstext,%
7006 GLStext,%
7007 Glsfirst,%
7008 GLSfirst,%
7009 glsplural,%
7010 Glsplural,%
7011 GLSplural,%
7012 glsfirstplural,%
7013 Glsfirstplural,%
7014 GLSfirstplural,%
7015 glsname,%
7016 Glsname,%
7017 GLSname,%
7018 glssymbol,%
7019 Glssymbol,%
7020 GLSsymbol,%
7021 glsdesc,%
7022 Glsdesc,%
7023 GLSdesc,%
7024 glsuseri,%
7025 Glsuseri,%
7026 GLSuseri,%
7027 glsuserii,%
7028 Glsuserii,%
7029 GLSuserii,%
7030 glsuseriii,%
7031 Glsuseriii,%
7032 GLSuseriii,%
7033 glsuseriv,%
7034 Glsuseriv,%
7035 GLSuseriv,%
7036 glsuserv,%
7037 Glsuserv,%
7038 GLSuserv,%
7039 glsuservi,%

```

```

7040     Glsuservi,%
7041     GLSuservi%
7042     }%
7043 %

```

First those which arguments are [*options*]{*label*}{*link text*}.

```

7044     \renewcommand{\do}[1]{%
7045         \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,
usedefault]{%
7046             \doedindexlabel%
7047             \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
7048         }%
7049         \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,usedefault]{%
7050             \doedindexlabel%
7051             \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
7052         }%
7053     }%
7054     \docsvlist{glsdisp,glslink}%
7055     }{}%
7056 }%
7057 %

```

XXVIII Verse

The original code is principally Wayne Sullivan's code from `edstanza`. However, the code has been many time modified by Maïeul Rouquette in order to obtain new features and improved compatibility with `reledpar`.

XXVIII.1 Hanging symbol management

`\@hangingsymbol` The macro `\@hangingsymbol` is used to insert a symbol on each hanging of verses. It is set by user level macro `\sethangingsymbol`.
`\ifinstanza` For example, in french typographie the symbol is '['. We obtain it by the next code:

```
\sethangingsymbol{[,}
```

The `\ifinstanza` boolean is used to be sure that we are in a stanza part.

```

7058 \def\@hangingsymbol{}
7059 \newcommand*\sethangingsymbol[1]{%
7060     \gdef\@hangingsymbol{#1}%
7061 }%
7062 \newif\ifinstanza
7063 %

```

`\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.

```

7064 \newif\ifinserthangingsymbol
7065 \newcommand{\inserthangingsymbol}{%
7066 \ifinserthangingsymbol%
7067   \ifinstanza%
7068     \@hangingsymbol%
7069   \fi%
7070 \fi%
7071 }
7072 %

```

XXVIII.2 Using & character

\ampersand Within a stanza the `\&` macro is going to be usurped. We need an alias in case an `&` needs to be typeset in a stanza. Define it rather than letting it in case some other package has already defined it.

```

7073 \newcommand*{\ampersand}{\char`\&}
7074
7075 %

```

XXVIII.3 Code category setting

\stanza@count Before we can define the main macros we need to save and reset some category codes.
\stanzaindentbase To save the current values we use `\next` and `\body` from the `\loop` macro.

```

7076 \chardef\body=\catcode`\@
7077 \catcode`\@=11
7078 \chardef\next=\catcode`\&
7079 \catcode`\&=\active
7080
7081 %

```

XXVIII.4 Stanza count and indent

A count register is allocated for counting lines in a stanza; also allocated is a dimension register which is used to specify the base value for line indentation; all stanza indentations are multiples of this value. The default value of `\stanzaindentbase` is 20pt.

```

7082 \newcount\stanza@count
7083 \newlength{\stanzaindentbase}
7084 \setlength{\stanzaindentbase}{20pt}
7085
7086 %

```

`\strip@szacnt` The indentations of stanza lines are non-negative integer multiples of the unit called
`\setstanzavalues` `\stanzaindentbase`. To make it easier for the user to specify these numbers, some list macros are defined. These take numerical values in a list separated by commas and assign the values to special control sequences using `\mathchardef`. Though this does limit the range from 0 to 32767, it should suffice for most applications, including *penalties*, which will be discussed below.

```

7087 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
7088 \newcommand*\setstanzavalues}[2]{\def\@tempa{#2,,|}%
7089   \stanza@count\z@
7090   \def\next{\expandafter\strip@szacnt\@tempa
7091     \ifx\@tempb\empty\let\next\relax\else
7092     \expandafter\mathchardef\csname #1@\number\stanza@count
7093     \@endcsname\@tempb\relax
7094     \advance\stanza@count\@ne\fi\next}%
7095   \next}
7096
7097 %

```

`\setstanzaindents` In the original edmac, `\setstanzavalues{sza}{...}` had to be called to set the in-
`\setstanzapenalties` dents, and similarly `\setstanzavalues{szp}{...}` to set the penalties. `\setstanzaindents` and `\setstanzapenalties` macros are a convenience to give the user one less thing to worry about (misspelling the first argument).

```

7098 \newcommand*\setstanzaindents}[1]{\setstanzavalues{sza}{#1}}
7099 \newcommand*\setstanzapenalties}[1]{\setstanzavalues{szp}{#1}}
7100 %
7101 %

```

`\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.

```

7102 \newcounter{stanzaindentsrepetition}
7103 \newcount\stanza@modulo
7104
7105 \newcommand*\managestanza@modulo}[0]{%
7106   \advance\stanza@modulo\@ne%
7107   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
7108   \stanza@modulo\@ne%
7109   \fi%
7110 }
7111 %

```

`\stanzaindent` The macro `\stanzaindent`, when called at the beginning of a verse, changes the in-
`\stanzaindent*` dentation normally defined for this verse by `\setstanzaindent`. The starred version skips the current verse for the repetition of stanza indent.

```

7112 \newcommand{\stanzaindent}[1]{%
7113   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
7114   \ignorespaces%
7115 }%
7116 \WithSuffix\newcommand\stanzaindent*[1]{%
7117   \stanzaindent{#1}%
7118   \global\advance\stanza@modulo-\@ne%
7119   \ifnum\stanza@modulo=0%
7120     \global\stanza@modulo=\value{stanzaindentrepetition}%
7121   \fi%
7122   \ignorespaces%
7123 }%
7124 %

```

XXVIII.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza25 \newcounter{stanza}
7126 \renewcommand{\thestanza}{%
7127   \textbf{\arabic{stanza}}%
7128 }
7129 %

```

\ifnumberstanza Then, macro to activate automatically numbering of stanza.

```

7130 \newif\ifnumberstanza%
7131 %

```

\@insertstanzanumber Now, macro called at the first line of of verse of a stanza.

```

7132 \newcommand{\@insertstanzanumber}[0]{%
7133   \ifnumberstanza%
7134     \ifl@dpairing%
7135       \ifledRcol%
7136         \stanzanumwrapper{\thestanzaR}%
7137       \else%
7138         \stanzanumwrapper{\thestanzaL}%
7139       \fi%
7140     \else%
7141       \stanzanumwrapper{\thestanza}%
7142     \fi%
7143     \setline{1}%
7144   \fi%
7145 }%
7146 %

```

\@advancestanzanumber Also a command to advance the counter of stanza.

```

7147 \newcommand{\@advancestanzanumber}[0]{%
7148   \ifnumberstanza%
7149   \ifl@dpairing%
7150     \ifledRcol%
7151       \addtocounter{stanzaR}{1}%
7152     \else%
7153       \addtocounter{stanzaL}{1}%
7154     \fi%
7155   \else%
7156     \addtocounter{stanza}{1}%
7157   \fi%
7158 }%
7159 %
7160 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

7161 \newcommand{\stanzanumwrapper}[1]{%
7162   \flagstanza{#1}%
7163 }%
7164 %

```

XXVIII.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

7165 \newcommand{\printstanza}[0]{%
7166   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
7167     l@dprintingcolumns}}{%
7168     \ifledRcol%
7169       \thestanzaR%
7170     \else%
7171       \thestanzaL%
7172     \fi%
7173   }{%
7174     \thestanza%
7175   }%
7176 }%

```

XXVIII.7 Main work

`\stanza@line` Now we arrive at the main works. `\stanza@line` sets the indentation for the line and starts a numbered paragraph—each line is treated as a paragraph. `\stanza@hang` sets the hanging indentation to be used if the stanza line requires more than one print line.

`\sza@penalty` If it is known that each stanza line will fit on one print line, it is advisable to set the hanging indentation to zero. `\sza@penalty` places the specified penalty following each stanza line. By default, this facility is turned off so that no penalty is included. However,

the user may initiate these penalties to indicate good and bad places in the stanza for page breaking.

```

7177 \newcommandx{\stanza@line}[1][1]{
7178   \ifnum\value{stanza@indentsrepetition}=0
7179     \ifcsdef{sza@\number\stanza@count @}{%
7180       {%
7181         \parindent=\csname sza@\number\stanza@count @\endcsname\
stanza@indentbase%
7182       }{%
7183         \led@err@StanzaIndentNotDefined%
7184       }%
7185     }else
7186       \ifcsdef{sza@\number\stanza@modulo @}{%
7187         \parindent=\csname sza@\number\stanza@modulo @\endcsname\
stanza@indentbase%
7188         \managestanza@modulo%
7189       }%
7190       {%
7191         \led@err@StanzaIndentNotDefined%
7192       }%
7193     }fi
7194     \pstart[#1]\stanza@hang\ignorespaces}
7195 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
7196   \hangindent\expandafter
7197   \noexpand\csname sza@0@\endcsname\stanza@indentbase
7198   \hangafter\@ne}
7199 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
7200   \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
7201   \penalty\fi\count@}
7202 %

```

`\@startstanza` Now we have the components of the `\stanza` macro, which appears at the start of a 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 `\&`.

```

7203 \xdef\@startstanza[#1]{%
7204   \noexpand\instanzatrue\expandafter
7205   \begingroup%
7206   \catcode`\noexpand\&\active%
7207   \global\stanza@count\@ne\stanza@modulo\@ne
7208   \noexpand\ifnum\expandafter\noexpand
7209   \csname sza@0@\endcsname=z@\let\noexpand\stanza@hang\relax
7210   \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
7211   \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum

```



```

7212 \expandafter\noexpand\csname szp@0@\endcsname=\z@
7213 \let\noexpand\sza@penalty\relax\noexpand\fi%
7214 \def\noexpand&{%
7215     \noexpand\newverse[] []}%
7216 \def\noexpand\&{\noexpand\@stopstanza}%
7217 \noexpand\@advancestanzanumber%
7218 \noexpand\stanza@line[#1]%
7219 \noexpand\@insertstanzanumber%
7220 \let\par\relax\ignorespaces%No paragraph in verses
7221 }
7222
7223 \newcommandx{\stanza}[1][1,usedefault]{%
7224     \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}} and test{\ifstrempy
7225         {\@startstanza[\at@every@stanza]}}%
7226         {\@startstanza[#1]}}%
7227 }%
7228
7229 \newcommandx{\@stopstanza}[1][1,usedefault]{%
7230     \unskip%
7231     \endlock%
7232     \ifboolexpr{not test{\ifdefvoid{\at@every@stop@stanza}} and test{\
7233         \ifstrempy{#1}}}%
7234         {\pend[\at@every@stop@stanza]}}%
7235         {\pend[#1]}}%
7236     \endgroup%
7237     \instanzafalse%
7238 }
7239
7240 \newcommand{\AtEveryStopStanza}[1]{%
7241     \ifstrempy{#1}%
7242     {\xdef\at@every@stop@stanza{}}%
7243     {\gdef\at@every@stop@stanza{#1}}%
7244 }%
7245 \def\at@every@stop@stanza{}%
7246
7247 \newcommand{\AtEveryStanza}[1]{%
7248     \ifstrempy{#1}%
7249     {\xdef\at@every@stanza{}}%
7250     {\gdef\at@every@stanza{#1}}%
7251 }%
7252 \def\at@every@stanza{}%
7253
7254 \newcommandx*{\newverse}[2][1,2,usedefault]{%
7255     \unskip%
7256     \endlock\pend[#1]\sza@penalty\global%
7257     \advance\stanza@count\@ne\stanza@line[#2]%
7258 }
7259

```

7260 %

\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`.

```
7261 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
7262   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
7263
7264 %
```

XXVIII.8 Restore catcode and penalties

The ampersand & is used to mark the end of each stanza line, except the last, which is marked with `\&`. This means that `\halign` may not be used directly within a stanza line. This does not affect macros involving alignments defined outside `\stanza \&`. Since these macros usurp the control sequence `\&`, the replacement `\ampersand` is defined to be used if this symbol is needed in a stanza. Also we reset the modified category codes and initialize the penalty default.

```
7265 \catcode`\&=\next
7266 \catcode`\@=\body
7267 \setstanzavalues{szp}{0}
7268
7269 %
```

XXIX Apparatus of Manuscripts

XXIX.1 User level macro

\msdata The user level `\msdata` command only writes the manuscripts data in numbered auxiliary file.

```
7270 \newcommand{\msdata}[1]{%
7271   \leavevmode%
7272   \unless\ifstopmsdata@inserted{%
7273     \stopmsdata%
7274     \led@warning@msdatawithoutstop%
7275   \fi%
7276   \global\stopmsdata@inserted@false%
7277   \unless\ifledRcol%
7278     \protected@write\linenum@out{}{%
7279       \string\@msd{#1}%
7280     }%
7281   \else%
7282     \protected@write\linenum@outR{}{%
7283       \string\@msd{#1}%
7284     }%
7285   \fi%
}
```

```
7286 }%
7287 %
```

\stopmsdata The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```
7288 \newcommand{\stopmsdata}[0]{%
7289   \leavevmode%
7290   \unless\ifledRcol%
7291     \protected@write\linenum@out{}{%
7292       \string\stopmsd%
7293     }%
7294   \else%
7295     \protected@write\linenum@outR{}{%
7296       \string\stopmsd%
7297     }%
7298   \fi%
7299   \global\stopmsdata@inserted@true%
7300 }%
7301 %
```

\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`

```
7302 \newif\ifstopmsdata@inserted@%
7303 %
```

XXIX.2 Setting macro

Setting macros for the manuscripts apparatus tools is very easy: they just save their argument in an internal macro.

\setmsdataseries In which series of notes will be printed the apparatus of manuscripts?

```
7304 \newcommand{\setmsdataseries}[1]{%
7305   \gdef\@msdata@series{#1}%
7306 }%
7307 \def\@msdata@series{A}%
7308 %
```

\setmsdatalabel The label for the manuscripts data.

```
7309 \def\ms@data@label{Ms.}%
7310 \newcommand{\setmsdatalabel}[1]{%
7311   \gdef\ms@data@label{#1}%
7312 }%
7313 %
```

XXIX.3 Counters and lists

\@msd@c \@msd@c is a counter incremented at each \@msd read in auxiliary file.

```
7314 \numdef{\@msd@c}{0}
7315 \numdef{\@msd@cR}{0}
7316 %
```

\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.

```
7317 \numdef{\add@msd@c}{0}%
7318 \numdef{\add@msd@cR}{0}%
7319 %
```

\@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.

```
7320 \def\@msdata@list{}%
7321 %
```

XXIX.4 Auxiliary file macros

\@msd The \@msd macro is written in the auxiliary file. It just defines three macros by \@msdata macro, which allow us to know the manuscripts data, the line number and the absolute line number where it was called

It also stores the action code 1010 in the list of actions by line.

```
7322 \newcommand{\@msd}[1]{%
7323   \unless\ifledRcol%
7324     \global\numdef{\@msd@c}{\@msd@c+\@ne}%
7325     \csgdef{\@msdata@\@msd@c @data}{#1}%
7326     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
7327     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
7328     \xright@appenditem{\the\absline@num}\to\actionlines@list%
7329     \xright@appenditem{-1010}\to\actions@list%
7330   \else%
7331     \global\numdef{\@msd@cR}{\@msd@cR+\@ne}%
7332     \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
7333     \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
7334     \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
7335     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
7336     \xright@appenditem{-1010}\to\actions@listR%
7337   \fi%
7338 }%
7339 %
```

\@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.

```

7340 \newcommand{\@stopmsd}[0]{%
7341   \unless\ifledRcol%
7342     \ifcsundef{@msdata@\@msd@c @stoplinenumber}{%
7343       \csxdef{@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
7344       \csxdef{@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
7345     }{}%
7346   \else%
7347     \ifcsundef{@msdata@\@msd@cR @stoplinenumberR}{%
7348       \csxdef{@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
7349       \csxdef{@msdata@\@msd@cR @stoplinenumberR}{\the\line@numR}%
7350     }%
7351   }{}%
7352   \fi%
7353 }%
7354 %

```

XXIX.5 Action macro

\add@msdata \add@msdata is executed on each line when action code 1010 is seen. It will not insert immediately the manuscript data footnote, as action code are executed before the line be typeset, and, consequently, could be on the previous page. So it just stores the manuscript data footnote to \@msdata@list.

```

7355 \newcommand{\add@msdata}{%
7356   \bgroup%
7357   \normalfont%
7358   \unless\ifledRcol%
7359     \numgdef{\add@msd@c}{\add@msd@c+\@one}%
7360     \ifcsdef{@msdata@\add@msd@c @data}{%
7361       \letcs{\@data}{@msdata@\add@msd@c @data}%
7362       \edef\l@d@nums{%
7363         000| % Start page = we don't print it
7364         \csuse{@msdata@\add@msd@c @linenumber}| % Start line number
7365         000| % Start subline number, for now, not used
7366         000| % End page number, we don't print it
7367         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
7368         {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
7369         {\csuse{@lastline@forpage@\the\page@num}}}%Otherwiser, last
number of the page
7370         |%
7371         000| % End sub line number, for now, not used
7372         \edfont@info%Font
7373       }%
7374       \@msd@options@fullpagefalse%
7375       \if@firstlineofpage%Try if the data are for the full page. If yes
, will add options to the list.
7376       \unless\if@msdata@insertedfrompreviouspage%

```

```

7377         \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse
{\@msdata@add@msd@c @stopabslinenumber}+\@ne}%
7378         {%
7379         \numdef{\@tmp}{\add@msd@c+\@ne}%
7380         \ifcsdef{\@msdata@\@tmp @abslinenumber}%
7381         {\ifnumequal{\csuse{\@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}}%
7382         {}}%
7383         {\@msd@options@fullpagetrue}%
7384         }%
7385         {\@msd@options@fullpagetrue}%
7386         }%
7387         {}}%
7388         \fi%
7389         \fi%
7390         \listxadd{\@msdata@list}{%
7391         \@msd@options@iffullpage%
7392         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\msdata@label}{\expandonce\data}}%
7393         \reset@msd@options@iffullpage%
7394         }%
7395         }%
7396         {}}%
7397         \else%
7398         \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
7399         \ifcsdef{\@msdata@\add@msd@cR @dataR}{%
7400         \letcs{\@data}{\@msdata@\add@msd@cR @dataR}%
7401         \edef\l@d@nums{%
7402         000| Start page = we don't print it
7403         \csuse{\@msdata@\add@msd@cR @linenumberR}| Start line number
7404         000| Start subline number, for now, not used
7405         000| End page number, we don't print it
7406         \ifnumless{\csuse{\@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}%
7407         {\csuse{\@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
7408         {\csuse{@lastline@forpageR@the\page@numR}}%Otherwise, last
number of the page
7409         |%
7410         000| End sub line number, for now, not used
7411         \edfont@info%Font
7412         }%
7413         \@msd@options@fullpagefalse%
7414         \if@firstlineofpageR%
7415         \unless\if@msdata@insertedfrompreviouspage%
7416         \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
csuse{\@msdata@\add@msd@c @stopabslinenumberR}+\@ne}%
7417         {%
7418         \numdef{\@tmp}{\add@msd@cR+\@ne}%
7419         \ifcsdef{\@msdata@\@tmp @abslinenumberR}%

```

```

7420         {\ifunequal{\csuse{@msdata@\tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}}%
7421         {}%
7422         {\@msd@options@fullpagetrue}%
7423     }%
7424     {\@msd@options@fullpagetrue}%
7425 }%
7426 {}%
7427 \fi%
7428 \fi%
7429 \listxadd{\@msdata@list}{%
7430     \@msd@options@iffullpage%
7431     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
7432     \reset@msd@options@iffullpage%
7433 }%
7434 }%
7435 {}%
7436 \fi%
7437 \egroup%
7438 }%
7439 %

```

[@insertedfrompreviouspage](#) The `\if@msdata@insertedfrompreviouspage` boolean is set to TRUE if `reledmac` automatically inserts data from previous page in the first line of a page.

```

7440 \newif\if@msdata@insertedfrompreviouspage%
7441 %

```

[@msdata@firstlineofpage](#) `\add@msdata@firstlineofpage` is called at the first line of every page. It inserts manuscript data which start on one of the previous pages and continue on this page.

```

7442 \newcommand{\add@msdata@firstlineofpage}{%
7443     \bgroup%
7444     \normalfont%
7445     \unless\ifledRcol{%
7446         \ifcsdef{@msdata@\add@msd@c @data}{%
7447             \ifnumless{\the\absline@num-\@one}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}}%
7448         {%
7449             \global\@msdata@insertedfrompreviouspagetrue%
7450             \letcs{\@data}{@msdata@\add@msd@c @data}%
7451             \edef\l@d@nums{%
7452                 000| % Start page = we don't print it
7453                 \numexpr\the\line@num+\@one\relax| % Start line number = first line
of the page. As \add@msdata@firstlineofpage is called before line number
has been incremented, we increment it for printing
7454                 000| % Start subline number, for now, not used
7455                 000| % End page number, we don't print it

```

```

7456 \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
7457 {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
7458 {\csuse{@lastline@forpage@\the\page@num}}}%Otherwise, last
number of the page
7459 |%
7460 000|% End sub line number, for now, not used
7461 \edfont@info%Font
7462 }%
7463 \msd@options@fullpagefalse%
7464 \ifnumless{\csuse{@lastabsline@forpage@\the\page@num}}{\csuse{
@msdata@\add@msd@c @stopabslinenumber}+\@ne}%We will test if the ms data is
for the full page
7465 {%
7466 \numdef{\@tmp}{\add@msd@c+\@ne}%
7467 \ifcsdef{@msdata@\@tmp @abslinenumber}%
7468 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
7469 }%
7470 {\msd@options@fullpagetrue}%
7471 }%
7472 {\msd@options@fullpagetrue}%
7473 }%
7474 {}%
7475 \listxadd{\@msdata@list}{%
7476 \msd@options@iffullpage%
7477 \noexpand\csuse{v\@msdata@series footnote}}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
7478 \reset@msd@options@iffullpage%
7479 }%
7480 }%
7481 {\global\@msdata@insertedfrompreviouspagefalse}%
7482 }{}%
7483 \else%
7484 \ifcsdef{@msdata@\add@msd@cR @dataR}{%
7485 \ifnumless{\the\absline@numR-\@ne}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}}%
7486 {%
7487 \global\@msdata@insertedfrompreviouspagetrue%
7488 \letcs{\@data}{\@msdata@\add@msd@cR @dataR}%
7489 \edef\l@d@nums{%
7490 000|% Start page = we don't print it
7491 \numexpr\the\line@numR+\@ne\relax|% Start line number = first
line of the page. As \add@msdata@firstlineofpage is called before line
number has been incremented, we increment it for printing
7492 000|% Start subline number, for now, not used
7493 000|% End page number, we don't print it
7494 \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@\the\page@numR}}}%

```



```

7495         {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
7496         {\csuse{@lastline@forpageR@the\page@numR}}}%Otherwise, last
number of the page
7497         |%
7498         000|}% End sub line number, for now, not used
7499         \edfont@info%Font
7500         }%
7501         \@msd@options@fullpagefalse%
7502         \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@ne}%
7503         {%
7504         \numdef{\@tmp}{\add@msd@cR+\@ne}%
7505         \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7506         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
7507         {}}%
7508         {\@msd@options@fullpagetrue}%
7509         }%
7510         {\@msd@options@fullpagetrue}%
7511         }%
7512         {}%
7513         \listxadd{\@msdata@list}{%
7514         \@msd@options@iffullpage%
7515         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}}%
7516         \reset@msd@options@iffullpage%
7517         }%
7518         }%
7519         {\global\@msdata@insertedfrompreviouspagefalse}%
7520         }{}%
7521         \fi%
7522         \egroup%
7523     }%
7524     %

```

XXIX.6 Inserting footnote

Just before inserting standard insert (familiar and critical footnotes, sidenotes), we call `\insert@msdata` to insert manuscripts data's footnotes.

```

\insert@msdata 25 \newcommand{\insert@msdata}{%
7526     \def\do##1{##1}%
7527     \dolistloop{\@msdata@list}%
7528     \global\let\@msdata@list\relax%
7529 }%
7530 %

```

XXIX.7 Other

`\@msd@options@iffullpage` `\@msd@options@iffullpage` sets some options if the manuscripts data are for all the page. `\reset@msd@options@iffullpage` resets them after the footnote. `\if@msd@options@fullpage` is switch to true in `add@msdata@firstlineofpage` if these option must be inserted.

```

7531 \newif\if@msd@options@fullpage%
7532 \newcommand{\@msd@options@iffullpage}[0]{%
7533   \if@msd@options@fullpage%
7534     \noexpand\toggletrue{nonum@}%
7535     \ifdefvoid{\ms@data@label}%
7536       {\noexpand\toggletrue{nosep@}}%
7537     }%
7538   \fi%
7539 }%
7540 \newcommand{\reset@msd@options@iffullpage}[0]{%
7541   \noexpand\togglefalse{nonum@}%
7542   \noexpand\togglefalse{nosep@}%
7543 }%
7544 %

```

XXX Arrays and tables

XXX.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread ‘`eeq` and `amstex`’, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsngen` package.

```

7545 \newtoks\@emptytoks
7546
7547 %

```

The rest is from `amsmath`.

`\l@denbody` A token register to contain the body.

```

7548 \newtoks\l@denbody
7549
7550 %

```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```

7551 \newcommand{\addtol@denbody}[1]{%
7552   \global\l@denbody\expandafter{\the\l@denbody#1}}
7553
7554 %

```

`\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`.

```

7555 \newcommand{\l@dcollect@body}[1]{%
7556   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
7557   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}%
7558   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
7559   \begingroup
7560     \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
7561     \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
7562     \processl@denbody%
7563   }%
7564
7565 %

```

`\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.

```

7566 \def\l@dpush@begins#1\begin#2{%
7567   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
7568
7569 %

```

`\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.

```

7570 \def\l@dcollect@@body#1\end#2{%
7571   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
7572     \expandafter\@gobble\l@dbegin@stack}%
7573   \ifx\@empty\l@dbegin@stack
7574     \endgroup
7575     \@checkend{#2}%
7576     \addtol@denbody{#1}%
7577   \else
7578     \addtol@denbody{#1\end{#2}}%
7579   \fi
7580   \processl@denbody % A little tricky! Note the grouping
7581 }
7582
7583 %

```

There was a question on CTT about how to use `\collect@body` for a macro taking an argument. The following is part of that thread.

```
From: Heiko Oberdiek <oberdiek@uni-freiburg.de>
Newsgroups: comp.text.tex
Subject: Re: Using \collect@body with commands that take >1 argument
Date: Fri, 08 Aug 2003 09:03:20 +0200
```

```
eed132@psu.edu (Evan) wrote:
> I'm trying to make a new Latex environment that acts like the>
> \colorbox command that is part of the color package. I looked through
> the FAQ and ran across this bit about using the \collect@body command
> that is part of AMSLaTeX:
> http://www.tex.ac.uk/cgi-bin/texfaq2html?label=cmdasenv
>
> It almost works. If I do something like the following:
> \newcommand{\redbox}[1]{\colorbox{red}{#1}}
>
> \makeatletter
> \newenvironment{redbox}{\collect@body \redbox}{}
```

You will get an error message: Command `\redbox` already defined.
Thus you must rename either the command `\redbox` or the environment name.

```
> \begin{coloredbox}{blue}
> Yadda yadda yadda... this is on a blue background...
> \end{coloredbox}
> and can't figure out how to make the \collect@body take this.

> \collect@body \colorbox{red}
> \collect@body {\colorbox{red}}
```

The argument of `\collect@body` has to be one token exactly.

```
\documentclass{article}
\usepackage{color}
\usepackage{amsmath}

\newcommand{\redbox}[1]{\colorbox{red}{#1}}
\makeatletter
\newenvironment{coloredbox}[1]{%
  \def\next@{\colorbox{#1}}%
  \collect@body\next@
}{%

% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}[1]{%
  \def\next@{\mycoloredbox{#1}}%
  \collect@body\next@
```

```

}{}
\newcommand{\mycoloredbox}[2]{%
  \colorbox{#1}{\ignorespaces#2\unskip}%
}

% support of optional color model argument
\newcommand\coloredboxIII\endcsname{}
\def\coloredboxIII#1#{%
  \@coloredboxIII{#1}%
}
\def\@coloredboxIII#1#2{%
  \def\next@{\mycoloredboxIII{#1}{#2}}%
  \collect@body\next@
}
\newcommand{\mycoloredboxIII}[3]{%
  \colorbox{#1}{#2}{\ignorespaces#3\unskip}%
}

\makeatother

\begin{document}
  Black text before
  \begin{coloredbox}{blue}
    Hello World
  \end{coloredbox}
  Black text after

  Black text before
  \begin{coloredboxII}{blue}
    Hello World
  \end{coloredboxII}
  Black text after

  Black text before
  \begin{coloredboxIII}[rgb]{0,0,1}
    Hello World
  \end{coloredboxIII}
  Black text after

\end{document}

Yours sincerely
Heiko <oberdiek@uni-freiburg.de>

```

XXX.2 Tabular environments

This is based on the work by Herbert Breger in developing `tabmac.tex`.

The original `tabmac.tex` file was void of comments or any explanatory text other

than the above notice. The algorithm is Breger's. Peter Wilson have made some cosmetic changes to the original code and reimplemented some things so they are more LaTeX-like. All the commentary are from Peter Wilson, as are any mistake or errors.

However, Maïeul Rouquette has modified code in order to add new features of `eledmac` and `reledmac`.

XXX.2.1 Disabling and restoring commands

`\l@dtabnoexpands` More no expansion for critical and familiar footnotes in tabular environment.

```

7584 \newcommand*\l@dtabnoexpands}{%
7585   \let\rtab=0%
7586   \let\ctab=0%
7587   \let\ltab=0%
7588   \let\rtabtext=0%
7589   \let\ltabtext=0%
7590   \let\ctabtext=0%
7591   \let\edbeforetab=0%
7592   \let\edaftertab=0%
7593   \let\edatleft=0%
7594   \let\edatright=0%
7595   \let\edvertline=0%
7596   \let\edvertdots=0%
7597   \let\edrowfill=0%
7598 }
7599
7600 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in `edtabularx` and `edarrayx` environments.

`\restore@familiarnotes`

```

7601 \newcommand{\disable@familiarnotes}{%
7602   \unless\ifnofamiliar%
7603     \def\do##1{%
7604       \csletcs{footnote@##1}{footnote##1}%
7605       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
7606         \protected@csxdef{thefnmark##1}{\csuse{thefootnote##1}}%
7607         \csuse{@footnotemark##1}%
7608       }%
7609     }%
7610     \dolistloop{\@series}%
7611   \fi%
7612 }%
7613 \newcommand{\restore@familiarnotes}{%
7614   \unless\ifnofamiliar%
7615     \def\do##1{%
7616       \csletcs{footnote##1}{footnote@##1}%
7617     }%
7618     \dolistloop{\@series}%
7619   \fi%

```

```

7620 }%
7621
7622 %

```

`\disable@sidenotes` The same, for side notes.

`\restore@sidenotes`

```

7623 \newcommand{\disable@sidenotes}{%
7624   \let\@@ledrightnote\ledrightnote%
7625   \let\@@ledleftnote\ledleftnote%
7626   \let\@@ledsidenote\ledsidenote%
7627   \let\ledrightnote@gobble%
7628   \let\ledleftnote@gobble%
7629   \let\ledsidenote@gobble%
7630 }%
7631 \newcommand{\restore@sidenotes}{%
7632   \let\ledrightnote\@@ledrightnote%
7633   \let\ledleftnote\@@ledleftnote%
7634   \let\ledsidenote\@@ledsidenote%
7635 }%
7636 %

```

`\disable@notes` Disable/restore side and familiar notes.

`\restore@notes`

```

7637 \newcommand{\disable@notes}{%
7638   \disable@sidenotes%
7639   \disable@familiarnotes%
7640 }%
7641 \newcommand{\restore@notes}{%
7642   \restore@sidenotes%
7643   \restore@familiarnotes%
7644 }%
7645 %

```

`\EDTEXT` We need to be able to modify the `\edtext` macros and also restore their original definitions.

`\xedtext`

```

7646 \let\EDTEXT=\edtext
7647 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
7648 %

```

`\EDLABEL` We need to be able to modify and restore the `\edlabel` macro.

`\xedlabel`

```

7649 \let\EDLABEL=\edlabel
7650 \newcommand*\xedlabel[1]{\EDLABEL{#1}}
7651 %

```

`\xedindex` Macros supporting modification and restoration of `\edindex`.

`\nulledindex`

```

7652 \AtBeginDocument{\let\xedndex\edíndex}%
7653 \newcommand{\nulledíndex}[2][\jobname]{\@bsphack\@esphack}
7654
7655 %

```

\@line@num Macro supporting restoration of \linenum.

```

7656 \let\@line@num=\linenum
7657 %

```

\l@dgobblearg \l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩} replaces these two arguments (first is optional) by \relax.

```

7658 \newcommand*\l@dgobbleoptarg[2][\relax]%
7659
7660 %

```

\Relax₆₁ \let\Relax=\relax

\NEXT₆₂ \let\NEXT=\next

```

7663
7664 %

```

\l@dmodforedtext Modify and restore various macros for when \edtext is used.

\l@drestoreforedtext

```

7665 \newcommand{\l@dmodforedtext}{%
7666   \let\edtext\relax
7667   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
7668   \dolistloop{\@series}%
7669   \let\edíndex\nulledíndex
7670   \let\linenum@gobble}
7671 \newcommand{\l@drestoreforedtext}{%
7672   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
7673   \dolistloop{\@series}%
7674   \let\edíndex\xedndex}
7675 %

```

\l@dnullfills Nullify and restore some column fillers, etc.

\l@drestorefills

```

7676 \newcommand{\l@dnullfills}{%
7677   \def\edlabel##1{%
7678     \def\edrowfill##1##2##3{%
7679     }
7680   \newcommand{\l@drestorefills}{%
7681     \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
7682   }
7683
7684 %

```


`\letsforverteilen` Gathers some lets and other code that is common to the `*verteilen*` macros.

```

7685 \newcommand{\letsforverteilen}{%
7686   \let\edtext\xedtext
7687   \let\edindex\xedindex
7688   \def\do##1{\global\csletcs{##1footnote}{##1@@footnote}}
7689   \dolistloop{@series}%
7690   \let\linenum@line@num
7691   \hilfsskip=\l@dcwidth%
7692   \advance\hilfsskip by -\wd\hilfsbox
7693   \def\edlabel##1{\xedlabel{##1}}
7694
7695 %

```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for
`\enablel@dtabfeet` `\edtext`.

```

7696 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
7697 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
7698 %

```

XXX.2.2 Counters, boxes and lengths

`\l@dampcount` `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter
`\l@dcolcount` for the columns.

```

7699 \newcount\l@dampcount
7700 \l@dampcount=1\relax
7701 \newcount\l@dcolcount
7702 \l@dcolcount=0\relax
7703
7704 %

```

`\hilfsbox` Some (temporary) helper items.

```

7705 \hilfsskip
7706 \Hilfsbox
7707 \hilfscount
7708
7709 %
7710

```

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).

```

7711 \newdimen\dcoli
7712 \newdimen\dcolii
7713 \newdimen\dcoliii
7714 \newdimen\dcoliv
7715 \newdimen\dcolv

```

```

7716 \newdimen\dc colvi
7717 \newdimen\dc colvii
7718 \newdimen\dc colviii
7719 \newdimen\dc colix
7720 \newdimen\dc colx
7721 \newdimen\dc colxi
7722 \newdimen\dc colxii
7723 \newdimen\dc colxiii
7724 \newdimen\dc colxiv
7725 \newdimen\dc colxv
7726 \newdimen\dc colxvi
7727 \newdimen\dc colxvii
7728 \newdimen\dc colxviii
7729 \newdimen\dc colxix
7730 \newdimen\dc colxx
7731 \newdimen\dc colxxi
7732 \newdimen\dc colxxii
7733 \newdimen\dc colxxiii
7734 \newdimen\dc colxxiv
7735 \newdimen\dc colxxv
7736 \newdimen\dc colxxvi
7737 \newdimen\dc colxxvii
7738 \newdimen\dc colxxviii
7739 \newdimen\dc colxxix
7740 \newdimen\dc colxxx
7741 \newdimen\dc colerr % added for error handling
7742
7743 %

```

\l@dc colwidth This is a cunning way of storing the columnwidths indexed by the column number \l@dc colcount, like an array. (was \Dimenzuordnung)

```

7744 \newcommand{\l@dc colwidth}{\ifcase \the\l@dc colcount \dc oli %???
7745 \or \dc oli \or \dc olii \or \dc oliii
7746 \or \dc oliv \or \dc olv \or \dc olvi
7747 \or \dc olvii \or \dc olviii \or \dc olvix \or \dc olx
7748 \or \dc olxi \or \dc olxii \or \dc olxiii
7749 \or \dc olxiv \or \dc olxv \or \dc olxvi
7750 \or \dc olxvii \or \dc olxviii \or \dc olxix \or \dc olxx
7751 \or \dc olxxi \or \dc olxxii \or \dc olxxiii
7752 \or \dc olxxiv \or \dc olxxv \or \dc olxxvi
7753 \or \dc olxxvii \or \dc olxxviii \or \dc olxxix \or \dc olxxx
7754 \else \dc olerr \fi}
7755
7756 %

```

\step1@dc colcount This increments the column counter, and issues an error message if it is too large.

```

7757 \newcommand*{\step1@dc colcount}{\advance\l@dc colcount\@ne
7758 \ifnum\l@dc colcount>30\relax

```

```

7759 \led@err@TooManyColumns
7760 \fi}
7761
7762 %

```

\l@setmaxcolwidth Sets the column width to the maximum value seen so far.

```

7763 \newcommand{\l@setmaxcolwidth}{%
7764 \ifdim\l@dcwidth < \wd\hilfsbox
7765 \l@dcwidth = \wd\hilfsbox
7766 \else \relax \fi}
7767
7768 %

```

\measurecell Measure (recursively) the width required for a math cell.

```

7769 \def\measurecell #1{%
7770 \ifx #1\ \ifnum\l@dcwidth=0\let\NEXT\relax%
7771 \else\l@dcwidth=0%
7772 \l@dcwidth=0%
7773 \let\NEXT\measurecell%
7774 \fi%
7775 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7776 \step\l@dcwidth%
7777 \l@setmaxcolwidth%
7778 \let\NEXT\measurecell%
7779 \fi\NEXT}
7780
7781 %

```

\measuretextcell Measure (recursively) the width required for a text cell.

```

7782 \def\measuretextcell #1{%
7783 \ifx #1\ \ifnum\l@dcwidth=0\let\NEXT\relax%
7784 \else\l@dcwidth=0%
7785 \l@dcwidth=0%
7786 \let\NEXT\measuretextcell%
7787 \fi%
7788 \else\setbox\hilfsbox=\hbox{#1}%
7789 \step\l@dcwidth%
7790 \l@setmaxcolwidth%
7791 \let\NEXT\measuretextcell%
7792 \fi\NEXT}
7793
7794 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

7795 \def\measuremrow #1{%
7796 \ifx #1\ \let\NEXT\relax%

```

```

7797 \else\measuremcell #1&\\&\\&%
7798 \let\NEXT\measuremrow%
7799 \fi\NEXT}
7800 %

```

\measuretrow Measure (recursively) the width required for a text row.

```

7801 \def\measuretrow #1\\{%
7802 \ifx #1&\let\NEXT\relax%
7803 \else\measuretcell #1&\\&\\&%
7804 \let\NEXT\measuretrow%
7805 \fi\NEXT}
7806
7807 %

```

\edtabcolsep The length \edtabcolsep controls the distance between columns.

```

7808 \newskip\edtabcolsep
7809 \global\edtabcolsep=10pt
7810
7811 %

```

\variab₁₂ \newcommand{\variab}{\relax}

```

7813
7814 %

```

\l@dcheckcols Check that the number of columns is consistent.

```

7815 \newcommand*{\l@dcheckcols}{%
7816 \ifnum\l@dcolcount=1\relax
7817 \else
7818 \ifnum\l@dampcount=1\relax
7819 \else
7820 \ifnum\l@dcolcount=\l@dampcount\relax
7821 \else
7822 \l@d@err@UnequalColumns
7823 \fi
7824 \fi
7825 \l@dampcount=\l@dcolcount
7826 \fi}
7827
7828 %

```

\edfilldimen A length.

```

7829 \newdimen\edfilldimen
7830 \edfilldimen=0pt
7831
7832 %

```

`\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`

```
7833 \newcounter{addcolcount}
7834 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
7835 %
```

XXX.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.

```
7836 \def\setmcellright #1{\def\edlabel##1{}%
7837 \let\edindex\nulledindex
7838 \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
7839 \let\Next\relax%
7840 \else\l@dcolcount=0%
7841 \let\Next=\setmcellright%
7842 \fi%
7843 \else%
7844 \disablel@dtabfeet%
7845 \step1@dcolcount%
7846 \disable@notes%
7847 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7848 \restore@notes%
7849 \letsforverteilen%
7850 \hskip\hilfsskip$\displaystyle{#1}$%
7851 \hskip\edtabcolsep%
7852 \let\Next=\setmcellright%
7853 \fi\Next}
7854 %
7855 %
```

`\settcellright` Typeset (recursively) cells of text right justified.

```
7856 \def\settcellright #1{\def\edlabel##1{}%
7857 \let\edindex\nulledindex
7858 \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
7859 \let\Next\relax%
7860 \else\l@dcolcount=0%
7861 \let\Next=\settcellright%
7862 \fi%
7863 \else%
7864 \disablel@dtabfeet%
7865 \step1@dcolcount%
7866 \disable@notes%
7867 \setbox\hilfsbox=\hbox{#1}%
7868 \restore@notes%
7869 \letsforverteilen%
7870 \hskip\hilfsskip#1%
```

```

7871         \hskip\edtabcolsep%
7872         \let\Next=\settccllright%
7873     \fi\Next}
7874 %

```

\setmcellleft Typeset (recursively) cells of display math left justified.

```

7875 \def\setmcellleft #1&{\def\edlabel##1{}%
7876     \let\edindex\nulledindex
7877     \ifx #1\\ \ifnum\l@dc@colcount=0 \let\Next\relax%
7878         \else\l@dc@colcount=0%
7879         \let\Next=\setmcellleft%
7880         \fi%
7881     \else \disablel@dtabfeet%
7882         \stepl@dc@colcount%
7883         \disable@notes%
7884         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7885         \restore@notes%
7886         \letsforverteilen%
7887         $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
7888         \let\Next=\setmcellleft%
7889     \fi\Next}
7890
7891 %

```

\settcclleft Typeset (recursively) cells of text left justified.

```

7892 \def\settcclleft #1&{\def\edlabel##1{}%
7893     \let\edindex\nulledindex
7894     \ifx #1\\ \ifnum\l@dc@colcount=0 \let\Next\relax%
7895         \else\l@dc@colcount=0%
7896         \let\Next=\settcclleft%
7897         \fi%
7898     \else \disablel@dtabfeet%
7899         \stepl@dc@colcount%
7900         \disable@notes%
7901         \setbox\hilfsbox=\hbox{#1}%
7902         \restore@notes%
7903         \letsforverteilen%
7904         #1\hskip\hilfsskip\hskip\edtabcolsep%
7905         \let\Next=\settcclleft%
7906     \fi\Next}
7907 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

7908 \def\setmcellcenter #1&{\def\edlabel##1{}%
7909     \let\edindex\nulledindex
7910     \ifx #1\\ \ifnum\l@dc@colcount=0\let\Next\relax%
7911         \else\l@dc@colcount=0%

```

```

7912         \let\Next=\setmcellcenter%
7913     \fi%
7914 \else    \disablel@dtabfeet%
7915         \stepl@dcolcount%
7916         \disable@notes%
7917         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7918         \restore@notes%
7919         \letsforverteilen%
7920         \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
7921         \hskip\edtabcolsep%
7922         \let\Next=\setmcellcenter%
7923     \fi\Next}
7924
7925 %

```

\settcellcenter Typeset (recursively) cells of text centered.

```

7926 \def\settcellcenter #1&{\def\edlabel##1{}%
7927     \let\edindex\nulledindex
7928     \ifx #1\\ \ifnum\l@dcolcount=0 \let\Next\relax%
7929         \else\l@dcolcount=0%
7930             \let\Next=\settcellcenter%
7931         \fi%
7932     \else    \disablel@dtabfeet%
7933             \stepl@dcolcount%
7934             \disable@notes%
7935             \setbox\hilfsbox=\hbox{#1}%
7936             \restore@notes%
7937             \letsforverteilen%
7938             \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
7939             \hskip\edtabcolsep%
7940             \let\Next=\settcellcenter%
7941     \fi\Next}
7942
7943 %

```

\NEXT `\let\NEXT=\relax`

```

7945
7946 %

```

\setmrowright Typeset (recursively) rows of right justified math.

```

7947 \def\setmrowright #1\\{%
7948     \ifx #1& \let\NEXT\relax
7949     \else \centerline{\setmcellright #1&\\&\\&}
7950         \let\NEXT=\setmrowright
7951     \fi\NEXT}
7952 %

```

\settroright Typeset (recursively) rows of right justified text.

```

7953 \def\settroright #1\{\%
7954     \ifx #1& \let\NEXT\relax
7955     \else \centerline{\settcclright #1&\&\&\&}
7956         \let\NEXT=\settroright
7957     \fi\NEXT}
7958
7959 %

```

\setmrowleft Typeset (recursively) rows of left justified math.

```

7960 \def\setmrowleft #1\{\%
7961     \ifx #1& \let\NEXT\relax
7962     \else \centerline{\setmcclleft #1&\&\&\&}
7963         \let\NEXT=\setmrowleft
7964     \fi\NEXT}
7965 %

```

\settrorleft Typeset (recursively) rows of left justified text.

```

7966 \def\settrorleft #1\{\%
7967     \ifx #1& \let\NEXT\relax
7968     \else \centerline{\settcclleft #1&\&\&\&}
7969         \let\NEXT=\settrorleft
7970     \fi\NEXT}
7971
7972 %

```

\setmrowcenter Typeset (recursively) rows of centered math.

```

7973 \def\setmrowcenter #1\{\%
7974     \ifx #1& \let\NEXT\relax%
7975     \else \centerline{\setmcclcenter #1&\&\&\&}
7976         \let\NEXT=\setmrowcenter
7977     \fi\NEXT}
7978 %

```

\settrorcenter Typeset (recursively) rows of centered text.

```

7979 \def\settrorcenter #1\{\%
7980     \ifx #1& \let\NEXT\relax
7981     \else \centerline{\settcclcenter #1&\&\&\&}
7982         \let\NEXT=\settrorcenter
7983     \fi\NEXT}
7984
7985 %

```


`\nullsetzen` `\newcommand{\nullsetzen}{%`

```

7987   \step1@dcolcount%
7988   \l@dcolwidth=0pt%
7989   \ifnum\l@dcolcount=30\let\NEXT\relax%
7990       \l@dcolcount=0\relax
7991   \else\let\NEXT\nullsetzen%
7992   \fi\NEXT}
7993
7994 %

```

`\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.`

```

7995 \newcommand{\edatleft}[3][\@empty]{%
7996   \ifx#1\@empty
7997     \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
7998         depth 0pt \right. $\hss}\vfil}
7999   \else
8000     \vbox to 4pt{\vss\hbox{#1\left#2\vrule width0pt height #3
8001         depth 0pt \right. $\vfil}
8002   \fi}
8003 %

```

`\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.`

```

8004 \newcommand{\edatright}[3][\@empty]{%
8005   \ifx#1\@empty
8006     \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
8007         depth 0pt \right#2 $\hss}\vfil}
8008   \else
8009     \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
8010         depth 0pt \right#2 #1 $\vfil}
8011   \fi}
8012
8013 %

```

`\edvertline` `\edvertline{ $\langle len \rangle$ }` vertical line $\langle len \rangle$ high.

```

8014 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
8015
8016 %

```

`\edvertdots` `\edvertdots{ $\langle len \rangle$ }` vertical dotted line $\langle len \rangle$ high.

```

8017 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
8018     {\cleaders\hbox{$\m@th\hbox{.}\vbox to 0.5em{ }\vfil}}}
8019
8020 %

```

\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@`.

```

8021 \newcommand{\l@dtabaddcols}[2]{%
8022   \l@dcheckstartend{#1}{#2}%
8023   \ifl@dstartendok
8024     \setcounter{addcolcount}{#1}%
8025     \@whilenum \value{addcolcount}<#2\relax \do
8026     {\advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
8027      \advance\edfilldimen by \edtabcolsep
8028      \stepcounter{addcolcount}}%
8029     \advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
8030     \fi
8031   }
8032
8033 %

```

\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.

```

8034 \newif\ifl@dstartendok
8035 \newcommand{\l@dcheckstartend}[2]{%
8036   \l@dstartendoktrue
8037   \ifnum #1<\@ne
8038     \l@dstartendokfalse
8039     \led@err@LowStartColumn
8040   \fi
8041   \ifnum #2>30\relax
8042     \l@dstartendokfalse
8043     \led@err@HighEndColumn
8044   \fi
8045   \ifnum #1>#2\relax
8046     \l@dstartendokfalse
8047     \led@err@ReverseColumns
8048   \fi
8049 }
8050
8051 %

```

\edrowfill `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a \TeX style reimplementation and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktel` macros.

```

8052 \newcommand*{\edrowfill}[3]{%
8053   \l@dtabaddcols{#1}{#2}%
8054   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
8055 \let\@edrowfill@=\edrowfill
8056 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}

```

```
8057
8058 %
```

\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`.

```
8059 \newcommand{\leftltab}[1]{%
8060   \hb@xt@ \z@{\vbox{\edtabindent%
8061     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8062
8063 %
```

\leftrtab `\leftrtab{⟨text⟩}{⟨math⟩}` for `\edbeforetab` in `\rtab`.

```
8064 \newcommand{\leftrtab}[2]{%
8065   #2\hb@xt@ \z@{\vbox{\edtabindent%
8066     \advance\Hilfsskip by\dcoli%
8067     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8068
8069 %
```

\leftctab `\leftctab{⟨text⟩}{⟨math⟩}` for `\edbeforetab` in `\ctab`.

```
8070 \newcommand{\leftctab}[2]{%
8071   \hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8072     \advance\Hilfsskip by 0.5\dcoli%
8073     \setbox\hilfsbox=\hbox{\def\edlabel##1}%
8074     \disablel@dtabfeet$\displaystyle{#2}$}%
8075     \advance\Hilfsskip by -0.5\wd\hilfsbox%
8076     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
8077   #2}
8078
8079 %
```

\rightctab `\rightctab{⟨math⟩}{⟨text⟩}` for `\edaftertab` in `\ctab`.

```
8080 \newcommand{\rightctab}[2]{%
8081   \setbox\hilfsbox=\hbox{\def\edlabel##1}%
8082   \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8083   #1\hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8084     \advance\Hilfsskip by 0.5\l@dcolwidth%
8085     \advance\Hilfsskip by -\wd\hilfsbox%
8086     \setbox\hilfsbox=\hbox{\def\edlabel##1}%
8087     \disablel@dtabfeet$\displaystyle{#1}$}%
8088     \advance\Hilfsskip by -0.5\wd\hilfsbox%
```

```

8089 \advance\Hilfsskip by \edtabcolsep%
8090 \moveright\Hilfsskip\hbox{ #2}}\hss}%
8091 }
8092
8093 %

```

\rightltab `\rightltab{<math>}<{<text>}` for `\edaftertab` in `\ltab`.

```

8094 \newcommand{\rightltab}[2]{%
8095     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8096         \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8097         #1\hb@xt@{\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8098         \advance\Hilfsskip by\l@dcolwidth%
8099         \advance\Hilfsskip by-\wd\hilfsbox%
8100         \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8101             \disablel@dtabfeet$\displaystyle{#1}$}%
8102             \advance\Hilfsskip by-\wd\hilfsbox%
8103             \advance\Hilfsskip by\edtabcolsep%
8104             \moveright\Hilfsskip\hbox{ #2}}\hss}%
8105         }
8106     }
8107 %

```

\rightrtab `\rightrtab{<math>}<{<text>}` for `\edaftertab` in `\rtab`.

```

8108 \newcommand{\rightrtab}[2]{%
8109     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8110         \disablel@dtabfeet#2}%
8111         #1\hb@xt@{\z@{\vbox{\edtabindent%
8112         \advance\Hilfsskip by-\wd\hilfsbox%
8113         \advance\Hilfsskip by\edtabcolsep%
8114         \moveright\Hilfsskip\hbox{ #2}}\hss}%
8115     }
8116
8117 %

```

\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.

```

8118 \newcommand{\rtab}[1]{%
8119     \l@dnnullfills
8120     \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
8121     \def\edaftertab##1##2{\rightltab{##1}{##2}}%
8122     \measurebody{#1}%
8123     \l@drestorefills
8124     \variab
8125     \setmrowright #1\\&\\%
8126     \enablel@dtabfeet}
8127
8128 %

```

`\measurebody` `\measurebody{<body>}` measures the array `<body>`.

```
8129 \newcommand{\measurebody}[1]{%
8130   \disablel@dtabfeet%
8131   \l@dc@count=0%
8132   \nullsetzen%
8133   \l@dc@count=0
8134   \measuremrow #1\\&\\%
8135   \global\l@dampcount=1}
8136
8137 %
```

`\rtabtext` `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```
8138 \newcommand{\rtabtext}[1]{%
8139   \l@dnullfills
8140   \measuretbody{#1}%
8141   \l@drestorefills
8142   \variab
8143   \settr@right #1\\&\\%
8144   \enablel@dtabfeet}
8145
8146 %
```

`\measuretbody` `\measuretbody{<body>}` measures the tabular `<body>`.

```
8147 \newcommand{\measuretbody}[1]{%
8148   \disable@notes%
8149   \disablel@dtabfeet%
8150   \l@dc@count=0%
8151   \nullsetzen%
8152   \l@dc@count=0
8153   \measuretbody #1\\&\\%
8154   \restore@notes%
8155   \global\l@dampcount=1}
8156
8157 %
```

`\ltab` Array with entries left justified.

```
\edbeforetab
\edaftertab
8158 \newcommand{\ltab}[1]{%
8159   \l@dnullfills
8160   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
8161   \def\edaftertab##1##2{\rightltab{##1}{##2}}%
8162   \measuretbody{#1}%
8163   \l@drestorefills
8164   \variab
8165   \setmrowleft #1\\&\\%
8166   \enablel@dtabfeet}
8167
8168 %
```

\ltabtext Tabular with entries left justified.

```

8169 \newcommand{\ltabtext}[1]{%
8170   \l@dnnullfills
8171   \measuretbody{#1}%
8172   \l@drestorefills
8173   \variab
8174   \settrorleft #1\&\%
8175   \enablel@dtabfeet}
8176
8177 %

```

\ctab Array with centered entries.

```

\edbeforetab
\edaftertab
8178 \newcommand{\ctab}[1]{%
8179   \l@dnnullfills
8180   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
8181   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
8182   \measuretbody{#1}%
8183   \l@drestorefills
8184   \variab
8185   \setmrowcenter #1\&\%
8186   \enablel@dtabfeet}
8187
8188 %

```

\ctabtext Tabular with entries centered.

```

8189 \newcommand{\ctabtext}[1]{%
8190   \l@dnnullfills
8191   \measuretbody{#1}%
8192   \l@drestorefills
8193   \variab
8194   \settrorcenter #1\&\%
8195   \enablel@dtabfeet}
8196
8197 %

```

```

\spreadtext
8198 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
8199   \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
8200 %

```

```

\spreadmath
8201 \newcommand{\spreadmath}[1]{%
8202   \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
8203
8204 %

```

\HILFSskip More helpers.

\Hilfsskip

```

8205 \newskip\HILFSskip
8206 \newskip\Hilfsskip
8207
8208 %

```

```

\EDTABINDENT \newcommand{\EDTABINDENT}{%
8209
8210   \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
8211   \else\step1@dcolcount%
8212     \advance\Hilfsskip by\l@dcolwidth%
8213     \ifdim\l@dcolwidth=0pt\advance\hilfscount\@ne
8214     \else\advance\Hilfsskip by \the\hilfscount\edtabcolsep%
8215     \hilfscount=1\fi%
8216     \let\NEXT=\EDTABINDENT%
8217   \fi\NEXT}%
8218 %

```

`\edtabindent` (was `\tabindent`)

```

8219 \newcommand{\edtabindent}{%
8220   \l@dcolcount=0\relax
8221   \Hilfsskip=0pt%
8222   \hilfscount=1\relax
8223   \EDTABINDENT%
8224   \hilfsskip=\hsize%
8225   \advance\hilfsskip -\Hilfsskip%
8226   \Hilfsskip=0.5\hilfsskip%
8227 }%
8228
8229 %

```

`\EDTAB` (was `\TAB`)

```

8230 \def\EDTAB #1|#2|{%
8231   \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
8232   \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
8233   \advance\tabelskip -\wd\tabhilfbox%
8234   \advance\tabelskip -\wd\tabHilfbox%
8235   \unhbox\tabhilfbox\hskip\tabelskip%
8236   \unhbox\tabHilfbox}%
8237
8238 %

```

`\EDTABtext` (was `\TABtext`)

```

8239 \def\EDTABtext #1|#2|{%
8240   \setbox\tabhilfbox=\hbox{#1}%
8241   \setbox\tabHilfbox=\hbox{#2}%
8242   \advance\tabelskip -\wd\tabhilfbox%
8243   \advance\tabelskip -\wd\tabHilfbox%

```

`\tabhilfbox` Further helpers.

```
\newbox\tabhilfbox
\newbox\tabHilfbox

%
```

edarrayl edarrayc edarrayr The ‘environment’ forms for \ltab, \ctab and \rtab.

```
\newenvironment{edarrayl}{\l@ddcollect@body\ltab}{  
\newenvironment{edarrayc}{\l@ddcollect@body\ctab}{  
\newenvironment{edarrayr}{\l@ddcollect@body\rtab}{  
  
%  
%
```

```
\newenvironment{edtabularl}{\l@collect@body\ltabtext}{  
\newenvironment{edtabularc}{\l@collect@body\ctabtext}{  
\newenvironment{edtabularr}{\l@collect@body\rtabtext}{  
  
%
```

`\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.

```

\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%
  }
```



```

8272 \global\rightskip=\leftmargin%
8273 }
8274 \renewcommand{\endquotation}{\par%
8275 \global\leftskip=0pt%
8276 \global\rightskip=0pt%
8277 \leavevmode%
8278 \skipnumbering%
8279 \ifautopar%
8280 \vskip-\parskip%
8281 \else%
8282 \vskip\topsep%
8283 \fi%
8284 }
8285 \renewcommand{\quote}{\par\leavevmode%
8286 \parindent=0pt%
8287 \skipnumbering%
8288 \ifautopar%
8289 \vskip-\parskip%
8290 \else%
8291 \vskip\topsep%
8292 \fi%
8293 \global\leftskip=\leftmargin%
8294 \global\rightskip=\leftmargin%
8295 }
8296 \renewcommand{\endquote}{\par%
8297 \global\leftskip=0pt%
8298 \global\rightskip=0pt%
8299 \leavevmode%
8300 \skipnumbering%
8301 \ifautopar%
8302 \vskip-\parskip%
8303 \else%
8304 \vskip\topsep%
8305 \fi%
8306 }
8307 \fi
8308 }
8309 %

```

XXXII Section's title commands

XXXII.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```

8310 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
8311 %

```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```

8312 \newcommand{\ledsectnomark}{%
8313   \let\chaptermark@gobble%
8314   \let\sectionmark@gobble%
8315   \let\subsectionmark@gobble%
8316 }
8317 %

```

XXXII.2 General overview

The system of `\eledxxxx` commands to section text work like this:

1. When one of these commands is called, `reledmac` writes to an auxiliary files:
 - The section level.
 - The section title.
 - The side (when `eledpar` is used).
 - The `pstart` where the command is called.
 - If we have starred version or not.
2. `reledmac` adds the title of the section to `pstart`, as normal content. This is to enable critical notes.
3. When \TeX is run a other time, this file is read. That:
 - Adds the `pstart` number to a list of `pstarts` where a sectioning command is used.
 - Defines a command, the name of which contains the `pstart` number, and which calls the normal \TeX sectioning command.
4. This last command is called when the `pstart` is effectively printed.

XXXII.3 `\beforeeledchapter` command

We do not define commands for `\eledsection` and related if the `noeledsec` option is loaded. We use `etoolbox` tests and not the `\ifxxx...else...\fi` structure to prevent problem of expansions with command after the `\ifxxx` which contains `\fi`. As we patch command inside this test, we need to change the category code of `#` character *before* `\notbool` statement, because the second argument is read with the standard `catcode` (read *The TeXbook* to understand when the `catcode`'s change has effect).

```

8318 \catcode`\#=12
8319 \notbool{@noeled@sec}{%
8320 %

```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```

8321 \ifl@dmemoir
8322   \newcommand\beforeeledchapter{%
8323     \clearforchapter%
8324   }
8325 \else
8326   \newcommand\beforeeledchapter{%
8327     \if@openright%
8328       \cleardoublepage%
8329     \else%
8330       \clearpage%
8331     \fi%
8332   }
8333 \fi
8334 %

```

XXXII.4 Auxiliary commands

`\print@leftmargin@eledsection` and `\print@rightmargin@eledsection` are added by `reledmac` inside the code of sectioning command, in order to affix line numbers. They include tests for RTL languages.

```

8335 \def\print@rightmargin@eledsection{%
8336   \if@eled@sectioning%
8337     \begingroup%
8338     \if@RTL%
8339       \let\llap\rlap%
8340       \let\leftlinenum\rightlinenum%
8341       \let\leftlinenumR\rightlinenumR%
8342       \let\l@drd@ta\l@dld@ta%
8343       \let\l@drsn@te\l@dlsn@te%
8344     \fi%
8345     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
8346   \endgroup%
8347 \fi%
8348 }%
8349
8350 \def\print@leftmargin@eledsection{%
8351   \if@eled@sectioning%
8352     \leavevmode%
8353     \begingroup%
8354     \if@RTL%
8355       \let\rlap\llap%
8356       \let\rightlinenum\leftlinenum%
8357       \let\rightlinenumR\leftlinenumR%
8358       \let\l@dld@ta\l@drd@ta%
8359       \let\l@dlsn@te\l@drsn@te%
8360     \fi%

```

```

8361 \l@ddld@ta\csuse{LR}{\l@dlsn@te}%
8362 \endgroup%
8363 \fi%
8364 }%
8365
8366 %

```

XXXII.5 Patching standard commands

`\M@sect` We have to patch `LATEX`, `book` and `memoir` sectioning commands in order to:

`\@mem@old@ssect`

`\@makechapterhead`

`\@makechapterhead`

`\@makeschapterhead`

`\@sect`

`\@ssect`

- Disable `\edtext` inside.
- Disable page breaking (for `\chapter`).
- Add line numbers and sidenotes.

Unfortunately, Maïeul Rouquette was not able to try if `memoir` is loaded. That is why `eledmac` tries to define for both standard class and `memoir` class.

```

8367 \AtBeginDocument{%
8368
8369
8370 \pretocmd{\M@sect}
8371 {\let\old@edtext=\edtext%
8372 \let\edtext=\dummy@edtext@showlemma%
8373 }
8374 {}
8375 {}
8376
8377 \apptocmd{\M@sect}
8378 {\let\edtext=\old@edtext}
8379 {}
8380 {}
8381
8382 \patchcmd{\M@sect}
8383 { #9}
8384 { #9%
8385 \print@rightmargin@eledsection%
8386 }
8387 {}
8388 {}
8389
8390 \patchcmd{\M@sect}
8391 {\hskip #3\relax}
8392 {\hskip #3\relax%
8393 \print@leftmargin@eledsection%
8394 }
8395 {}
8396 {}

```

```

8397 \patchcmd{\@mem@old@ssect}
8398   {\#5}
8399   {\#5%
8400   \print@leftmargin@eledsection%
8401   }
8402   {}
8403   {}
8404   {}
8405
8406 \patchcmd{\@mem@old@ssect}
8407   {\hskip #1}
8408   {\hskip #1%
8409   \print@rightmargin@eledsection%
8410   }
8411   {}
8412   {}
8413
8414
8415
8416 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8417   \if@eled@sectioning\else%
8418   \ifl@dprintingpages\else%
8419   \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a
8420   \Pages: will keep critical notes from printing on the title page. Here for
8421   \scrbook.
8422   \fi%
8423   \fi%
8424   }
8425   {}
8426   {}
8427
8428 \patchcmd{\@makechapterhead}
8429   {\#1}
8430   {\print@leftmargin@eledsection%
8431   #1%
8432   \print@rightmargin@eledsection%
8433   }
8434   {}
8435   {}
8436
8437 \patchcmd{\@makechapterhead}% For BIDI
8438   {\if@RTL\raggedleft\else\raggedright\fi}%
8439   {\if@eled@sectioning\else%
8440   \if@RTL\raggedleft\else\raggedright\fi%
8441   }%
8442   {}%
8443   {}%

```

```

8444 \patchcmd{\@makeschapterhead}
8445   {#1}
8446   {\print@leftmargin@eledsection%
8447     #1%
8448     \print@rightmargin@eledsection%
8449   }
8450   {}
8451   {}
8452
8453 \pretocmd{\@sect}
8454   {\let\old@edtext=\edtext
8455     \let\edtext=\dummy@edtext@showlemma%
8456   }
8457   {}
8458   {}
8459
8460 \apptocmd{\@sect}
8461   {\let\edtext=\old@edtext}
8462   {}
8463   {}
8464
8465 \pretocmd{\@ssect}
8466   {\let\old@edtext=\edtext%
8467     \let\edtext=\dummy@edtext@showlemma%
8468   }
8469   {}
8470   {}
8471
8472 \apptocmd{\@ssect}
8473   {\let\edtext=\old@edtext}
8474   {}
8475   {}
8476
8477 %

```

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.

```

8478 \@ifpackageloaded{nameref}{
8479
8480   \patchcmd{\NR@sect}
8481     {#8}
8482     {#8%
8483       \print@rightmargin@eledsection%
8484     }
8485     {}
8486     {}
8487
8488   \patchcmd{\NR@sect}
8489     {\hskip #3\relax}
8490     {\hskip #3\relax%

```

```

8491 \print@leftmargin@eledsection%
8492 }
8493 {}
8494 {}
8495
8496 \patchcmd{\NR@ssect}
8497 {#5}
8498 {#5%
8499 \print@rightmargin@eledsection%
8500 }
8501 {}
8502 {}
8503
8504 \patchcmd{\NR@ssect}
8505 {\hskip #1}
8506 {\hskip #1%
8507 \print@leftmargin@eledsection%
8508 }
8509 {}
8510 {}
8511 }%
8512 {
8513 \patchcmd{\@sect}
8514 {#8}
8515 {#8%
8516 \print@rightmargin@eledsection%
8517 }
8518 {}
8519 {}
8520
8521 \patchcmd{\@sect}
8522 {\hskip #3\relax}
8523 {\hskip #3\relax%
8524 \print@leftmargin@eledsection%
8525 }
8526 {}
8527 {}
8528
8529 \patchcmd{\@ssect}
8530 {#5}
8531 {#5%
8532 \print@rightmargin@eledsection%
8533 }
8534 {}
8535 {}
8536
8537 \patchcmd{\@ssect}
8538 {\hskip #1}
8539 {\hskip #1%
8540 \print@leftmargin@eledsection%

```

```

8541 }
8542 {}
8543 {}
8544 }%
8545 }%
8546 %

```

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 `#`.

```

8547 {}}%
8548 \protect\catcode`\#=6 %Space NEEDS by \catcode
8549 %

```

\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.

```

8550 \AtBeginDocument{%
8551 \patchcmd{\chapter}{\clearforchapter}{%
8552   \if@eled@sectioning\else%
8553   \ifl@dprintingpages\else%
8554   \clearforchapter%
8555   \fi%
8556 \fi%
8557 }%
8558 {}%
8559 {}%
8560
8561 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8562   \if@eled@sectioning\else%
8563   \ifl@dprintingpages%
8564   \endgraf%
8565   \else%
8566   \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
8567   \fi%
8568 \fi%
8569 }%
8570 {}%
8571 {}%
8572 }%
8573 %

```

\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.

```

8574 \newif\if@eled@sectioning%
8575 %

```


We reopen a new \notbool{@noeled@sec} statement, as we will define the \elesection commands.

```
8576 \notbool{@noeled@sec}{%
8577 %
```

XXXII.6 Main code of \eledxxx commands

`\eled@sectioning@out` \eled@sectioning@out is the output file, to dump the pstarts where a sectioning command is used.

```
8578 \newwrite\eled@sectioning@out
8579 %
```

`\eledchapter` And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsubsection
\eledsubsubsection
\eledchapter*
\eledsection*
\eledsubsection*
\eledsubsubsection*
8580 \newcommand{\eledchapter}[2] [] {%
8581   #2%
8582   \ifledRcol%
8583     \immediate\write\eled@sectioningR@out{%
8584       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
8585     }%
8586   \else%
8587     \immediate\write\eled@sectioning@out{%
8588       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
8589     }%
8590   \fi%
8591 }
8592
8593 \newcommand{\eledsection}[2] [] {%
8594   #2%
8595   \ifledRcol%
8596     \immediate\write\eled@sectioningR@out{%
8597       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
8598     }%
8599   \else%
8600     \immediate\write\eled@sectioning@out{%
8601       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
8602     }%
8603   \fi%
8604 }
8605
8606 \newcommand{\eledsubsection}[2] [] {%
8607   #2%
8608   \ifledRcol%
8609     \immediate\write\eled@sectioningR@out{%
8610       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
8611     }%

```

```

8612 \else%
8613 \immediate\write\eled@sectioning@out{%
8614 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
8615 }%
8616 \fi%
8617 }
8618 \newcommand{\eledsubsubsection}[2][{}]{%
8619 #2%
8620 \ifledRcol%
8621 \immediate\write\eled@sectioningR@out{%
8622 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR
8623 }{-}{R}
8624 }%
8625 \else%
8626 \immediate\write\eled@sectioning@out{%
8627 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL
8628 }{-}{-}
8629 }%
8630 \fi%
8631 }
8632 \WithSuffix\newcommand\eledchapter*[2][{}]{%
8633 #2%
8634 \ifledRcol%
8635 \immediate\write\eled@sectioningR@out{%
8636 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8637 }%
8638 \else%
8639 \immediate\write\eled@sectioning@out{%
8640 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{-}
8641 }%
8642 \fi%
8643 }
8644
8645 \WithSuffix\newcommand\eledsection*[2][{}]{%
8646 #2%
8647 \ifledRcol%
8648 \immediate\write\eled@sectioningR@out{%
8649 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8650 }%
8651 \else%
8652 \immediate\write\eled@sectioning@out{%
8653 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{-}
8654 }%
8655 \fi%
8656 }
8657
8658 \WithSuffix\newcommand\eledsubsection*[2][{}]{%
8659 #2%

```

```

8660 \ifledRcol%
8661 \immediate\write\eled@sectioningR@out{%
8662 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{
R}
8663 }%
8664 \else%
8665 \immediate\write\eled@sectioning@out{%
8666 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
}{*}{
8667 }%
8668 \fi%
8669 }
8670
8671 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
8672 #2%
8673 \ifledRcol%
8674 \immediate\write\eled@sectioningR@out{%
8675 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR
}{*}{R}
8676 }%
8677 \else%
8678 \immediate\write\eled@sectioning@out{%
8679 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
}{*}{
8680 }%
8681 \fi%
8682 }
8683 %

```

XXXII.7 Macros written in the auxiliary file

`\eled@chapter`
`\eled@section`
`\eled@subsection`
`\eled@subsubsection`

The sectioning macros, called in the auxiliary file. They have five arguments:

1. Optional arguments of \LaTeX sectioning command.
2. Mandatory arguments of \LaTeX sectioning command.
3. Pstart number.
4. Side: R if right, nothing if left.
5. Starred or not.

```

8684 \def\eled@chapter#1#2#3#4#5{%
8685 \ifstrempy{#4}%
8686 {%
8687 \ifstrempy{#1}%
8688 {%
8689 \csgdef\eled@sectioning@#3#5{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}}%

```

```

8690 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
8691 {#2}}%
8692 }%Need for \pairs, because of using parbox.
8693 {%
8694 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
8695 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
8696 {#2}}%Need for \pairs, because of using parbox.
8697 }%
8698 }%
8699 {\ifstrempy{#1}%
8700 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
8701 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#1}{#2}}}%Bug in LaTeX!
8702 }%
8703 \listcsgadd{eled@sections#5@@}{#3}%
8704 }
8705 \def\eled@section#1#2#3#4#5{%
8706 \ifstrempy{#4}%
8707 {\ifstrempy{#1}%
8708 {%
8709 \csgdef{eled@sectioning@#3#5}{\section{#2}}%
8710 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
8711 {#2}}%Need for \pairs, because of using parbox.
8712 }%
8713 {%
8714 \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
8715 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
8716 {#1}}%Need for \pairs, because of using parbox.
8717 }%
8718 {\ifstrempy{#1}%
8719 {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
8720 {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
8721 }
8722 \listcsgadd{eled@sections#5@@}{#3}%
8723 }
8724 \def\eled@subsection#1#2#3#4#5{%
8725 \ifstrempy{#4}%
8726 {\ifstrempy{#1}%
8727 {%
8728 \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
8729 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{
subsectionmark}{#2}}%Need for \pairs, because of using parbox. \csuse in
case of \subsectionmark is not defined (book)
8730 }%
8731 {%
8732 \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%

```

```

8731 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{
subsectionmark}{#1}}%Need for \pairs, because of using parbox. \csuse in
case of \subsectionmark is not defined (book)
8732 }%
8733 }%
8734 {\ifstrempy{#1}%
8735 {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
8736 {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
8737 }
8738 \listcsgadd{eled@sections#5@@}{#3}%
8739 }
8740 \def\eled@subsubsection#1#2#3#4#5{%
8741 \ifstrempy{#4}%
8742 {\ifstrempy{#1}%
8743 {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
8744 {\csgdef{eled@sectioning@#3#5}{\subsubsection{#1}{#2}}}%
8745 }%
8746 {\ifstrempy{#1}%
8747 {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
8748 {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in
LaTeX!
8749 }
8750 \listcsgadd{eled@sections#5@@}{#3}%
8751 }
8752 %
8753 %

```

End of the conditional test about noeledsec option.

```

8754 }{}
8755 %

```

XXXIII Page breaking or no page breaking depending of specific lines

By default, page breaks are automatic. However, the user can define lines which will force page breaks, or prevent page breaks around one specific line. On the first run, the line-list file records the line number of where the page break is being changed (either forced, or prevented). On the next run, page breaks occur either before or after this line, depending on how the user sets the command. The default setting is after the line.

\normal@page@break \normal@page@break is an etoolbox list which contains the absolute line number of the last line, for each page.

```

8756 \def\normal@page@break{}
8757 %

```

`\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.

```
8758 \def\l@prev@pb{}
8759 \def\l@prev@nopb{}
8760 %
```

`\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.

```
8761 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
8762 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
8763 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
8764 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
8765 %
```

`\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.

```
8766 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
8767 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
8768 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
8769 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
8770 %
```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

```
8771 \def\led@pb@setting{before}
8772 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
8773 %
```

`\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`.

```
8774 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
8775 \newcommand{\led@check@nopb}{%
8776   \IfStrEq{\led@pb@setting}{before}{%
8777     \xifinlist{\the\absline@num}{\l@prev@nopb}%
8778     {\numdef{\abs@prevline}{\the\absline@num-1}%
8779     \xifinlist{\abs@prevline}{\normal@page@break}%
8780     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
8781     }}%
8782   }%}
```

```

8783     {}%
8784   {}%
8785   \IfStrEq{\led@pb@setting}{after}{%
8786     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
8787       \xifinlist{\the\absline@num}{\normal@page@break}%
8788       {\nopagebreak[4]\enlargethispage{\baselineskip}}%
8789     }%
8790 }%
8791   {}%
8792   {}%
8793   {}%
8794 }
8795 %

```

XXXIV Long verse: prevents being separated by a page break

\iflednopbinverse The `\lednopbinverse` boolean is set to false by default. If set to true, `reledmac` will automatically prevent page breaks inside verse. The declaration is made at the beginning of the file, because it is used as a package option.

\check@pb@in@verse The `\check@pb@in@verse` checks if a verse is broken in two page. If true, it adds:

- The absolute line number of the first line of the verse -1 in the `\led@pb` list, if the page break must occur before the verse.
- The absolute line number of the first line of the verse -1 in the `\led@nopb` list, if the page break must occur after the verse.

```

8796 \newcommand{\check@pb@in@verse}{%
8797   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
8798   enabling page breaks in verse control, while on a hanging verse.
8799   \ifnum\page@num=\last@page@num\else%If we have change page
8800     \IfStrEq{\led@pb@setting}{before}{%
8801       \numgdef{\abs@line@verse}{\the\absline@num-1}%
8802       \ledpbnum{\abs@line@verse}%
8803     }{}%
8804     \IfStrEq{\led@pb@setting}{after}{%
8805       \numgdef{\abs@line@verse}{\the\absline@num-1}%
8806       \lednopbnum{\abs@line@verse}%
8807     }{}%
8808   \fi%
8809 }
8810 %

```

XXXV Tools for hyperref package

`\Hy@raisedlink@left` The hyperref package provides a `\Hy@raisedlink` command, to be used to add an anchor to the top of a line and not to the bottom of it.³⁴

However, this command disrupts the line breaking mechanism when it is called before any word. This is why `reledmac` defines `\Hy@raisedlink@left` that is called to the left of words, at the beginning of `\edtext` or inside the `\edlabel` commands.³⁵

```

8811 \def\Hy@raisedlink@left#1{%
8812     \ifvmode
8813         #1%
8814     \else
8815         \Hy@SaveSpaceFactor
8816         \llap{\smash{%
8817             \begingroup
8818                 \let\HyperRaiseLinkLength\@tempdima
8819                 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
8820                 \HyperRaiseLinkHook
8821             \expandafter\endgroup
8822             \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
8823                 \Hy@RestoreSpaceFactor
8824                 #1%
8825                 \Hy@SaveSpaceFactor
8826             }%
8827         }}%
8828         \Hy@RestoreSpaceFactor
8829         \penalty\@M\hskip\z@\relax
8830     \fi
8831 }
8832 %

```

XXXVI Compatibility with eledmac

Here, we define some commands for the `eledmac-compat` option.

```

8833 \ifeledmaccompat%
8834
8835     \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
8836     \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
8837     \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
8838     \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
8839
8840     \unless\ifnocritical@
8841         \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
8842         \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%

```

³⁴<http://tex.stackexchange.com/a/17138/7712>.

³⁵The code is inspired by an answer given by @unbonpetit. Thanks to him. <http://texnique.fr:80/osqa/questions/781/hyraisedlink-perturbe-la-maniere-dont-se-fait-la-coupe-de-ligne/801>.


```

8843 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
8844 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
8845 \let\hsizetwocol\Xhsizetwocol
8846 \let\hsizethreecol\Xhsizethreecol
8847 \let\bhookXnote\Xbhooknote
8848 \let\boxsymlinenum\Xboxsymlinenum
8849 \let\symlinenum\Xsymlinenum
8850 \let\beforenumberinfootnote\Xbeforenumber
8851 \let\afternumberinfootnote\Xafternumber
8852 \let\beforeXsymlinenum\XbeforeXsymlinenum
8853 \let\afterXsymlinenum\XafterXsymlinenum
8854 \let\inplaceofnumber\Xinplaceofnumber
8855 \let\Xlemmaseparator\lemmaseparator
8856 \let\afterlemmaseparator\Xafterlemmaseparator
8857 \let\beforelemmaseparator\Xbeforelemmaseparator
8858 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
8859 \let\txbeforeXnotes\Xtxbeforenotes
8860 \let\afterXrule\Xafterrule
8861 \let\numberonlyfirstinline\Xnumberonlyfirstinline
8862 \let\numberonlyfirstintwoline\Xnumberonlyfirstintwoline
8863 \let\nonumberinfootnote\Xnonumberinfootnote
8864 \let\pstartinfootnote\Xpstart
8865 \let\pstartinfootnoteeverytime\Xpstarteverytime
8866 \let\onlyXpstart\Xonlypstart
8867 \let\Xnonumberinfootnote\Xnonumber
8868 \let\nonbreakableafternumber\Xnonbreakableafternumber
8869 \let\maxhXnotes\Xmaxhnotes
8870 \let\beforeXnotes\Xbeforenotes
8871 \let\boxlinenum\Xboxlinenum
8872 \let\boxlinenumalign\Xboxlinenumalign
8873 \let\boxstartlinenum\Xboxstartlinenum
8874 \let\boxendlinenum\Xboxendlinenum
8875 \let\twoline\Xtwoline
8876 \let\morethantwoline\Xmorethantwoline
8877 \let\twolinebutnotmore\Xtwolinebutnotmore
8878 \let\twolineonlyinsamepage\Xtwolineonlyinsamepage
8879 \fi
8880
8881 \unless\ifnofamiliar@
8882 \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
8883 \fi
8884 \newcommandx{\parafootsep}[2][1,usedefault]{%
8885 \Xparafootsep[#1]{#2}%
8886 \parafootsepX[#1]{#2}
8887 }%
8888
8889 \newcommandx{\afternote}[2][1,usedefault]{%
8890 \Xafternote[#1]{#2}%
8891 \afternoteX[#1]{#2}%
8892 }%

```

```

8893 \unless\ifnoend@
8894 \let\XendXtwolines\Xendtwolines
8895 \let\XendXmorethantwolines\Xendmorethantwolines
8896 \let\hookXendnote\Xendhooknote
8897 \let\boxXendlinenum\Xendboxlinenum%
8898 \let\boxXendlinenumalign\Xendboxlinenumalign%
8899 \let\boxXendstartlinenum\Xendboxstartlinenum%
8900 \let\boxXendendlinenum\Xendboxendlinenum%
8901 \let\XendXlemmaseparator\Xendlemmaseparator
8902 \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
8903 \let\XendXafterlemmaseparator\Xendafterlemmaseparator
8904 \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
8905 \fi
8906
8907 \AtBeginDocument{%
8908 \ifdef\lineref{}\let\lineref\edlineref}%
8909 }%
8910
8911
8912 \fi%
8913 %
8914 %

```

</code>

Appendix A Things to do when changing versions

Appendix A.1 Migrating from edmac to ledmac

If you have never used edmac, ignore this section. If you have used edmac and are starting on a completely new document, ignore this section. Only read this section if you are converting an original edmac document to use ledmac.

The package still provides the original `\text` command, but it is (a) deprecated, and (b) its name has been changed³⁶ to `\critext`; use the `\edtext` macro instead. However, if you do use `\critext` (the new name for `\text`), the following is a reminder.

`\critext` Within numbered paragraphs, footnotes and endnotes are generated by forms of the `\critext` macro:

```
\critext{⟨lemma⟩}⟨commands⟩/
```

The `⟨lemma⟩` argument is the lemma in the main text: `\critext` both prints this as part of the text, and makes it available to the `⟨commands⟩` you specify to generate notes. The `/` at the end terminates the command; it is part of the macro's definition so that spaces after the macro will be treated as significant.

For example:

<pre>I saw my friend \critext{Smith} \Afootnote{Jones C, D.}/ on Tuesday.</pre>	<pre>1 I saw my friend 2 Smith on Tuesday. 2 Smith] Jones C, D.</pre>
---	---

The lemma `Smith` is printed as part of this sentence in the text, and is also made available to the footnote that specifies a variant, `Jones C, D`. The footnote macro is supplied with the line number at which the lemma appears in the main text.

The `⟨lemma⟩` may contain further `\critext` commands. Nesting makes it possible to print an explanatory note on a long passage together with notes on variants for individual words within the passage. For example:

<pre>\critext{I saw my friend \critext{Smith}{\Afootnote{Jones C, D.}/ on Tuesday.} \Bfootnote{The date was July 16, 1954.} /</pre>	<pre>1 I saw my friend 2 Smith on Tuesday. 2 Smith] Jones C, D. 1-2 I saw my friend Smith on Tuesday.] The date was July 16, 1954.</pre>
---	--

However, `\critext` cannot handle overlapping but unnested notes—for example, one note covering lines 10–15, and another covering 12–18; a `\critext` that starts in the `⟨lemma⟩` argument of another `\critext` must end there, too. (The `\lemma` and `\linenum` commands may be used to generate overlapping notes if necessary.)

The second argument of the `\critext` macro, `⟨commands⟩`, is the same as the second argument to the `\edtext` macro.

It is possible to define aliases for `\critext`, which can be easier to type. You can make a single character substitute for `\critext` by saying this:

```
\catcode`\<=\active
```

³⁶A name like `\text` is likely to be defined by other \TeX packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.

```
\let<=\critext
```

Then you might say `<{Smith}\variant{Jones}/`. This of course destroys the ability to use `<` in any new macro definitions, so long as it remains in effect; hence it should be used with care.

Changing the character at the end of the command requires more work:

```
\catcode`\<=\active
\def\xtext#1#2>{\critext{#1}{#2}/}
\let<=\xtext
```

This allows you to say `<{Smith}\Afootnote{Jones}>`.

Aliases for `\critext` of the first kind shown here also can't be nested—that is, you can't use the alias in the text that forms the first argument to `\critext`. (See VI p. 119 to find out why.) Aliases of the second kind may be nested without any problem.

If you really have to use `\critext` in any of the tabular or array environments, then `\edtext` must not be used in the same environment. If you use `\critext` in one of these environments then you have to issue the declaration `\usingcritext` beforehand. The declaration `\usingedtext` must be issued to revert to the default assumption that `\edtext` will be used.

Appendix A.2 Migration from ledmac to eledmac

In `eledmac`, some changes were made in the code to allow easy customization. This may cause problems for people who have already made their own. The next sections explain how to handle this.

If you have created your own series using `\addfootins` and `\addfootinsX`, you must use instead the `\newseries` command (see 6.6.1 p. 33), and remove any `\Xfootnote` command.

If you have customized the `\XXXXXfmt` command, please check whether you can achieve the same by the commands documented for display options (7 p. 34) or `\Xfootnote` options (6.2.2 p. 24). Otherwise please add a new ticket on Github to request a new function for doing this.³⁷

If for some reason you do not want to make the modifications to use the new functions of `eledmac`, you can continue using your own `\XXXXXfmt` command, but you must replace:

```
\renewcommand*{XXXXfmt}[3]
```

with

```
\renewcommandx*{XXXXfmt}[4][4=Z]
```

³⁷<https://github.com/maieul/ledmac/issues>

If you do not make that, you will get a spurious [X], where X is series letter.

If you used a `\protect` command inside a `\footnote` command inside a numbered section, you must change the `\protect` to `\noexpand`. Otherwise the command after the `\protect` will be discarded.

Appendix A.3 Migration to eledmac 1.5.1

The version 1.5.1 corrects a bug in `stanzaindentsrepetition` (cf. 9.3 p. 48). 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. 64.

Appendix A.5 Migration to eledmac 17.1

This version changes the default setting of `\Xpstart`. Henceforth, `pstart` numbers will be printed in footnotes within the section of text where you have called `\numberpstarttrue`.

We do not see any reason to print them in the other sections. However, if you want to print the `pstart` numbers in all of the footnotes, whatever the section, without having to use `\numberpstarttrue`, you can use `\Xpstarteverytime`.

Appendix A.6 Migration to eledmac 1.21.0

Appendix A.6.1 `\Xledsetnormalparstuff` and `\ledsetnormalparstuffX`

The `\ledsetnormalparstuff` has been split into two different commands:

- `\Xledsetnormalparstuff` for critical notes;
- `\ledsetnormalparstuffX` for familiar notes.

Both commands can take an optional argument which is the series letter. If you have redefined `\ledsetnormalparstuff` or any of the commands which call them, you must change them accordingly.

Appendix A.6.2 Endnotes

In any case, delete the `.end` file before the next run.

The previous version of Eledmac had a bug: there were two spaces between the starting page number and the starting line number, but only one space between the ending page number and the ending line number.

As a matter of fact, a spurious space was added after the first `\printnnum`. This spurious space has been deleted. However, if you want to keep the previous spurious space, you may load the package with the `oldprintnnumspace` option.

If you have redefined `\endprint`, you must:

- Contact us and ask for the feature that required your hack, in order to avoid such a hack in the future.
- Use the new fifth argument.
- Add `\xdef\@currentseries{#4}` at the beginning of your own command.

Appendix A.7 Migration to eledmac 1.22.0

The `\ledinnote` command now takes a first optional argument, which is the label for the hyperreference. If you have redefined it, change your redefinition, and check whether you can avoid this redefinition by only redefining `\ledinnotemark`.

Appendix A.8 Migration to eledmac 1.23.0

You must delete the numbered auxiliary files before compiling with the new version of eledmac.

Appendix A.9 Migration from eledmac to reledmac

There are many changes in reledmac which require the user to make modifications.

Appendix A.9.1 Risk of ‘no room for a new’

The risk to obtain a ‘no room for a new something’ error is greater in reledmac than it is in eledmac. See 19.1.3 p. 66 in order to know how to limit it.

Appendix A.9.2 Multiple indices with memoir

Eledmac and ledmac used the specific indexing tools of the memoir class designed to produce multiple indices. However, eledmac could also use imakeidx or indextools tools independently of the memoir class. This system forced to maintain redundant code. Since reledmac, we use only the imakeidx or indextools tools.

Consequently: Users of memoir are invited to use indextool or imakeidx to produce multiple indices.

Appendix A.9.3 Deprecated commands and options

The table of deprecated commands and their alternatives follows. Note that the way some commands must be used may have changed. Please read the handbook.

<i>Deprecated command</i>	<i>Replaced with</i>
<code>\addfootins</code>	<code>\newseries</code>
<code>\addfootinsX</code>	<code>\newseries</code>
<code>\critext</code>	<code>\edtext</code>
<code>\falseverse</code>	<code>\newverse</code>
<code>\interparanoteglue</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\ledchapter</code>	<code>\eledchapter</code>
<code>\ledsection</code>	<code>\eledsection</code>
<code>\ledsetnormalparstuff</code>	<code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code>
<code>\ledsubsection</code>	<code>\eledsubsection</code>
<code>\ledsubsubsection</code>	<code>\eledsubsubsection</code>
<code>\noeledsec</code>	Package option <code>noeledsec</code>
<code>\noendnotes</code>	Package option <code>noendnotes</code>
<code>\pageparbreak</code>	<code>\ledpb</code>

The `ledsecnolinenumber` option has been removed, because it was related to deprecated commands.

The `oldprintnpnumspace` option has been removed too, because it was related to a historical bug. The `\usingedtext` and `\usingcritext` commands are also deprecated.

Appendix A.9.4 `\renewcommand` replaced by command

Many uses of `\renewcommand` have been replaced with uses of specific commands. Please read handbook about specific commands.

<i>Deprecated <code>\renewcommand</code></i>	<i>Replaced with</i>
<code>\@led@extranofeet</code>	<code>\newseries</code>
<code>\apprefprefixmore</code>	<code>\setapprefprefixmore</code>
<code>\apprefprefixsingle</code>	<code>\setapprefprefixsingle</code>
<code>\endstanzaextra</code>	Optional argument of <code>\&</code>
<code>\hangingsymbol</code>	<code>\sethangingsymbol</code>
<code>\ledfootinsdim</code>	<code>\Xmaxhnotes</code> and <code>\maxhnotesX</code>
<code>\parafootftmsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\notenumfont</code>	<code>\Xnotenumfont</code> , <code>\Xendnotenumfont</code> and <code>\notenumfontX</code>
<code>\notefontsetup</code>	<code>\Xnotefontsize</code> , <code>\Xendnotefontsize</code> and <code>\notefontsizeX</code>
<code>\sidenotesep</code>	<code>\setsidenotsep</code>
<code>\startstanzahook</code>	Optional argument of <code>\stanza</code>
<code>\symplinenum</code>	<code>\Xsymplinenum</code>

Appendix A.9.5 Commands the names of which have been changed

In order to help the migration from `eledmac` to `reledmac`, you may load `reledmac` with `eledmac-compat` option. However, it is advised not to, and to change the command names themselves instead. In many cases, you use only a few of them, except the `\footparagraph` command.

<i>Old command</i>	<i>New command</i>
<code>\footparagraph</code>	<code>\Xarrangement</code>
<code>\footnormal</code>	<code>\Xarrangement</code>
<code>\foottwocol</code>	<code>\Xarrangement</code>
<code>\footthreecol</code>	<code>\Xarrangement</code>
<code>\footparagraphX</code>	<code>\arrangementX</code>
<code>\footnormalX</code>	<code>\arrangementX</code>
<code>\foottwocolX</code>	<code>\arrangementX</code>
<code>\footthreecolX</code>	<code>\arrangementX</code>
<code>\afterlemmaseparator</code>	<code>\Xafterlemmaseparator</code>
<code>\afternote</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\afternumberinfootnote</code>	<code>\Xafternumber</code>
<code>\afterXrule</code>	<code>\Xafterrule</code>
<code>\afterXsymplinenum</code>	<code>\Xaftersymplinenum</code>
<code>\beforelemmaseparator</code>	<code>\Xbeforelemmaseparator</code>
<code>\beforenumberinfootnote</code>	<code>\Xbeforenumber</code>
<code>\beforeXnotes</code>	<code>\Xbeforenotes</code>
<code>\beforeXsymplinenum</code>	<code>\Xbeforesymplinenum</code>

<i>Old command</i>	<i>New command</i>
<code>\bhookXnote</code>	<code>\Xbhookendnote</code>
<code>\bhookXnote</code>	<code>\Xbhooknote</code>
<code>\boxendlinenum</code>	<code>\Xboxendlinenum</code>
<code>\boxlinenum</code>	<code>\Xboxlinenum</code>
<code>\boxlinenumalign</code>	<code>\Xboxlinenumalign</code>
<code>\boxstartlinenum</code>	<code>\Xboxstartlinenum</code>
<code>\boxsymlinenum</code>	<code>\Xboxsymlinenum</code>
<code>\boxXendlinenum</code>	<code>\Xendboxlinenum</code>
<code>\boxXendlinenumalign</code>	<code>\Xendboxlinenumalign</code>
<code>\boxXendstartlinenum</code>	<code>\boxXendstartlinenum</code>
<code>\letboxXendendlinenum</code>	<code>\Xendletboxendlinenum</code>
<code>\hsizetwocol</code>	<code>\Xhsizetwocol</code>
<code>\hsizethreecol</code>	<code>\Xhsizethreecol</code>
<code>\inplaceoflemmaseparator</code>	<code>\Xinplaceoflemmaseparator</code>
<code>\inplaceofnumber</code>	<code>\Xinplaceofnumber</code>
<code>\lemmaseparator</code>	<code>\Xlemmaseparator</code>
<code>\maxhXnotes</code>	<code>\Xmaxhnotes</code>
<code>\morethantwolines</code>	<code>\Xmorethantwolines</code>
<code>\nonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\notesXwidthliketwocolumns</code>	<code>\noteswidthliketwocolumnsX</code>
<code>\noXlemmaseparator</code>	<code>\Xnolemmaseparator</code>
<code>\numberonlyfirstinline</code>	<code>\Xnumberonlyfirstinline</code>
<code>\numberonlyfirstintwolines</code>	<code>\Xnumberonlyfirstintwolines</code>
<code>\nonbreakableafternumber</code>	<code>\Xnonbreakableafternumber</code>
<code>\onlyXpstart</code>	<code>\Xonlypstart</code>
<code>\parafootsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\pstartinfootnote</code>	<code>\Xpstart</code>
<code>\pstartinfootnoteeverytime</code>	<code>\Xpstarteverytime</code>
<code>\symlinenum</code>	<code>\Xsymlinenum</code>
<code>\twolines</code>	<code>\Xtwolines</code>
<code>\twolinesbutnotmore</code>	<code>\Xtwolinesbutnotmore</code>
<code>\twolinesonlyinsamepage</code>	<code>\Xtwolinesonlyinsamepage</code>
<code>\txtbeforeXnotes</code>	<code>\Xtxtbeforenotes</code>
<code>\XendXafterlemmaseparator</code>	<code>\Xendafterlemmaseparator</code>
<code>\XendXbeforelemmaseparator</code>	<code>\Xendbeforelemmaseparator</code>
<code>\XendXinplaceoflemmaseparator</code>	<code>\Xendinplaceoflemmaseparator</code>
<code>\XendXlemmaseparator</code>	<code>\Xendlemmaseparator</code>
<code>\XendXmorethantwolines</code>	<code>\Xendmorethantwolines</code>
<code>\XendXtwolines</code>	<code>\Xendtwolines</code>
<code>\Xnonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\lineref</code>	<code>\edlineref</code>

Appendix A.9.6 Endnotes

With `reledmac`, there is now one auxiliary file for every endnotes set (`.Aend`, `.Bend`, `.Cend` etc.). If you have overridden `\doendnotes` (which you would not have done) you must adapt your code.

Appendix A.9.7 Z Series

The ‘Z’ series of notes has been removed. Only five series are provided now by default: A, B, C, D, E.

Appendix A.9.8 Internal commands

Users who have overridden internal commands, which is wrong, must adapt according to the following. Or better, they should not override any of such commands and use `reledmac` options instead.

- If you have modified `\Xfootfmt`, note that the fourth argument is now mandatory.
- `\unvxh` has been replaced with `\Xunvxh` and `\unvxhX` with two mandatory arguments.

Appendix A.10 Migration to `reledmac` 2.1.0

`Reledmac` 2.1.0 fix some bugs when using `\Xbhooknote` and `\bhooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }.`
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` in the argument of `\Xbhooknote/\bhooknoteX`.

Appendix A.11 Migration to `reledmac` 2.1.3

`Reledmac` 2.1.3 fix an historical bug, (style in `ledmac` 0.7!) which doubled the space before the rules of paragraphed familiar footnotes. Consequently, if you use paragraphed familiar footnotes, you should maybe adapt it, playing with `\beforenotesX`.

Appendix A.12 Migration to `reledmac` 2.3.0

Before `reledmac` 2.3.0, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use optional argument of `\stanza` or `\endverse`.

Appendix A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{\langle...\rangle}` to the use of `\Xlinangeseparator` or `/` and `\Xendlinangeseparator` (7.2.4 p. 36).

Appendix A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 7.3 p. 40.

`\xlineref` does not print anymore the side flag (R for right side), because it is incompatible with numerical test. Use `\xflagref` to obtain it.

The `\printlines` and `\printendlines` commands take now an eighth argument, which is the side flag. It is strongly recommended to NEVER redefine these two commands and to use the setting commands instead (or to ask for new setting commands if the actual does not answer to your needs). However, if you have done it, just change your redefinition to have a new argument.

It is strongly recommended to stop redefining `\fullstop` and to use `\Xsublinesep` instead.

Appendix A.15 Migration to reledmac 2.7.0

`\Serefonlypage` (introduced in reledmac 2.5.0) added an parenthesis after the page number. This was just an error, linked to a bad imitation of `\Serefwithpage`. That has been deleted. And so, the `\XendafterpagenumberSerefonlypage` to set it was also deleted.

`\rigidbalance` is split to two new commands: `\Xrigidbalance` for critical footnotes and `\rigidbalanceX` for familiar footnotes. If you have redefined it — but why should you have ? —, you should split your single redefinition in two redefinitions.

Appendix A.16 Migration to reledmac 2.7.2

`\Xhsize` is already defined in the `floatrow` package. It becomes `\Xwidth`, and, consequently, `\hsizeX` becomes `\widthX`.

The ancient names are temporarily maintained as aliases.

Appendix A.17 Migration to reledmac 2.8.0

Reledmac 2.8.0 fix spurious indents for paragraphed critical and familiar footnotes in `ledgroup` and `minipage`. You can re-establish the indent with `\Xparinden` and `\parindentX`.

Appendix A.18 Migration to reledmac 2.13.1

Reledmac 2.5.0 added a bug, which makes the right flag to be printed on the right side of critical footnotes, even if not explicitly requested by using `\Xlineflag`.

Version 2.13.1 solves this issue. Please use `\Xlineflag` if you want to add the right flag.

References

- [Bre96] Herbert Breger. `tabmac`. October 1996. (Available from CTAN in `macros/plain/contrib/tabmac`)
- [Bur01] John Burt. ‘Typesetting critical editions of poetry’. *TUGboat*, **22**, 4, pp 353–361, December 2001. (Code available from CTAN in `macros/latex/contrib/poemscol`)
- [Eck03] Matthias Eckermann. *The Parallel-Package*. April 2003. (Available from CTAN in `macros/latex/contrib/parallel`)
- [Fai03] Robin Fairbairns. *footmisc—a portmanteau package for customising footnotes in L^AT_EX*. February 2003. (Available from CTAN in `macros/latex/contrib/footmisc`)
- [LW90] John Lavagnino and Dominik Wujastyk. ‘An overview of `edmac`: a PLAIN TeX format for critical editions’. *TUGboat*, **11**, 4, pp. 623–643, November 1990. (Code available from CTAN in `macros/plain/contrib/edmac`)
- [Lüc03] Uwe Lück. ‘`ednotes`—critical edition typesetting with LaTeX’. *TUGboat*, **24**, 2, pp. 224–236, 2003. (Code available from CTAN in `macros/latex/contrib/ednotes`)
- [Sul92] Wayne G. Sullivan. *The file `edstanzan.doc`*. June 1992. (Available from CTAN in `macros/plain/contrib/edmac`)
- [Wil02] Peter Wilson. *The memoir class for configurable typesetting*. November 2002. (Available from CTAN in `macros/latex/contrib/memoir`)
- [Wil04] Peter Wilson and Maïeul Rouquette. *Parallel typesetting for critical editions: the `eledpar` package*. December 2004. (Available from CTAN in `macros/latex/contrib/ledmmac`)

Index

Symbols

<code>\&</code>	48
<code>\@EDROWFILL@</code>	1
<code>\@adv</code>	1
<code>\@advancestanzanumber</code>	1
<code>\@beforeinsertofthisedtext</code>	1
<code>\@doclearpage</code>	1
<code>\@doreinfeetX</code>	1
<code>\@edindex@fornote@</code>	1
<code>\@edindex@hyperref</code>	1
<code>\@edrowfill@</code>	1
<code>\@edtext@level</code>	1

\@emptytoks	1
\@fnpos	1
\@footnotemark	1
\@footnotetext	1
\@getfirstseries	1
\@gobblefive	1
\@gobblefour	1
\@gobblethree	1
\@h	1
\@hangingsymbol	1
\@iiiminipage	1
\@insertstanza	1
\@k	1
\@l@dttempcnta	1
\@l@dttempcntb	1
\@lab	1
\@led@testifnofoot	1
\@lemma	1
\@line@num	1
\@lock	1
\@lopL	1
\@lopR	1
\@makechapterhead	1
\@makeschapterhead	1
\@mem@extranofeet	1
\@mem@old@ssect	1
\@mpfnpos	1
\@msd	1
\@msd@c	1
\@msd@options@iffullpage	1
\@msdata@list	1
\@nl	1
\@nl@reg	1
\@opXfeet	1
\@pend	1
\@pendR	1
\@ref	1
\@ref@reg	1
\@ref@reg@parse	1
\@sect	1
\@series	1
\@set	1
\@sidenotesep	1
\@ssect	1
\@startstanza	1
\@stopmsd	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1

\@xloop	1
\@xympar	1
CLASSarticle	66
CLASSbook	66, 332
CLASSmemoir	191, 192, 251–253, 285, 332, 351, 411, 415
CLASSscrbook	415
COMMAND*footnote	68
COMMAND\...\@footnotemark...	195
COMMAND\...d@ta	143
COMMAND\<hook	
@<series	240
COMMAND\<hookname	
<pseudoserries	242
COMMAND\<type	
footfmt	181
COMMAND\@@line	173
COMMAND\@MM	159, 412
COMMAND\@Rlineflag	286, 412
COMMAND\@Serefprefix	264
COMMAND\@Serefprefixmore	264
COMMAND\@add@	322
COMMAND\@adv	104
COMMAND\@apprefprefixmore	264
COMMAND\@apprefprefixsingle	264
COMMAND\@beforeinsertofthisedtext	126
COMMAND\@bsphack	254
COMMAND\@doclearpage	252, 253, 405, 415
COMMAND\@doreinfeetX	415
COMMAND\@dprintingcolumns	412
COMMAND\@edindex@hyperref	287, 288
COMMAND\@edtext@level	123
COMMAND\@esphack	254
COMMAND\@fnpos	213, 248
COMMAND\@footnotemark	192, 405, 415
COMMAND\@footnotetext	192, 193, 405
COMMAND\@gobble	121, 122, 235
COMMAND\@gobblefive	236, 414
COMMAND\@gobblefour	411
COMMAND\@gobblethree	404
COMMAND\@h	176
COMMAND\@hangingsymbol	291
COMMAND\@iiiminipage	276, 278, 404, 415
COMMAND\@iiiminpage	276
COMMAND\@l	410
COMMAND\@l@dtempcnta	145, 147, 154
COMMAND\@l@dtempcntb	147
COMMAND\@l@reg	410
COMMAND\@lab	101, 254, 257, 261, 404
COMMAND\@ldunboxmpfoot	279
COMMAND\@led@extranofeet	352

COMMAND\@ledinnote@command	283
COMMAND\@lemma	125, 127
COMMAND\@lock	95, 291
COMMAND\@lopL	405
COMMAND\@lopR	405
COMMAND\@makecol	248, 249, 251, 252, 415
COMMAND\@mpfnpos	213
COMMAND\@msd	300
COMMAND\@msd@c	300
COMMAND\@msd@options@iffullpage	306
COMMAND\@msdata@list	300, 301
COMMAND\@nl	101–104, 106, 115, 257, 404, 405
COMMAND\@nl@reg	102, 349, 405, 410
COMMAND\@opXfeet	405
COMMAND\@opfeetX	415
COMMAND\@opxtrafeeti	415
COMMAND\@page	103, 257
COMMAND\@pend	405
COMMAND\@pendR	405
COMMAND\@ref	101, 111–113, 116, 121
COMMAND\@ref@later	112, 116
COMMAND\@ref@reg	111, 405
COMMAND\@ref@reg@parsearg	112
COMMAND\@reinserts	248, 250–252, 415
COMMAND\@secondoftwo	68
COMMAND\@sect	334
COMMAND\@series	238
COMMAND\@set	105
COMMAND\@sidenotesep	275
COMMAND\@stopmsd	300
COMMAND\@sw	112, 113, 129, 132
COMMAND\@tag	122, 123, 127
COMMAND\@tempcnta	81
COMMAND\@tempcntb	81
COMMAND\@toksa	87
COMMAND\@toksb	87
COMMAND\@xloop	155
COMMAND\@xympar	268, 415
COMMAND\Aendnote	15, 24
COMMAND\Afootfmt	159
COMMAND\Afootgroup	159
COMMAND\Afootnote	8, 15, 23, 24, 27, 124, 167, 191, 214, 229, 414
COMMAND\Afootstart	159
COMMAND\AtBeginDocument	252
COMMAND\AtEveryPend	18, 50, 138, 412, 413, 415
COMMAND\AtEveryPstart	18, 50, 412, 413, 415, 418
COMMAND\AtEveryStanza	50, 419
COMMAND\AtEveryStopStanza	50, 419
COMMAND\Bendnote	15, 23
COMMAND\Bfootnote	8, 15, 191, 214, 229

COMMAND\Centering	44
COMMAND\Cfootnote	191
COMMAND\Columns	82, 164
COMMAND\Dfootnote	191
COMMAND\Efootnote	191
COMMAND\Gls	60
COMMAND\Hy@raisedlink	344
COMMAND\Hy@raisedlink@left	344
COMMAND\LTR	43
COMMAND\NR	334
COMMAND\Pages	82, 249, 250
COMMAND\ProcessOptionsX	72
COMMAND\RL	42
COMMAND\RaggedLeft	44
COMMAND\RaggedRight	44
COMMAND\SEonlypage	262, 417
COMMAND\SEref	54–56, 262, 265, 418, 420
COMMAND\SErefonlypage	54–56, 355, 417
COMMAND\SErefwithpage	54–56, 262, 265, 355, 417, 419
COMMAND\Stanza	410
COMMAND\Waklam	322
COMMAND\X@doreinfeet	250, 415
COMMAND\XR@prefix	268
COMMAND\XR@test	268
COMMAND\XR@test@mac	268
COMMAND\XR@test@mac@test	268
COMMAND\XXXXXXfmt	348
COMMAND\XXXXXfmt	348
COMMAND\Xafterlemmaseparator	41, 352
COMMAND\Xafternote	44, 45, 351, 352
COMMAND\Xafternumber	38, 39, 352
COMMAND\Xafterrule	46, 214, 352, 410, 413
COMMAND\Xaftersymlinenum	39, 352
COMMAND\Xarrangement	35, 45, 68, 160, 161, 240, 352
COMMAND\Xarrangement@footparagraph	166
COMMAND\Xarrangement@normal	161
COMMAND\Xarrangement@paragraph	165
COMMAND\Xbeforeinserting	43
COMMAND\Xbeforelemmaseparator	40, 352
COMMAND\Xbeforenotes	45, 214, 352, 410, 413
COMMAND\Xbeforenumber	36, 38, 39, 352
COMMAND\Xbeforesymlinenum	39, 352
COMMAND\Xbhookendnote	353
COMMAND\Xbhookgroup	45, 417, 418
COMMAND\Xbhooknote	43, 353, 354, 415, 416
COMMAND\Xboxendlinum	39, 40, 353, 414
COMMAND\Xboxlinenum	39, 40, 353
COMMAND\Xboxlinenumalign	39, 40, 353, 414
COMMAND\Xboxstartlinenum	39, 40, 353, 414
COMMAND\Xboxsymlinenum	39, 353

COMMAND\Xcolalign	43, 413
COMMAND\Xdo@feet	415, 420
COMMAND\Xend	236
COMMAND\XendXafterlemmaseparator	353
COMMAND\XendXbeforelemmaseparator	353
COMMAND\XendXinplaceoflemmaseparator	353
COMMAND\XendXlemmaseparator	353
COMMAND\XendXmorethantwolines	353
COMMAND\XendXtwolines	353
COMMAND\Xendafterenumber	38, 416
COMMAND\Xendafterlemmaseparator	41, 353
COMMAND\Xendafternote	47, 418
COMMAND\Xendafternumber	40
COMMAND\Xendafterpagenumbe	419
COMMAND\Xendafterpagenumber	40, 55
COMMAND\XendafterpagenumberSErefonlypage	355
COMMAND\Xendaftersymlinenum	39, 40, 416
COMMAND\Xendahookinplaceofnumber	40, 416
COMMAND\Xendahooklinenum	40, 416
COMMAND\Xendbeforelemmaseparator	41, 353
COMMAND\Xendbeforelinenum	40
COMMAND\Xendbeforenumber	38, 416
COMMAND\Xendbeforepagenumber	40, 55
COMMAND\XendbeforepagenumberSErefonlypage	55
COMMAND\Xendbeforesymlinenum	39, 40, 416
COMMAND\Xendbhookinplaceofnumber	40, 416
COMMAND\Xendbhooklinenum	40, 416
COMMAND\Xendbhooknote	43
COMMAND\Xendboxendlinenum	40, 414
COMMAND\Xendboxlinenum	40, 353, 412
COMMAND\Xendboxlinenumalign	40, 353, 414
COMMAND\Xendboxstartlinenum	40, 414
COMMAND\Xendboxsymlinenum	39, 416
COMMAND\Xendhangindent	43, 416, 418
COMMAND\Xendinplaceoflemmaseparator	24, 41, 353
COMMAND\Xendinplaceofnumber	39, 415
COMMAND\Xendinplaceofpagenumber	36, 420
COMMAND\Xendinsertsep@	220
COMMAND\Xendlemmadisablefontselection	42
COMMAND\Xendlemmafont	42, 417
COMMAND\Xendlemmaseparator	25, 41, 353
COMMAND\Xendletboxendlinenum	353
COMMAND\Xendlineflag	55
COMMAND\Xendlineprefixmore	40, 55
COMMAND\Xendlineprefixsingle	40, 56
COMMAND\Xendlinerangeseparator	36, 56, 180, 355, 416
COMMAND\Xendmorethantwolines	24, 37, 56, 353, 413, 414
COMMAND\Xendnonumber	37, 415
COMMAND\Xendnote	217, 235, 236, 413
COMMAND\Xendnotefontsize	42, 352

COMMAND\Xendnotenumfont	40, 41, 352
COMMAND\Xendnotes	220
COMMAND\Xendnumberonlyfirstinline	35, 416
COMMAND\Xendnumberonlyfirstintwoline	35, 416
COMMAND\Xendpagenumberonlyfirst	36, 420
COMMAND\Xendpagenumberonlyfirstifsingle	36, 420
COMMAND\Xendpagenumberonlyfirstintwo	36, 420
COMMAND\Xendparagraph	47, 410
COMMAND\Xendsep	47
COMMAND\Xendsublinesep	38, 56, 180
COMMAND\Xendsymlinenum	35, 416
COMMAND\Xendsympagenum	36, 420
COMMAND\Xendtwoline	24, 37, 56, 353, 413, 414
COMMAND\Xendtwolinebutnotmore	37, 56, 413, 414
COMMAND\Xendtwolineonlyinsamepage	37, 56, 413, 414
COMMAND\Xendwrapcontent	42, 419
COMMAND\Xendwraplemma	42, 419
COMMAND\Xfootfmt	354
COMMAND\Xfootgroup	165
COMMAND\Xfootins	164
COMMAND\Xfootnote	52, 58, 122, 348, 407, 411, 413, 417, 419
COMMAND\Xfootstarts	165
COMMAND\Xgroupbyline	45, 153, 190
COMMAND\Xgroupbylines	420
COMMAND\Xgroupbylineseparatetwoline	45
COMMAND\Xhangindent	43, 416
COMMAND\Xhsize	355, 417, 418
COMMAND\Xsizethreecol	44, 46, 353
COMMAND\Xsizetwocol	44, 46, 241, 353
COMMAND\Xinplaceoflemmaseparator	24, 41, 353
COMMAND\Xinplaceofnumber	39, 353, 413, 414
COMMAND\Xinsertparafootsep	170, 172
COMMAND\Xledsetnormalparstuff	350, 351, 413
COMMAND\Xlemmadisablefontselection	41
COMMAND\Xlemmafont	42, 417
COMMAND\Xlemmaseparator	40, 41, 180, 243, 245, 248, 353
COMMAND\Xlineflag	55, 355, 356, 419
COMMAND\Xlinerangeseparator	36, 55, 180, 355, 416
COMMAND\Xmaxhnotes	46, 66, 68, 352, 353, 410, 412
COMMAND\Xmorethantwoline	24, 37, 55, 353, 412
COMMAND\Xnoindent	416
COMMAND\Xnolemmaseparator	41, 248, 353
COMMAND\Xnonbreakableafternumber	38, 353, 408
COMMAND\Xnonumber	37, 353
COMMAND\Xnonumberinfootnote	353
COMMAND\Xnotefontsize	42, 352
COMMAND\Xnotefontsize@<s>	170, 174, 176
COMMAND\Xnotenumfont	41, 352
COMMAND\Xnoteswidthliketwocolumns	47, 411
COMMAND\Xnumberonlyfirstinline	35, 36, 45, 98, 242, 243, 245, 353, 407, 412

COMMAND\Xnumberonlyfirstintwolines	35, 45, 353, 407
COMMAND\Xonlypstart	38, 353, 407, 412
COMMAND\Xpagelinesep	38, 420
COMMAND\Xparafootsep	44, 45, 98, 352, 353, 419
COMMAND\Xparafootsep@series	170
COMMAND\Xparinden	355
COMMAND\Xparindent	42, 413, 416, 418
COMMAND\Xprenotes	45, 214, 419
COMMAND\Xprenotes@	164, 214, 407
COMMAND\Xpstart	37, 38, 350, 353, 407, 412
COMMAND\Xpstarteverytime	38, 350, 353, 412
COMMAND\Xragged	44
COMMAND\Xrigidbalance	173, 355, 417
COMMAND\Xstanza	38, 50
COMMAND\Xstanzaseparator	38
COMMAND\Xsublinesep	21, 38, 55, 180, 355
COMMAND\Xsublinesepside	21, 38
COMMAND\Xsymlinenum	35, 44, 352, 353, 414
COMMAND\Xtoendnotes	25, 236
COMMAND\Xtwolines	24, 37, 55, 186, 187, 242, 353, 412
COMMAND\Xtwolinesappref	242
COMMAND\Xtwolinesbutnotmore	37, 55, 353, 413
COMMAND\Xtwolinesbutnotmoreappref	242
COMMAND\Xtwolinesonlyinsamepage	37, 55, 353, 413
COMMAND\Xtxtbeforenotes	45, 353, 419, 420
COMMAND\Xunvxh	168, 354
COMMAND\Xwidth	46, 355, 418
COMMAND\Xwrapcontent	42, 419
COMMAND\Xwraplemma	42, 43, 419
COMMAND\&	352
COMMAND\(\XXX)vfootnote	190
COMMAND\absline@num	95, 144
COMMAND\accent	121
COMMAND\actionlines@list	96, 146
COMMAND\actions@list	96
COMMAND\add@Xgroupbyline	153
COMMAND\add@inserts	96, 152, 153
COMMAND\add@inserts@next	152, 153
COMMAND\add@msd@	300
COMMAND\add@msdata	300, 301
COMMAND\add@msdata@firstlineofpage	303
COMMAND\add@msddata	300
COMMAND\add@penalties	144, 154
COMMAND\addcontentsline	329
COMMAND\addfootins	348, 351
COMMAND\addfootinsX	348, 351
COMMAND\addtoennotes	236
COMMAND\advancelabel@refs	256
COMMAND\advanceline	21, 22, 97, 104, 117, 415
COMMAND\affixlin@num	275

COMMAND\affixline@num	147, 150, 151, 405
COMMAND\affixpstart@num	151
COMMAND\afterXrule	352
COMMAND\afterXsymlinenum	352
COMMAND\afterenumber	38
COMMAND\aftergroup	120, 125
COMMAND\afterlemmaseparator	352
COMMAND\afternote	352
COMMAND\afternoteX	44, 351, 352
COMMAND\afternumberinfootnote	352
COMMAND\afterruleX	46, 410, 413
COMMAND\applabel	54, 257, 258, 413, 419
COMMAND\appref	52, 54–56, 262, 265, 417, 418
COMMAND\apprefprefixmore	54, 352
COMMAND\apprefprefixsingle	54, 352
COMMAND\apprefwithpage	54–56, 262, 265, 414, 417
COMMAND\arrangementX	35, 68, 195, 240, 352
COMMAND\arrangementX@normal	201
COMMAND\article	14
COMMAND\at@every@pend	138
COMMAND\autopar	17, 135, 138, 139, 211, 406, 408, 409, 413
COMMAND\ballast	67
COMMAND\ballast@count	144, 154
COMMAND\baselineskip	35, 166, 170
COMMAND\beforeXnotes	352
COMMAND\beforeXsymlinenum	352
COMMAND\beforeeledchapter	10, 64, 330, 331
COMMAND\beforeinsertingX	43
COMMAND\beforelemmaseparator	352
COMMAND\beforenotesX	45, 354, 409, 410, 413
COMMAND\beforenumberinfootnote	352
COMMAND\begin	307
COMMAND\beginnumbering	16, 18, 19, 82, 83, 85, 94, 99, 114, 138, 216, 299, 407, 410, 414, 415, 420, 421
COMMAND\bf	407
COMMAND\bfseries	41, 407
COMMAND\bhookXnote	353
COMMAND\bhookgroupX	45, 417
COMMAND\bhooknoteX	43, 354, 415, 416
COMMAND\body	292
COMMAND\bodyfootmarkA	32
COMMAND\book	14
COMMAND\boxXendlinenum	353
COMMAND\boxXendlinenumalign	353
COMMAND\boxXendstartlinenum	353
COMMAND\boxendlinenum	353
COMMAND\boxlinefootnote	183
COMMAND\boxlinenum	353
COMMAND\boxlinenumalign	353
COMMAND\boxstartlinenum	353

COMMAND\boxsymlinenum	353
COMMAND\break	35, 168
COMMAND\brokenpenalty	154
COMMAND\centering	44
COMMAND\ch@ck@l@ck	405
COMMAND\ch@cksub@l@ck	150, 405
COMMAND\chapter	63, 332, 336, 410, 413, 415, 419
COMMAND\chaptermark	330
COMMAND\check@pb@in@verse	343
COMMAND\colalignX	43, 413
COMMAND\collect@body	308
COMMAND\color	419
COMMAND\colorbox	68
COMMAND\columns	47
COMMAND\columnwidth	166, 411
COMMAND\command names	241, 242
COMMAND\copyright	121
COMMAND\correct@Xfootins@box	412
COMMAND\correct@footinsX@box	412
COMMAND\count	174
COMMAND\critex	406
COMMAND\critext	127, 347, 348, 351
COMMAND\csname	73, 131
COMMAND\ctab	323, 328
COMMAND\ctabtext	328
COMMAND\dcoll	317
COMMAND\def	70
COMMAND\detokenize	131
COMMAND\dimen	174
COMMAND\dimexpr	46
COMMAND\discretionary	168
COMMAND\displaywidowpenalty	154
COMMAND\do@Xfeet	249, 405, 415, 420
COMMAND\do@actions	144–146, 405
COMMAND\do@actions@fixedcode	405
COMMAND\do@actions@next	145, 146
COMMAND\do@ballast	144, 154
COMMAND\do@feet@custom@order	248
COMMAND\do@insidelinehook	408
COMMAND\do@line	96, 120, 137, 140, 143, 152, 154, 292, 405, 406, 408, 410
COMMAND\do@linehook	405
COMMAND\do@lockoff	97
COMMAND\do@lockon	97
COMMAND\do@reinxtrafeet	404
COMMAND\doendnotes	25, 220, 221, 354, 414
COMMAND\doendnotesbysection	25, 221, 237, 414
COMMAND\doennotes	421
COMMAND\doinsidelinehook	22, 411
COMMAND\dolinehook	22, 411
COMMAND\doreinxtrafeeti	415

COMMAND\doreintrafeetii	415
COMMAND\doxtrafeet	248, 404
COMMAND\doxtrafeeti	415
COMMAND\doxtrafeetii	415
COMMAND\dummy@ref	121
COMMAND\edaftertab	63, 323, 324
COMMAND\edatleft	62, 321
COMMAND\edatright	62, 63, 321
COMMAND\edbeforetab	63, 323
COMMAND\edfilldimen	322
COMMAND\edfont@info	127
COMMAND\edgls	60, 282
COMMAND\edgls...	418
COMMAND\edindex	58–60, 281, 282, 285, 287, 288, 311, 408, 411, 412, 415, 416, 420, 421
COMMAND\edindexlab	59, 60
COMMAND\edlabel	52–54, 56, 121, 254, 256, 257, 259, 260, 267, 282, 311, 344, 404, 407–409, 412, 417
COMMAND\edlabelE	54, 258
COMMAND\edlabelS	54, 258
COMMAND\edlabelSE	54
COMMAND\edlineref	52, 254, 353, 412, 414, 417, 421
COMMAND\edmakelabel	53, 267
COMMAND\edpageref	52, 254, 259, 267
COMMAND\edrowfill	322
COMMAND\edsublineref	52
COMMAND\edtabcolsep	316
COMMAND\edtext	6, 23, 24, 26–29, 47, 52–54, 58, 61, 68, 95, 96, 111, 113, 116, 119–127, 129, 130, 132, 133, 257, 258, 261, 311, 313, 332, 344, 347, 348, 351, 404, 406, 408, 410–414, 419, 420
COMMAND\edtext@level	414
COMMAND\edtextlater	113
COMMAND\edvertdots	63, 321
COMMAND\edvertline	63, 321
COMMAND\elechapter	64
COMMAND\eled@sectioning@out	337
COMMAND\eledchapter	64, 351, 411, 415
COMMAND\eledchapter*	64
COMMAND\eledmac@error	404
COMMAND\eledsection	6, 15, 64, 121, 142, 330, 351, 413
COMMAND\eledsection*	64
COMMAND\eledsubsection	64, 351
COMMAND\eledsubsection*	64
COMMAND\eledsubsubsection	64, 351
COMMAND\eledsubsubsection*	64
COMMAND\eledxxx	10, 65, 336, 337, 410
COMMAND\eledxxxx	330
COMMAND\elesection	337
COMMAND\else	281, 330
COMMAND\empty	81, 148, 254
COMMAND\end	307
COMMAND\end@lemmas	120
COMMAND\endashchar	180

COMMAND\endgraf	137, 170, 211
COMMAND\endlock	21, 97, 118, 296
COMMAND\endminipage	276, 278, 404, 415
COMMAND\endmsdata	31
COMMAND\endnotes	413, 417
COMMAND\endnumbering	16, 19, 82–85, 405, 414, 420, 421
COMMAND\endprint	217, 219, 236, 350
COMMAND\endstanzaextra	352
COMMAND\endsub	21, 97, 117
COMMAND\endverse	354
COMMAND\everypar	139
COMMAND\extensionchars	66, 82
COMMAND\externaldocument	56, 267
COMMAND\f@x@l@cks	405
COMMAND>falseverse	351, 408, 410
COMMAND\fi	330
COMMAND\firstlinenum	19, 147, 406
COMMAND\firstsublinenum	19, 406
COMMAND\fix@page	102, 103, 405
COMMAND\flag@end	116, 126, 410
COMMAND\flag@end@later	116
COMMAND\flag@start	116, 126, 410, 411
COMMAND\flag@start@later	116
COMMAND\flagstanza	51
COMMAND\floatingpenalty	159, 412
COMMAND\flush@notes	155
COMMAND\fnpos	33, 213, 409, 420
COMMAND\footfmt	159, 161
COMMAND\footfmt...	196
COMMAND\footfootmarkA	32
COMMAND\footfudgefactor	168
COMMAND\footfudgefiddle	68, 166, 404
COMMAND\footgroup	159
COMMAND\footins	164
COMMAND\footnormal	241, 352, 404
COMMAND\footnormalX	352
COMMAND\footnote	32, 67, 191–193, 349, 405
COMMAND\footnote@lang	180
COMMAND\footnoteA	16, 32
COMMAND\footnoteB	16
COMMAND\footnoteC	23
COMMAND\footnoteE	32
COMMAND\footnoteX	8, 233, 235, 419
COMMAND\footnoteX@reading	234
COMMAND\footnoteXmk	247
COMMAND\footnote⟨X⟩	122
COMMAND\footnotelang@lua	158
COMMAND\footnotelang@poly	158
COMMAND\footnoteoption@	157, 416
COMMAND\footnoterule	174

COMMAND\footnotesize	42
COMMAND\footparagraph	166, 241, 352, 410
COMMAND\footparagraphX	206, 352, 410
COMMAND\footplitskips	405, 412
COMMAND\footstart	159, 164, 174
COMMAND\footstrut	170
COMMAND\footthreecol	352
COMMAND\footthreecolX	352, 414
COMMAND\foottwocol	352
COMMAND\foottwocolX	352, 414
COMMAND\foreignlanguage	42
COMMAND\fullstop	355
COMMAND\get@edindex@hyperref	287
COMMAND\get@edindex@ledinnote@command	283
COMMAND\get@fnmark	193
COMMAND\get@index@command	409
COMMAND\get@linelistfile	405
COMMAND\get@thisfootnote	199
COMMAND\getline@num	144, 145
COMMAND\gl@p	87
COMMAND\global	101
COMMAND\globaldefs	101
COMMAND\gls	60, 289
COMMAND\hangindentX	43, 413, 416
COMMAND\hangingsymbol	352, 406
COMMAND\hbox	168
COMMAND\hfill	409
COMMAND\hidenumbering	22, 109, 413
COMMAND\hidenumberingonleftpage	22, 109, 419
COMMAND\hidenumberingonrightpage	22, 109, 419
COMMAND\hline	61
COMMAND\hrulefill	322
COMMAND\hsize	35, 164, 166–168, 175, 177, 212, 405, 411
COMMAND\hsizeX	355, 417, 418
COMMAND\hsizethreecol	353
COMMAND\hsizethreecolX	44, 46
COMMAND\hsizetwocol	353
COMMAND\hsizetwocolX	44, 46
COMMAND\hyperlinkR	286
COMMAND\hyperlinkformat	286
COMMAND\hyperlinkformatR	286
COMMAND\if@RTL	74
COMMAND\if@edtext@	411, 414
COMMAND\if@eled@sectioning	336
COMMAND\if@firstlineofpage	74
COMMAND\if@firstlineofpageR	74
COMMAND\if@msd@options@fullpage	306
COMMAND\if@msdata@insertedfrompreviouspage	303
COMMAND\if@nobreak	138
COMMAND\if@noneed@Footnote	116

COMMAND\ifXnote@	82
COMMAND\ifbypage@	88
COMMAND\ifbypage@R	88
COMMAND\ifbypstart@	88
COMMAND\ifbypstart@R	88
COMMAND\iffirst@linenum@out@	114
COMMAND\ifindtl@innote	82
COMMAND\ifindtl@notenumber	82
COMMAND\ifinserthangingsymbol	291
COMMAND\ifinstanza	291
COMMAND\ifstwofollowinglines	187, 188
COMMAND\ifl@d@Xmorethantwolines	185, 412
COMMAND\ifl@d@Xtwolines	185
COMMAND\ifl@d@dash	185
COMMAND\ifl@d@elin	184
COMMAND\ifl@d@esl	184
COMMAND\ifl@d@pnum	184
COMMAND\ifl@d@ssub	184
COMMAND\ifl@dend@X	235
COMMAND\ifl@dmemoir	404
COMMAND\ifl@dpadding	411
COMMAND\ifl@dpairing	81, 406
COMMAND\ifl@dprintingpages	412
COMMAND\ifl@dskipnumber	147
COMMAND\ifl@dstartendok	322
COMMAND\ifl@imakeidx	74
COMMAND\ifledRcol	82, 406
COMMAND\ifledRcol@	82, 410
COMMAND\iflemmacommand@	411
COMMAND\ifnoend@	221
COMMAND\ifnoledgroup@	281
COMMAND\ifnoteschanged@	98
COMMAND\ifnumberedpar@	135
COMMAND\ifnumbering	83, 85
COMMAND\ifnumberingR	82, 406
COMMAND\ifnumberline	126, 147
COMMAND\ifpst@rted	406
COMMAND\ifpst@rtedL	83
COMMAND\ifseriesbefore	239
COMMAND\ifstopmsdata@inserted@	299
COMMAND\ifsublines@	94, 107
COMMAND\iftrue	414
COMMAND\ifvmode	256
COMMAND\ifxxx	330
COMMAND\ignorespaces	124
COMMAND\imki@wrindexentry	74
COMMAND\immediate	114, 216
COMMAND\indent	18, 138, 354
COMMAND\index	289, 420
COMMAND\indtl@wrindexentry	74

COMMAND\initnumbering@quote	328, 415
COMMAND\initnumbering@reg	405
COMMAND\initnumbering@sectcmd	415
COMMAND\inplaceoflemmaseparator	353
COMMAND\inplaceofnumber	353
COMMAND\insert	126, 152, 159, 161, 162, 175, 190, 196
COMMAND\insert@Xtxtbeforenotes	156, 175
COMMAND\insert@count	111, 116, 124
COMMAND\insert@countR	124
COMMAND\insert@msdata	300, 305
COMMAND\insertthangingsymbol	409
COMMAND\insertlines@list	96, 111
COMMAND\insertparafootsepX	209
COMMAND\inserts@list	120, 135, 152, 167
COMMAND\interAfootnotelinepenalty	406
COMMAND\interfootnotelinepenalty	406
COMMAND\interlinepenalty	159
COMMAND\interparanoteglue	351
COMMAND\justifying	43
COMMAND\l@advance@parledegroupp@beforenormalnotes	415
COMMAND\l@d@wrindexhyp	411
COMMAND\l@d@add	128
COMMAND\l@d@end	216, 235
COMMAND\l@d@nums	124, 126, 128, 184, 185
COMMAND\l@d@section	217
COMMAND\l@d@set	106, 118
COMMAND\l@dampcount	313
COMMAND\l@dbfnote	193, 405
COMMAND\l@dcheckstartend	322
COMMAND\l@dchset@num	106
COMMAND\l@dcolcount	313, 314
COMMAND\l@dcollect@@body	307
COMMAND\l@dcollect@body	307
COMMAND\l@dcsnote	410
COMMAND\l@dcsnotetext	143, 272
COMMAND\l@dcsnotetext@l	143, 272
COMMAND\l@dcsnotetext@r	143, 272
COMMAND\l@ddodoreinextrafeet	250, 404
COMMAND\l@ddoxtrafeet	250, 404
COMMAND\l@demptyd@ta	406
COMMAND\l@dend@close	216
COMMAND\l@dend@open	216
COMMAND\l@dend@stuff	216
COMMAND\l@denvbody	307
COMMAND\l@dfeetbeginmini	404
COMMAND\l@dfeetendmini	404
COMMAND\l@dgetline@margin	406
COMMAND\l@dgetlock@disp	406
COMMAND\l@dgetref@num	261
COMMAND\l@dgetsidenote@margin	269, 406

COMMAND\l@dgobbeloptarg	411
COMMAND\l@dgobblearg	411
COMMAND\l@dgobbleoptarg	312
COMMAND\l@dlabel@parse	261
COMMAND\l@dld@ta	147, 149
COMMAND\l@dlp@rbox	274
COMMAND\l@dlsn@te	406
COMMAND\l@dlsnote	410
COMMAND\l@dmake@labels	256, 268
COMMAND\l@dmake@labelsR	268
COMMAND\l@dnumstartsL	83, 406
COMMAND\l@dp@rsefootspec	185
COMMAND\l@dp@rsefootspec	185
COMMAND\l@dpush@begins	307
COMMAND\l@drd@ta	147, 149
COMMAND\l@dref@undefined	260
COMMAND\l@drsn@te	406
COMMAND\l@drsnote	410
COMMAND\l@dtabaddcols	322
COMMAND\l@dtabnoexpands	404
COMMAND\l@dumboxmpfoot	415
COMMAND\l@dunboxmpfoot	406
COMMAND\l@dzeropenalties	406, 411
COMMAND\l@pb	342
COMMAND\l@prev@nopb	342
COMMAND\l@prev@pb	342
COMMAND\l@reg	349
COMMAND\label	18, 54, 56, 59, 60, 254, 255, 261
COMMAND\label@refs	254
COMMAND\labelstarttrue	18, 407
COMMAND\labelref@list	254, 257
COMMAND\language	168
COMMAND\last@page@num	405
COMMAND\lastbox	139
COMMAND\lastskip	117
COMMAND\latex@makecol	252
COMMAND\leavevmode	18, 139
COMMAND\led@check@nopb	342
COMMAND\led@check@pb	342
COMMAND\led@nopb	342, 343
COMMAND\led@nopbnum	342
COMMAND\led@pb	342, 343
COMMAND\led@pb@macro	342
COMMAND\led@pbnum	342
COMMAND\led@reinit@index@fornote	289
COMMAND\led@set@index@fornote	289
COMMAND\ledRflag	286
COMMAND\ledchapter	351, 408
COMMAND\ledfootinsdim	352
COMMAND\ledinnernote	56, 270, 410, 419

COMMAND\ledinnote	284, 350, 414
COMMAND\ledinnotemark	58, 350, 413
COMMAND\ledleftnote	57, 270
COMMAND\ledlinenum	93, 406
COMMAND\ledllfill	143
COMMAND\ledsnotefontsetup	419
COMMAND\ledsnotesep	57
COMMAND\ledsnotewidth	57
COMMAND\lednopb	65, 342
COMMAND\lednopbinverse	343
COMMAND\lednopbinversetrue	50, 65
COMMAND\lednopbnum	342
COMMAND\ledouternote	56, 270, 410, 419
COMMAND\ledpb	65, 342, 351
COMMAND\ledpbnum	342
COMMAND\ledpbsetting	66, 342, 416
COMMAND\ledrightnote	57, 270
COMMAND\ledrsnotefontsetup	419
COMMAND\ledrsnotesep	57
COMMAND\ledrsnotewidth	57
COMMAND\ledsection	351
COMMAND\ledsectnomark	330
COMMAND\ledsectnotoc	329
COMMAND\ledsetnormalparstuff	350, 351, 413
COMMAND\ledsetnormalparstuff@common	211
COMMAND\ledsetnormalparstuffX	350, 351, 413
COMMAND\ledsidenote	56, 57, 270, 272
COMMAND\ledsubsection	351
COMMAND\ledsubsubsection	351
COMMAND\ledxxx	410
COMMAND\left	62
COMMAND\leftctab	323
COMMAND\leftheadline	93
COMMAND\leftlinenum	20, 93, 404, 406
COMMAND\leftltab	323
COMMAND\leftnoteupfalse	57
COMMAND\leftpstartnum	151
COMMAND\leftftab	323
COMMAND\leftsidenote	272
COMMAND\leftskip	164, 167, 168
COMMAND\lemma	3, 24, 26–29, 119, 123, 124, 127, 129, 347, 406, 407, 414, 415, 417
COMMAND\lemmaseparator	353
COMMAND\let	27, 48, 122, 126, 235, 252, 296, 404
COMMAND\letboxXendendlinenum	353
COMMAND\line	173, 176
COMMAND\line@list	95, 112, 126
COMMAND\line@list@stuff	83, 99, 114, 404, 406
COMMAND\line@list@version	101
COMMAND\line@margin	89, 149, 269
COMMAND\line@num	94–96, 147, 404

COMMAND\line@set	128
COMMAND\lineation	20, 88
COMMAND\linebreak	35
COMMAND\linenum	24, 26, 27, 53, 54, 119, 128, 259, 261, 267, 347, 418
COMMAND\linenum@out	113, 114, 254, 257
COMMAND\linenumberlist	19, 20, 81, 148, 404
COMMAND\linenumberstyle	22, 93, 404, 421
COMMAND\linenumincrement	19, 406
COMMAND\linenummargin	20, 89, 269
COMMAND\linenumr@p	92, 404, 406
COMMAND\linenumrep	92, 93, 406
COMMAND\linenumsep	20, 57, 93, 270
COMMAND\linrangesep@	247
COMMAND\lineref	254, 259, 267, 353, 412
COMMAND\list@clear	87
COMMAND\list@clearing@reg	406
COMMAND\list@create	87
COMMAND\lock@disp	91
COMMAND\lock@off	108
COMMAND\lock@on	107
COMMAND\lockdisp	21, 91
COMMAND\loop	155, 156, 292
COMMAND\ltab	323, 324, 328
COMMAND\ltabtext	328
COMMAND\m@mmf@prepare	192
COMMAND\makeatletter	143
COMMAND\makehboxoffhboxes	169, 170
COMMAND\makeindex	58, 285
COMMAND\makelabel	267
COMMAND\managestanza@modulo	293
COMMAND\marginpar	56, 67, 268, 269, 405
COMMAND\marginparwidth	57, 270
COMMAND\markboth	143
COMMAND\mathchardef	293
COMMAND\maxhXnotes	353
COMMAND\maxhnotesX	46, 68, 352, 409, 410, 412–414
COMMAND\maxlinesinpar@list	99
COMMAND\measurebody	325
COMMAND\measuretbody	325
COMMAND\memorybreak	19
COMMAND\morenoexpands	68, 69, 120, 121
COMMAND\morethantwolines	353
COMMAND\mpfnpos	33, 213, 409, 420
COMMAND\mpnornalffootgroup	405
COMMAND\mpnornalvfootnote	405
COMMAND\msdata	31, 298–300, 421
COMMAND\multfootsep	32, 192
COMMAND\multiplefootnotemarker	192
COMMAND\musixtex	410
COMMAND\n@num	406, 413

COMMAND\n@num@ref	413
COMMAND\new@line	115, 405
COMMAND\new@series	122
COMMAND\newcommand	27, 70, 191, 256
COMMAND\newcommandx	27
COMMAND\newhookarg@specific	247
COMMAND\newhookcommand@series	241, 243, 413
COMMAND\newhookcommand@series@reload	243
COMMAND\newhookcommand@toggle@reload	242, 411
COMMAND\newhooktoggle@series	242, 413
COMMAND\newhooktoggle@specific	247
COMMAND\newif	413
COMMAND\newline	35
COMMAND\newlinechar	235
COMMAND\newseries	33, 348, 351, 352
COMMAND\newseries@	227, 240
COMMAND\newverse	50, 51, 351, 410
COMMAND\next	292
COMMAND\next@action	100
COMMAND\next@actionline	100
COMMAND\next@insert	153
COMMAND\nl@regR	102
COMMAND\no@expands	68, 122, 127, 235, 404
COMMAND\noXlemmaseparator	353
COMMAND\nobreak	184
COMMAND\nocritical	228
COMMAND\noeledsec	65, 351
COMMAND\noendnotes	351
COMMAND\noexpand	349
COMMAND\nofamiliar	244
COMMAND\noindent	18, 138, 416
COMMAND\noindentX	416
COMMAND\nomk@	247
COMMAND\nonbreakableafternumber	353
COMMAND\nonumberinfootnote	353
COMMAND\normal@footnotemarkX	196
COMMAND\normal@page@break	341
COMMAND\normal@pars	211
COMMAND\normalbfnoteX	406
COMMAND\normalbodyfootmarkX	196
COMMAND\normalfont	420
COMMAND\normalfootfmt	48, 163, 170, 180, 217, 418
COMMAND\normalfootfmtX	197
COMMAND\normalfootfootmarkX	197
COMMAND\normalfootgroup	165
COMMAND\normalfootgroupX	198
COMMAND\normalfootnoterule	160
COMMAND\normalfootstart	163, 167
COMMAND\normalfootstartX	197
COMMAND\normalvfootnote	161, 162

COMMAND\normalvfootnote@inserted	162
COMMAND\normalvfootnoteX	196
COMMAND\notbool	330
COMMAND\notfontsetup	352
COMMAND\notfontsizeX	42, 352
COMMAND\notenumfont	352
COMMAND\notenumfontX	41, 352
COMMAND\notesXwidthliketwocolumns	353
COMMAND\noteswidthliketwocolumnsX	47, 353, 411, 413
COMMAND\num@lines	135, 154
COMMAND\numberlinefalse	19
COMMAND\numberlinetrue	19
COMMAND\numberonlyfirstinline	239, 353
COMMAND\numberonlyfirstintwolines	353
COMMAND\numberpstartfalse	18
COMMAND\numberpstarttrue	18, 37, 350, 407, 415, 421
COMMAND\numberstanza	38
COMMAND\numberstanzafalse	50
COMMAND\numberstanzatrue	50
COMMAND\numlabfont	20, 47, 93
COMMAND\one@line	135
COMMAND\onehalfspacing	416
COMMAND\onlyXpstart	353
COMMAND\onlysideX	234
COMMAND\page@action	96, 106
COMMAND\page@start	96, 406
COMMAND\pagecontents	96
COMMAND\pagelinesep	59
COMMAND\pageparbreak	351
COMMAND\pageref	54, 259
COMMAND\par	26, 35, 138, 139, 211
COMMAND\par@line	135, 154
COMMAND\para@footgroup	167
COMMAND\para@footgroupX	209
COMMAND\para@footsetup	166, 404
COMMAND\para@footsetupX	206, 404, 411
COMMAND\para@vfootnoteX	207
COMMAND\parafootfmt	169, 170, 418
COMMAND\parafootfmtX	208
COMMAND\parafootftm	172
COMMAND\parafootftmX	209
COMMAND\parafootftmsep	352
COMMAND\parafootsep	353, 409, 414
COMMAND\parafootsepX	44, 98, 352, 353, 419
COMMAND\parafootstart	167
COMMAND\parafootstartX	207
COMMAND\paravfootnote	167, 170
COMMAND\parfillskip	170
COMMAND\parindent	416
COMMAND\parindentX	43, 355, 416, 418

COMMAND\parshape	67
COMMAND\parskip	139
COMMAND\pausenumbering	19, 85, 99, 101, 139, 409, 411, 418
COMMAND\penalty	170
COMMAND\pend	2, 6, 17–20, 22, 64, 65, 118, 120, 123, 128, 135–140, 151, 152, 349, 409, 410, 419, 420
COMMAND\preXnotes	413, 419
COMMAND\prenotesX	45, 215, 413
COMMAND\prepare@Xgroupbyline	190
COMMAND\prepare@Xprenotes	214
COMMAND\prev@nopb	342
COMMAND\prev@pb	342
COMMAND\prevlineX	98
COMMAND\prevpageX@num	98
COMMAND\print@Xfootnoterule	413
COMMAND\print@Xnotes	249, 250
COMMAND\print@Xnotes@forpages	412
COMMAND\print@eledsection	142
COMMAND\print@footnoteXrule	413
COMMAND\print@leftmargin@eledsection	331
COMMAND\print@lemma	181
COMMAND\print@line	140
COMMAND\print@notesX@forpages	412
COMMAND\print@rightmargin@eledsection	331
COMMAND\printendlines	221, 227, 265, 355, 404, 406
COMMAND\printlinefootnote	181, 183, 412
COMMAND\printlinefootnotearea	183, 184, 412
COMMAND\printlinefootnotenumbers	181
COMMAND\printlines	163, 180, 184, 185, 221, 265, 355, 404, 406, 412, 417
COMMAND\printnpnum	350, 355
COMMAND\printpstart	180
COMMAND\protect	121, 349
COMMAND\providecommand	191, 404
COMMAND\pstart	2, 6, 17–20, 22, 63–65, 106, 118, 123, 128, 135, 137–139, 142, 152, 336, 349, 406–408, 410, 411, 413–415, 418–420
COMMAND\pstartinfootnote	353
COMMAND\pstartinfootnoteeverytime	353
COMMAND\pstartnum	151
COMMAND\pstartref	52, 254, 260, 409
COMMAND\pstarts	407
COMMAND\raggedX	44
COMMAND\raggedleft	44
COMMAND\raggedright	43
COMMAND\raw@text	135
COMMAND\rbracket	40, 41
COMMAND\read@linelist	99–101
COMMAND\ref	54, 56, 59
COMMAND\refformated@	265
COMMAND\refformatedwithpage	265
COMMAND\relax	18, 106, 126, 145, 152, 296, 312, 349
COMMAND\renewcommand	68, 352, 355

COMMAND\reset@msd@options@iffullpage	306
COMMAND\resetprevline@	98
COMMAND\resetprevpage@	98
COMMAND\resumenumbering	19, 82, 85, 99, 101, 139, 406, 410, 411, 418
COMMAND\right	62
COMMAND\rightctab	323
COMMAND\rightlinenum	20, 93, 404, 406
COMMAND\rightltab	324
COMMAND\rightnoteupfalse	57
COMMAND\rightrtab	324
COMMAND\rightsidenote	272
COMMAND\rightskip	164, 167, 168, 170
COMMAND\rightstartnum	151
COMMAND\rigidbalance	172–174, 176, 355, 417
COMMAND\rigidbalanceX	173, 355, 417
COMMAND\robustify	35
COMMAND\roman	317, 417
COMMAND\rtab	323, 324, 328
COMMAND\rtabtext	325, 328
COMMAND\sameword	27–29, 113, 129–131, 133, 412, 414, 416, 419
COMMAND\sameword@inedtext	130
COMMAND\saweword	129
COMMAND\scriptsize	93
COMMAND\section	63, 406, 420
COMMAND\section@num	82
COMMAND\sectionmark	330
COMMAND\select@lemmafont	47, 48, 157
COMMAND\series	227
COMMAND\series@	227
COMMAND\seriesatbegin	33, 238, 413
COMMAND\seriesatend	33, 239, 413
COMMAND\set@Xtxtbeforenotes	156
COMMAND\set@line	126
COMMAND\set@line@action	97, 106
COMMAND\setSErefonlypageprefixmore	55, 265, 418
COMMAND\setSErefonlypageprefixsingle	55, 265, 418
COMMAND\setSErefprefixmore	55
COMMAND\setSErefprefixsingle	55
COMMAND\setapprefprefixmore	55, 352
COMMAND\setapprefprefixsingle	55, 352, 417
COMMAND\setcommand@series	241
COMMAND\sethangingsymbol	50, 291, 352, 416
COMMAND\sethanginsymbol	48
COMMAND\setistwofollowinglines	187
COMMAND\setl@dlprbox	274
COMMAND\setline	21, 22, 97, 102, 105, 118, 121, 137, 415
COMMAND\setlinenum	22, 102, 106, 118, 404
COMMAND\setmsdatalabel	32
COMMAND\setmsdataseries	32
COMMAND\setprintendlines	222, 223, 406

COMMAND\setprintlines	185, 187, 222, 406
COMMAND\setsidenotesep	57
COMMAND\setsidenotsep	352
COMMAND\setstanzaindent	293
COMMAND\setstanzaindents	49, 293, 349
COMMAND\setstanzapenalties	293
COMMAND\setstanzavalues	293
COMMAND\settoggle@series	240, 407, 411
COMMAND\showlemma	121, 404, 405
COMMAND\showwordrank	30, 130
COMMAND\sidenote@margin	405
COMMAND\sidenotemargin	56, 405, 410
COMMAND\sidenotesep	352
COMMAND\sidesstartnumtrue	18
COMMAND\skip	164
COMMAND\skipnumbering	22, 109, 118, 406, 413, 414
COMMAND\skipnumbering@reg	413
COMMAND\small	42
COMMAND\special	12
COMMAND\splitmaxdepth	159, 174
COMMAND\splitoff	173
COMMAND\splittopskip	159, 174, 176
COMMAND\stanza	21, 22, 50, 51, 296, 352, 354, 416
COMMAND\stanza@hang	295
COMMAND\stanza@line	295
COMMAND\stanzaindent	49, 293, 412
COMMAND\stanzaindent*	49
COMMAND\stanzaindentbase	292, 293
COMMAND\stanzanumwrapper	51
COMMAND\startlock	21, 97, 118, 296
COMMAND\startstanzahook	352
COMMAND\startsub	21, 97, 117
COMMAND\stopmsd	300
COMMAND\stopmsdata	31, 299
COMMAND\strip@pt	167
COMMAND\strutbox	174
COMMAND\sub@action	97, 107
COMMAND\sub@lock	95
COMMAND\sub@off	104, 257
COMMAND\sub@on	104, 257
COMMAND\subline@num	94–96
COMMAND\sublinenum@rep	404
COMMAND\sublinenumberstyle	22, 93, 404
COMMAND\sublinenumincrement	19
COMMAND\sublinenumr@p	92, 404, 406
COMMAND\sublinenumrep	92, 93, 406
COMMAND\sublineref	52, 254, 260
COMMAND\subsectionmark	330
COMMAND\sw@inthisedtext	123
COMMAND\sw@list@inedtext	127, 133

COMMAND\symlinenum	353
COMMAND\symplinenum	352
COMMAND\sza@penalty	295
COMMAND>tag	412
COMMAND\text	347
COMMAND\text< <i>language</i> >	42
COMMAND\textcolor	68
COMMAND\textheight	68
COMMAND\the	404
COMMAND\thefootnoteA	32
COMMAND\thefootnoteX	408
COMMAND\thelabidx	287, 288
COMMAND\thepage	102
COMMAND\thepstart	18
COMMAND\thepstartL	407
COMMAND\thepstartR	407
COMMAND\thestanza	50
COMMAND>this@line@list@version	114
COMMAND>thisfootnote	199
COMMAND\threecol@begin@insert	175
COMMAND\threecolfootfmt	175, 418
COMMAND\threecolfootfmtX	205
COMMAND\threecolfootgroup	174
COMMAND\threecolfootgroupX	205
COMMAND\threecolfootsetup	174
COMMAND\threecolfootsetupX	204
COMMAND\threecolvfootnote	174
COMMAND\threecolvfootnote@inserted	174
COMMAND\threecolvfootnoteX	204
COMMAND\toendnotes	25, 220, 419
COMMAND\twocolfootfmt	418
COMMAND\twocolfootfmtX	202
COMMAND\twocolfootgroupX	203
COMMAND\twocolfootsetupX	202
COMMAND\twocolvfootnoteX	202
COMMAND\twolines	240, 353
COMMAND\twolines@A	240
COMMAND\twolines@B	240
COMMAND\twolines@C	240
COMMAND\twolinesbutnotmore	353
COMMAND\twolinesonlyinsamepage	353
COMMAND\xtbeforeXnotes	353
COMMAND\unhbox	168
COMMAND\unpenalty	169, 170
COMMAND\unskip	170
COMMAND\unvxh	170, 354
COMMAND\unvxhX	354
COMMAND\upbracefill	322
COMMAND\usingcritext	348, 351
COMMAND\usingedtext	348, 351

COMMAND\VAfootnote	159
COMMAND\variant	27
COMMAND\vbox	137, 138, 168, 173, 213
COMMAND\vfootnote	159, 164, 167, 174
COMMAND\vl@dbfnote	193, 405
COMMAND\vl@disnote	273
COMMAND\vl@dlsnote	273
COMMAND\vl@dosnote	273
COMMAND\vl@drsnote	273
COMMAND\vnumfootnoteX	406
COMMAND\vsizer	46, 68
COMMAND\vsplit	154
COMMAND\waklam	322
COMMAND\waklamec	322
COMMAND\wapunktel	322
COMMAND\wastricht	322
COMMAND\widthX	46, 355, 418
COMMAND\wrap@edcrossref	259, 411
COMMAND\wrapcontentX	42, 419
COMMAND\wrapped@bodyfootmarkX	210
COMMAND\wrapped@footfootmarkX	210
COMMAND\x...	53
COMMAND\xdef	87, 296
COMMAND\xflagref	53, 260, 355, 417
COMMAND\xleft@appenditem	87, 120
COMMAND\xlineref	53, 355, 417
COMMAND\xpageref	53
COMMAND\xpstartref	53, 409
COMMAND\xr	56
COMMAND\xright@appenditem	87
COMMAND\xsublineref	53
COMMAND\xxref	53, 261, 267, 409, 412, 419
COMMAND\zz@@@	404
ENVIRONMENTastanza	417
ENVIRONMENTedarrayc	328
ENVIRONMENTedarrayl	328
ENVIRONMENTedarrayr	328
ENVIRONMENTedtabularc	328
ENVIRONMENTedtabularl	328
ENVIRONMENTedtabularr	328
ENVIRONMENTledgroup	73, 279, 355, 417
ENVIRONMENTledgroupsize	280
PACKAGE(r)(e)ledmac	33
PACKAGEEledmac	11, 70, 96, 285, 350, 351, 412, 413
PACKAGEEledpar	413
PACKAGEEtoolbox	73
PACKAGEParallel	357
PACKAGEReledmac	354, 355
PACKAGEamsgen	306
PACKAGEamsmath	306

PACKAGEbabel	42, 69, 317, 417
PACKAGEbiblatex	67
PACKAGEbidi	42, 43, 74, 416
PACKAGEccaption	81
PACKAGEcolor	68
PACKAGEedmac	1, 6, 10, 12, 13, 70, 184, 191, 254, 293, 347, 357, 404
PACKAGEedstanza	1, 13, 291
PACKAGEeledmac	1, 10, 13–16, 59, 191, 281, 285, 310, 332, 344, 348, 350–352, 408, 410, 412
PACKAGEeledpar	81, 159, 330, 357, 406, 410–412
PACKAGEetex	416
PACKAGEetoolbox	86, 129, 227, 240, 248, 272, 330, 341, 342
PACKAGEfancyhdr	252, 420
PACKAGEfloatrow	66, 355
PACKAGEfootmisc	32, 69, 74, 191, 357, 420
PACKAGEgeometry	14
PACKAGEglossaries	60, 289, 417
PACKAGEhandout	411
PACKAGEhyperlink	234
PACKAGEhyperref	53, 123, 210, 255, 287, 334, 344, 409–411, 418
PACKAGEifluatex	73
PACKAGEifxetex	73
PACKAGEimakeidx	58, 67, 74, 281, 285, 351, 408–410, 412
PACKAGEindextols	289
PACKAGEindextool	351
PACKAGEindextools	58, 67, 74, 82, 281, 285, 289, 351, 412, 417, 420
PACKAGEinputenc	131
PACKAGEledarab	69
PACKAGEledmac	1, 10, 13, 14, 69, 86, 285, 347, 348, 351, 354
PACKAGEledpar	69
PACKAGEMemoir	73, 285, 351, 357, 411
PACKAGEMorewrites	67
PACKAGEMusixtex	410
PACKAGEperpage	417
PACKAGEpolyglossia	40, 69, 124, 158, 180, 417
PACKAGERagged2e	44, 73
PACKAGEREledmac	1, 2, 10–12, 14–16, 18, 19, 22–24, 26–28, 30, 32, 33, 36, 37, 39, 42, 43, 45–48, 50, 51, 53, 54, 56, 58–60, 65–71, 88, 90, 96, 97, 100, 101, 104, 113, 114, 121, 126, 129, 152, 160, 164, 168, 180, 191, 217, 228, 232, 240, 248, 259, 262, 267, 268, 285, 303, 310, 330, 331, 343, 344, 351, 352, 354, 355, 415, 418, 420
PACKAGEREledpar	1, 4, 6, 8, 15, 18, 47, 53, 55, 65–67, 69, 71, 81, 88, 99, 104, 124, 126, 161, 164, 211, 212, 227, 233, 247, 249, 250, 281, 291, 416, 417, 419, 420
PACKAGESuffix	73
PACKAGETabmac	1, 13, 357
PACKAGEuninormalize	28
PACKAGExargs	27, 73
PACKAGExkeyval	71, 247
PACKAGEXr	5, 56, 268, 418
PACKAGEXref	267
PACKAGEXstring	73, 287

A

<code>\absline@num</code>	1
Abu Kamil Shuja' b. Aslam	13
<code>\actionlines@list</code>	1
<code>\actions@list</code>	1
<code>\add@inserts</code>	1
<code>\add@inserts@next</code>	1
<code>\add@msd@</code>	1
<code>\add@msdata</code>	1
<code>\add@msdata@firstlineofpage</code>	1
<code>\add@penalties</code>	1
<code>\add@Xgroupbyline</code>	1
<code>\addtol@denbody</code>	1
Adelard II	13
<code>\advancelabel@refs</code>	1
<code>\advanceline</code>	1, 21
<code>\Aendnote</code>	24
<code>\affixline@num</code>	1
<code>\affixpstart@num</code>	1
<code>\affixside@note</code>	1
<code>\Afootnote</code>	24
<code>\afternoteX</code>	44
<code>\afterruleX</code>	46
<code>\ampersand</code>	1, 51
<code>\append@notesX</code>	1
<code>\append@Xnotes</code>	1
<code>\applabel</code>	1, 54
<code>\appref</code>	1, 54
<code>\apprefwithpage</code>	1, 54
<code>\arrangementX</code>	1, 34
<code>\arrangementX@normal</code>	1
<code>\arrangementX@threecol</code>	1
<code>\arrangementX@twocol</code>	1
<code>\at@every@pend</code>	1
<code>\AtEveryPend</code>	1, 18
<code>\AtEveryPstart</code>	1, 18
<code>\AtEveryStanza</code>	1
<code>\AtEveryStopStanza</code>	1
<code>\autopar</code>	1, 17

B

<code>\ballast</code>	67
<code>\ballast@count</code>	1
Beeton, Barbara Ann Neuhaus Friend	18
<code>\beforeeledchapter</code>	1
<code>\beforeinsertingX</code>	43
<code>\beforenotesX</code>	45
<code>\beginnumbering</code>	1, 16
<code>\Bendnote</code>	24
<code>\Bfootnote</code>	24

\bhookgroupX	45
\bhooknoteX	43
\bodyfootmarkA	32
\boxfootnotenumbers	1
Bredon, Simon	13
Breger, Herbert	13, 309
Brey, Gerhard	13
Busard, Hubert L. L.	13
\bypage@false	1
\bypage@true	1
\bypstart@false	1
\bypstart@true	1

C

\c@addcolcount	1
\c@ballast	1
\c@firstlinenum	1
\c@firstsublinenum	1
\c@labidx	1
\c@linenumincrement	1
\c@sublinenumincrement	1
\Cendnote	24
\Cfootnote	24
\ch@ck@l@ck	1
\ch@cksub@l@ck	1
\chapter	1
\check@pb@in@verse	1
Chester, Robert of	13
Claassens, Geert H. M.	13
\colalignX	43
Copernicus, Nicolaus	13
\critext	347
\ctab	1
\ctabtext	1

D

Dekker, Dirk-Jan	68
\Dendnote	24
\Dfootnote	24
\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

<code>\do@line</code>	1
<code>\do@linehook</code>	1
<code>\do@lockoff</code>	1
<code>\do@lockoffL</code>	1
<code>\do@lockon</code>	1
<code>\do@lockonL</code>	1
<code>\do@Xfeet</code>	1
<code>\doedindexlabel</code>	1
<code>\doendnotes</code>	1, 25
<code>\doendnotesbysection</code>	1, 25
<code>\doinsidelinehook</code>	1, 22
<code>\dolinehook</code>	1, 22
<code>\dosplits</code>	1
Downes, Michael	67, 168, 170
<code>\doxtrafeet</code>	1
<code>\dummy@edtext</code>	1
<code>\dummy@edtext@showlemma</code>	1
<code>\dummy@ref</code>	1

E

<code>\edaftertab</code>	1, 63, 323
<code>edarrayc</code> (environment)	60
<code>edarrayl</code> (environment)	60
<code>edarrayr</code> (environment)	60
<code>\edatleft</code>	1, 62
<code>\edatright</code>	1, 62
<code>\edbeforetab</code>	1, 63, 323
<code>\edfilldimen</code>	1
<code>\edfont@info</code>	1
<code>\edindex</code>	1, 57
<code>\edindexlab</code>	1, 59
<code>\EDLABEL</code>	1
<code>\edlabel</code>	1, 52
<code>\edlabelE</code>	1, 54
<code>\edlabelS</code>	1, 54
<code>\edlabelSE</code>	1, 54
<code>\edlineref</code>	1, 52
<code>\edmakelabel</code>	1, 53
<code>\edpageref</code>	1, 52
<code>\edrowfill</code>	1, 61
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 61
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc</code> (environment)	60
<code>edtabularl</code> (environment)	60
<code>edtabularr</code> (environment)	60
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 23

<code>\edvertdots</code>	1, 63
<code>\edvertline</code>	1, 63
<code>\Eendnote</code>	24
<code>\Efootnote</code>	24
<code>\eled@chapter</code>	1
<code>\eled@section</code>	1
<code>\eled@sectioning@out</code>	1
<code>\eled@subsection</code>	1
<code>\eled@subsubsection</code>	1
<code>\eledchapter</code>	1
<code>\eledchapter*</code>	1
<code>\eledsection</code>	1
<code>\eledsection*</code>	1
<code>\eledsubsection</code>	1
<code>\eledsubsection*</code>	1
<code>\eledsubsubsection</code>	1
<code>\eledsubsubsection*</code>	1
<code>\enablel@dtabfeet</code>	1
<code>\end@lemmas</code>	1
<code>\endashchar</code>	1
<code>\endline@num</code>	1
<code>\endlock</code>	1, 21
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 16
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 21
<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	60
<code>edarrayl</code>	60
<code>edarrayr</code>	60
<code>edtabularc</code>	60
<code>edtabularl</code>	60
<code>edtabularr</code>	60
<code>ledgroup</code>	51
<code>ledgroupsize</code>	51
<code>minipage</code>	51
Euclid	13
<code>\extensionchars</code>	1, 66

F

<code>\f@x@l@cks</code>	1
Fairbairns, Robin	32
<code>\first@linenum@out@false</code>	1
<code>\first@linenum@out@true</code>	1
<code>\firstlinenum</code>	1, 19
<code>\firstseriesX@</code>	1

\firstsublinenum	1, 19
\firstXseries@	1
\fix@page	1
\flag@end	1
\flag@end@later	1
\flag@start	1
\flag@start@later	1
\flagstanza	1, 51
\flush@notes	1
\fnpos	1, 33
Folkerts, Menso	13
\footfootmarkA	32
\footfudgefiddle	1, 68
\footnote	1
\footnoteA	32
\footnoteB	32
\footnoteC	32
\footnoteD	32
\footnoteE	32
\footnotelang@lua	1
\footnotelang@poly	1
\footnoteoptions@	1
\footsplitskips	1
\fullstop	1

G

Gädeke, Nora	13
\get@edindex@hyperref	1
\get@edindex@ledinnote@command	1
\get@fnmark	1
\get@fnmarkX	1
\get@index@command	1
\get@linelistfile	1
\get@sw@txt	1
\get@thisfootnote	1
\get@thisfootnoteX	1
\getline@num	1
\gl@p	1

H

\h@num	1
\hangindentX	43
\hidenumbering	1, 22
\hidenumberingonleftpage	1, 22
\hidenumberingonrightpage	1
\Hilfsbox	1
\hilfsbox	1
\hilfscount	1
\HILFSskip	1
\Hilfsskip	1

<code>\hilfsskip</code>	1
<code>\hsizethreecolX</code>	44
<code>\hsizetwocolX</code>	44
<code>\Hy@raisedlink@left</code>	1
<code>\hyperlinkformat</code>	1
<code>\hyperlinkformatR</code>	1
<code>\hyperlinkR</code>	1

I

<code>\if@addsw</code>	1
<code>\if@edtext@secondarg@</code>	1
<code>\if@eled@sectioning</code>	1
<code>\if@firstlineofpage</code>	1
<code>\if@led@nofoot</code>	1
<code>\if@ledgroup</code>	1
<code>\if@lemmacommand@</code>	1
<code>\if@msdata@insertedfrompreviouspage</code>	1
<code>\if@noeled@sec</code>	1
<code>\if@noneed@Footnote</code>	1
<code>\if@RTL</code>	1
<code>\ifautopar@pause</code>	1
<code>\ifbypage@</code>	1
<code>\ifbypage@R</code>	1
<code>\ifbypstart@</code>	1
<code>\ifbypstart@R</code>	1
<code>\ifeledmaccompat@</code>	1
<code>\iffirst@linenum@out@</code>	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, 66
<code>\ifledgroupnotesL@</code>	1
<code>\ifledgroupnotesR@</code>	1
<code>\iflednopbinverse</code>	1
<code>\ifledRcol</code>	1
<code>\ifledRcol@</code>	1
<code>\ifnocritical@</code>	1
<code>\ifnoend@</code>	1
<code>\ifnofamiliar@</code>	1
<code>\ifnoledgroup@</code>	1
<code>\ifnoquotation@</code>	1
<code>\ifnoteschanged@</code>	1
<code>\ifnumberedpar@</code>	1
<code>\ifnumbering</code>	1
<code>\ifnumberingR</code>	1
<code>\ifnumberline</code>	1
<code>\ifnumberstanza</code>	1
<code>\ifparapparatus@</code>	1
<code>\ifparledgroup</code>	1
<code>\ifpst@rtedL</code>	1
<code>\ifseriesbefore</code>	1
<code>\ifsidepstartnum</code>	1
<code>\ifstopmsdata@inserted@</code>	1
<code>\ifsublines@</code>	1
<code>\ifwidthliketwocolumns</code>	1
<code>\ifXendinsertsep@</code>	1
<code>\ifxindy@</code>	1
<code>\ifxindyhyperref@</code>	1
<code>\initnumbering@quote</code>	1
<code>\initnumbering@reg</code>	1
<code>\insert@count</code>	0, 1
<code>\insert@msdata</code>	1
<code>\insert@Xtxtbeforenotes</code>	1
<code>\inserthangingymbol</code>	1
<code>\insertlines@list</code>	1
<code>\insertparafootsepX</code>	1
<code>\inserts@list</code>	1

J

Jayaditya	13
-----------------	----

K

Kabelschacht, Alois	156
---------------------------	-----

L

<code>\l@advance@parledgroup@beforenormalnotes</code>	1
<code>\l@d@add</code>	1
<code>\l@d@nums</code>	1
<code>\l@d@section</code>	1

<code>\l@d@set</code>	1
<code>\l@d@Xend</code>	1
<code>\l@dampcount</code>	1
<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@ddodoreinxtrafeet</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@denvbody</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@dnumpstartsL</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

\l@dparsedstartline	1
\l@dparsedstartpage	1
\l@dparsedstartsub	1
\l@dparsefootspec	1
\l@dpush@begins	1
\l@drd@ta	1
\l@dref@undefined	1
\l@drestorefills	1
\l@drestoreforedtext	1
\l@drp@rbox	1
\l@drsn@te	1
\l@drsnote	1
\l@dsetmaxcolwidth	1
\l@dskipnumberfalse	1
\l@dskipnumbertrue	1
\l@dtabaddcols	1
\l@dtabnoexpands	1
\l@dunboxmpfoot	1
\l@dunhbox@line	1
\l@dzeropenalties	1
\label	54
\labelpstartfalse	1
\labelpstarttrue	1, 18
\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

<code>\led@err@toendnotes@outsidenumbering</code>	1
<code>\led@err@TooManyColumns</code>	1
<code>\led@err@UnequalColumns</code>	1
<code>\led@error@fail@patch@@doclearpage</code>	1
<code>\led@error@fail@patch@@iiiminipage</code>	1
<code>\led@error@fail@patch@@makecol</code>	1
<code>\led@error@fail@patch@@reinserts</code>	1
<code>\led@error@fail@patch@endminipage</code>	1
<code>\led@error@PackageAfterEledmac</code>	1
<code>\led@mess@NotesChanged</code>	1
<code>\led@mess@SectionContinued</code>	1
<code>\led@nopb</code>	1
<code>\led@nopbnum</code>	1
<code>\led@pb</code>	1
<code>\led@pb@setting</code>	1
<code>\led@pbnum</code>	1
<code>\led@reinit@index@fornote</code>	1
<code>\led@set@index@fornote</code>	1
<code>\led@toksa</code>	1
<code>\led@toksb</code>	1
<code>\led@warn@AppLabelOutSecondArgEdtext</code>	1
<code>\led@warn@BadAction</code>	1
<code>\led@warn@BadAdvancelineLine</code>	1
<code>\led@warn@BadAdvancelineSubline</code>	1
<code>\led@warn@BadLineation</code>	1
<code>\led@warn@BadLinenummargin</code>	1
<code>\led@warn@BadLockdisp</code>	1
<code>\led@warn@BadSetline</code>	1
<code>\led@warn@BadSetlinenum</code>	1
<code>\led@warn@BadSidenotemargin</code>	1
<code>\led@warn@BadSublockdisp</code>	1
<code>\led@warn@DuplicateLabel</code>	1
<code>\led@warn@edinde@outsidenumbering</code>	1
<code>\led@warn@LineFileObsolete</code>	1
<code>\led@warn@NoIndexFile</code>	1
<code>\led@warn@NoLineFile</code>	1
<code>\led@warn@NoMarginpars</code>	1
<code>\led@warn@RefUndefined</code>	1
<code>\led@warn@SeriesStillExist</code>	1
<code>\led@warning@hsizeX@deprecated</code>	1
<code>\led@warning@msdatawithoutstop</code>	1
<code>\led@warning@preXnotes@deprecated</code>	1
<code>\led@warning@Xhsize@deprecated</code>	1
<code>ledgroup (environment)</code>	51
<code>ledgroupsize (environment)</code>	51
<code>\ledinnernote</code>	1, 56
<code>\ledinnote</code>	1
<code>\ledinnotehyperpage</code>	1
<code>\ledinnotemark</code>	1
<code>\ledleftnote</code>	1, 56

<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 57
<code>\ledlsnotesep</code>	1, 57
<code>\ledlsnotewidth</code>	1, 57
<code>\lednopb</code>	1, 65
<code>\lednopbinversetrue</code>	65
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	56
<code>\ledouterote</code>	1
<code>\ledpb</code>	1, 65
<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 65
<code>\ledrightnote</code>	1, 56
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 57
<code>\ledrsnotesep</code>	1, 57
<code>\ledrsnotewidth</code>	1, 57
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 56
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 20
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	57
<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
<code>Leibniz</code>	13
<code>\lemma</code>	1, 26
<code>\letsforverteilen</code>	1
<code>\line@list</code>	1
<code>\line@list@stuff</code>	1
<code>\line@list@version</code>	1
<code>\line@margin</code>	1
<code>\line@num</code>	1
<code>\line@set</code>	1
<code>\lineation</code>	1, 20
<code>\linenum</code>	1, 26
<code>\linenum@out</code>	1
<code>\linenumberlist</code>	1, 19
<code>\linenumberstyle</code>	1, 22
<code>\linenumincrement</code>	1, 19
<code>\linenummargin</code>	1, 20
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 20
<code>\linerangesep@</code>	1
<code>\list@clear</code>	1

<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 21
Lorch, Richard	13
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	72

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>	46
Mayer, Gyula	13
<code>\measurebody</code>	1
<code>\measurecell</code>	1
<code>\measuremrow</code>	1
<code>\measuretbody</code>	1
<code>\measuretcell</code>	1
<code>\measuretrow</code>	1
Middleton, Thomas	13, 95
<code>minipage</code> (environment)	51
Mittelbach, Frank	12
<code>\morenoexpands</code>	1, 68
<code>\mp@append@notesX</code>	1
<code>\mp@append@Xnotes</code>	1
<code>\mpfnpos</code>	1, 33
<code>\mpnormalfootgroup</code>	1
<code>\mpnormalfootgroupX</code>	1
<code>\mpnormalvfootnote</code>	1
<code>\mpnormalvfootnote@inserted</code>	1
<code>\mpnormalvfootnoteX</code>	1
<code>\mppara@footgroupX</code>	1
<code>\mppara@vfootnoteX</code>	1
<code>\mpparafootgroup</code>	1
<code>\mpparavfootnote</code>	1
<code>\mpthreecolfootgroup</code>	1
<code>\mpthreecolfootgroupX</code>	1
<code>\mpthreecolfootsetup</code>	1
<code>\mpthreecolfootsetupX</code>	1
<code>\mptwocolfootgroup</code>	1
<code>\mptwocolfootgroupX</code>	1
<code>\mptwocolfootsetup</code>	1
<code>\mptwocolfootsetupX</code>	1
<code>\msdata</code>	1, 31

\multfootsep	1, 32
\multiplefootnotemarker	1

N

\n@num	1
\n@num@stanza	1
\new@line	1
\newhookarg@specific	1
\newhookcommand@series	1
\newhookcommand@series@reload	1
\newhooktoggle@series	1
\newhooktoggle@series@reload	1
\newhooktoggle@specific	1
\newseries@	1
\newverse	1
\NEXT	1
\no@expands	1
\noeledsec	65
\nomk@	1
\normal@footnotemarkX	1
\normal@page@break	1
\normal@pars	1
\normalbfnoteX	1
\normalbodyfootmarkX	1
\normalfootfmt	1
\normalfootfmtX	1
\normalfootfootmarkX	1
\normalfootgroup	1
\normalfootgroupX	1
\normalfootnoterule	1
\normalfootnoteruleX	1
\normalfootstart	1
\normalfootstartX	1
\normalvfootnote	1
\normalvfootnote@inserted	1
\normalvfootnoteX	1
\notefontsizeX	42
\notenumfontX	41
\noteschanged@false	1
\noteschanged@true	1
\noteswidthliketwocolumnsX	47
\nulledindex	1
\nullsetzen	1
\num@lines	1
\numberedpar@false	1
\numberedpar@true	1
\numberingfalse	1
\numberingtrue	1
\numberlinefalse	19
\numberlinetrue	19

<code>\numberpstartfalse</code>	1, 18
<code>\numberpstarttrue</code>	1, 18
<code>\numberstanzafalse</code>	50
<code>\numberstanzatru</code>	50
<code>\numlabfont</code>	1, 47

O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>optionauxdir</code>	15, 419
<code>optioncontinuousnumberingwithcolumns</code>	418
<code>optioninnote</code>	417
<code>optioninnote</code>	417
<code>optionlinrangesep</code>	247
<code>optionnocritical</code>	417
<code>optionnoeledsec</code>	336, 419
<code>optionnoend</code>	417
<code>optionnopenalties</code>	67
<code>optionnotenumber</code>	417

P

<code>\page@action</code>	1
<code>\page@num</code>	1
<code>\pagelinesep</code>	1, 58
<code>\pageref</code>	54
<code>\par@line</code>	1
<code>\para@footgroupX</code>	1
<code>\para@footsetup</code>	1
<code>\para@footsetupX</code>	1
<code>\para@vfootnoteX</code>	1
<code>\parafootfmt</code>	1
<code>\parafootfmtX</code>	1
<code>\parafootgroup</code>	1
<code>\parafootsepX</code>	44
<code>\parafootstart</code>	1
<code>\parafootstartX</code>	1
<code>\paravfootnote</code>	1
<code>\parindentX</code>	42
<code>\pausenumbering</code>	1, 18
<code>\pend</code>	1, 16
Plato of Tivoli	13
<code>\postbodyfootmark</code>	1
<code>\prebodyfootmark</code>	1
<code>\prenotesX</code>	45
<code>\prepare@edindex@fornote</code>	1
<code>\prepare@prenotesX</code>	1
<code>\prepare@Xgroupbyline</code>	1
<code>\prepare@Xprenotes</code>	1
<code>\prev@nopb</code>	1
<code>\prev@pb</code>	1

<code>\prevpage@num</code>	1
<code>\print@eledsection</code>	1
<code>\print@footnoteXrule</code>	1
<code>\print@leftmargin@eledsection</code>	1
<code>\print@lemma</code>	1
<code>\print@line</code>	1
<code>\print@notesX</code>	1
<code>\print@rightmargin@eledsection</code>	1
<code>\print@Xfootnoterule</code>	1
<code>\print@Xnotes</code>	1
<code>\printendlines</code>	1
<code>\printlineendnote</code>	1
<code>\printlineendnotearea</code>	1
<code>\printlinefootnote</code>	1
<code>\printlinefootnotearea</code>	1
<code>\printlinefootnotenumbers</code>	1
<code>\printlines</code>	1
<code>\printnpnum</code>	1
<code>\printpstart</code>	1
<code>\printsymlineendnotearea</code>	1
<code>\printsymlinefootnotearea</code>	1
<code>\printXafternumber</code>	1
<code>\printXbeforenumber</code>	1
<code>\pstart</code>	1, 16
<code>\pstarteref</code>	1
<code>\pstartnum</code>	1
<code>\pstartref</code>	52

Q

<code>\quotation</code>	1
<code>\quote</code>	1

R

<code>\raggedX</code>	44
<code>\raw@text</code>	1
<code>\rbracket</code>	1
<code>\read@linelist</code>	1
<code>\ref</code>	54
<code>\ref@reg@later</code>	1
<code>\Relax</code>	1
<code>\reledmac@error</code>	1
<code>\reledmac@warning</code>	1
<code>\removehboxes</code>	1
<code>\reset@msd@options@iffullpage</code>	1
<code>\resetprevline@</code>	1, 98
<code>\resetprevpage@</code>	1
<code>\resetprevpage@num</code>	98
<code>\restore@familiarnotes</code>	1
<code>\restore@notes</code>	1
<code>\restore@sidenotes</code>	1

\resumenumbering	1, 18
\rightctab	1
\rightlinenum	1, 20
\rightltab	1
\rightnoteupfalse	57
\rightrtab	1
\rightstartnum	1
\rigidbalance	1
\rigidbalanceX	1
\rtab	1
\rtabtext	1

S

Sacrobosco	13
\sameword	1, 27
\sameword@inedtext	1
Schöpf, Rainer	12
\section@num	1
\select@lemmafont	1
\select@lemmafont	1, 47
\Seref	1, 54
\Serefonlypage	54
\Serefwithpage	1, 54
\series	1
\seriesatbegin	1, 33
\seriesatend	1, 33
\set@line	1
\set@line@action	1
\set@Xtxtbeforenotes	1
\setapprefprefixmore	54
\setapprefprefixsingle	54
\setcommand@series	1
\sethangingsymbol	1, 50
\setistwofollowinglines	1
\setl@dlp@rbox	1
\setl@drpr@box	1
\setline	1, 21
\setlinenum	1, 22
\setmcellcenter	1
\setmcellleft	1
\setmcellright	1
\setmrowcenter	1
\setmrowleft	1
\setmrowright	1
\setmsdatalabel	1, 32
\setmsdataseries	1, 32
\setnoteswidthliketwocolumnsX@	1
\setnotesXpositionliketwocolumns@	1
\setprintendlines	1
\setprintlines	1

<code>\setSerefonlypageprefixmore</code>	55
<code>\setSerefonlypageprefixsingle</code>	55
<code>\setSerefprefixmore</code>	55
<code>\setSerefprefixsingle</code>	55
<code>\setsidenotesep</code>	57
<code>\setstanzaindents</code>	1, 48
<code>\setstanzapenalties</code>	1, 49
<code>\setstanzavalues</code>	1
<code>\settccllcenter</code>	1
<code>\settccllleft</code>	1
<code>\settccllright</code>	1
<code>\settoggle@series</code>	1
<code>\setthrowcenter</code>	1
<code>\setthrowleft</code>	1
<code>\setthrowright</code>	1
<code>\setXnotespositionliketwocolumns@</code>	1
<code>\setXnoteswidthliketwocolumns@</code>	1
<code>\showlemma</code>	1, 66
<code>\showwordrank</code>	1, 30
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 56
<code>\sidepstartnumtrue</code>	18
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 22
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 61
<code>\spreadtext</code>	1, 61
<code>\stanza</code>	1, 48
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 49
<code>\stanzaindent*</code>	1, 49
<code>\stanzaindentbase</code>	1, 48
<code>\stanzanumwrapper</code>	1, 51
<code>\startlock</code>	1, 21
<code>\startsub</code>	1, 21
<code>\stepl@dcclcount</code>	1
<code>\stopmsdata</code>	1, 31
<code>\strip@szacnt</code>	1
<code>\sub@action</code>	1
<code>\sub@lock</code>	1
<code>\sub@off</code>	1
<code>\sub@on</code>	1
<code>\subline@num</code>	1
<code>\sublinenumberstyle</code>	1, 22
<code>\sublinenumincrement</code>	1, 19
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 52

<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\subblock@disp</code>	1
<code>\subblockdisp</code>	1
Sullivan, Wayne	12, 13, 67, 81, 85, 168, 169, 254, 291
<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>\thefootnoteA</code>	32
Theodosius	13
<code>\thepageline</code>	1
<code>\thepstart</code>	1, 18
<code>\thestanza</code>	1, 50
<code>\thestartpageline</code>	1
<code>\this@line@list@version</code>	1
<code>\threecol@begin@insert</code>	1
<code>\threecolfootfmt</code>	1
<code>\threecolfootfmtX</code>	1
<code>\threecolfootgroup</code>	1
<code>\threecolfootgroupX</code>	1
<code>\threecolfootsetup</code>	1
<code>\threecolfootsetupX</code>	1
<code>\threecolvfootnote</code>	1
<code>\threecolvfootnote@inserted</code>	1
<code>\threecolvfootnoteX</code>	1
<code>\toendnotes</code>	25
<code>\toendnotes*</code>	1
<code>\twocolfootfmt</code>	1
<code>\twocolfootfmtX</code>	1
<code>\twocolfootgroup</code>	1
<code>\twocolfootgroupX</code>	1
<code>\twocolfootsetup</code>	1
<code>\twocolfootsetupX</code>	1
<code>\twocolvfootnote</code>	1
<code>\twocolvfootnote@inserted</code>	1
<code>\twocolvfootnoteX</code>	1

U

<code>\unvxhX</code>	1
----------------------------	---

V

Vamana	13
<code>\variab</code>	1
<code>\vbfnoteX</code>	1
<code>\vl@dbfnote</code>	1

<code>\vl@dcsnote</code>	1
<code>\vl@disnote</code>	1
<code>\vl@dlsnote</code>	1
<code>\vl@dosnote</code>	1
<code>\vl@drsnote</code>	1
<code>\vnumfootnoteX</code>	1

W

Whitney, Ron	12
<code>\widthX</code>	46
<code>\wrap@edcrossref</code>	1
<code>\wrapcontentX</code>	42
<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>\Xafterlemmaseparator</code>	41
<code>\Xafternote</code>	44
<code>\Xafternumber</code>	38
<code>\Xafterrule</code>	46
<code>\Xaftersymlinenum</code>	39
<code>\Xarrangement</code>	1, 34
<code>\Xarrangement@normal</code>	1
<code>\Xarrangement@paragraph</code>	1
<code>\Xarrangement@threecol</code>	1
<code>\Xarrangement@twocol</code>	1
<code>\Xbeforeinserting</code>	43
<code>\Xbeforelemmaseparator</code>	40
<code>\Xbeforenotes</code>	45
<code>\Xbeforenumber</code>	36, 38
<code>\Xbeforesymlinenum</code>	39
<code>\Xbhookgroup</code>	45
<code>\Xbhooknote</code>	43
<code>\Xboxlinenum</code>	39
<code>\Xboxlinenumalign</code>	39
<code>\Xboxsymlinenum</code>	39
<code>\Xcolalign</code>	43
<code>\xedindex</code>	1
<code>\xedlabel</code>	1
<code>\xedtext</code>	1
<code>\Xendafterrenumber</code>	38
<code>\Xendafterlemmaseparator</code>	41
<code>\Xendafternote</code>	47
<code>\Xendafterpagenumber</code>	40
<code>\Xendaftersymlinenum</code>	39
<code>\Xendahookinplaceofnumber</code>	40

\Xendahooklinenumber	40
\Xendbeforelemmaseparator	41
\Xendbeforenumber	38
\Xendbeforepagenumber	40
\Xendbeforesymlinenum	39
\Xendbhookinplaceofnumber	40
\Xendbhooklinenumber	40
\Xendbhooknote	43
\Xendboxendlinenumalign	40
\Xendboxlinenum	40
\Xendboxlinenumalign	40
\Xendboxstartlinenumalign	40
\Xendboxsymlinenum	39
\Xendhangindent	43
\Xendinplaceoflemmaseparator	41
\Xendinplaceofnumber	39
\Xendinplaceofpagenumber	36
\Xendlemmadisablefontselection	42
\Xendlemmafont	42
\Xendlemmaseparator	41
\Xendlineprefixmore	40
\Xendlineprefixsingle	40
\Xendlinerrangeseparator	36
\Xendmorethantwolines	37
\Xendnonumber	37
\Xendnotefontsize	42
\Xendnotenumfont	41
\Xendnumberonlyfirstinline	35
\Xendnumberonlyfirstintwolines	35
\Xendpagenumberonlyfirst	36
\Xendparagraph	47
\Xendsep	47
\Xendsublinesep	38
\Xendsymlinenum	35
\Xendsympagenum	36
\Xendtwolines	37
\Xendtwolinesbutnotmore	37
\Xendwrapcontent	42
\xflagref	1
\Xgroupbyline	45
\Xgroupbylineseparatetwolines	45
\Xhangindent	43
\Xhsizethreecol	44
\Xhsizetwocol	44
\Xinplaceoflemmaseparator	41
\Xinplaceofnumber	39
\Xinsertparafootsep	1
\Xledsetnormalparstuff	1
\xleft@appenditem	1
\Xlemmadisablefontselection	41

\Xlemmafонт	42
\Xlemmaseparator	40
\Xlinangeseparator	36
\xlineref	1, 53
\Xmaxhnotes	46
\Xmorethantwolines	37
\Xnolemmaseparator	1, 41
\Xnonbreakableafternumber	38
\Xnonumber	37
\Xnotefontsize	42
\Xnotenumfont	41
\Xnoteswidthliketwocolumns	47
\Xnumberonlyfirstinline	35
\Xnumberonlyfirstintwolines	35
\Xonlypstart	38
\Xpagelinesep	38
\xpageref	1, 53
\Xparafootsep	44
\Xparindent	42
\Xprenotes	1, 45
\Xprenotes@	1
\Xpstart	37
\Xstarteverytime	37
\xpstartref	1, 53
\XR@test	1
\XR@test@mac	1
\XR@test@mac@test	1
\Xragged	44
\xright@appenditem	1
\Xrigidbalance	1
\Xstanza	38
\Xstanzaseparator	38
\xsublineref	1, 53
\Xsublinesep	38
\Xsymlinenum	35
\Xtoendnotes	25
\Xtwolines	37
\Xtwolinesonlyinsamepage	37
\Xtxtbeforenotes	45
\Xunvxh	1
\Xwidth	46
\Xwrapcontent	42
\Xwrapendlemma	42
\Xwraplemma	42
\xxref	1, 53

Z

\zz@@@	1
--------	---

Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
\ifl@dmemoir: Added \ifl@dmemoir for memoir class having been used	73
\morenoexpands: Added \l@dtabnoexpands to \no@expands	122
\reledmac@error: Added \eledmac@error and replaced error messages	75
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	257
\doxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	248
\edlabel: Tweaked \edlabel to get correct page numbers	254
\l@ddodoreinxtrafeet: Renamed \dodoreinxtrafeet to \l@ddodoreinxtrafeet	250
\morenoexpands: Removed some \lets from \no@expands. These were in edmac but Peter Wilson feels that they should not have been as they disabled page/line refs in a footnotes	122
\zz@@@: Minor change to \zz@@@	254
General: Added text about normal labeling	54
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	251
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	166
\line@list@stuff: Added initial write of page number in \line@list@stuff	114
\para@footsetup: Added \footfudgefiddle to \para@footsetup	166
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	206
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	81
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	257
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	102
\ledlinenum: Added \linenumr@p and \sublinenum@rep to \leftlinenum and \rightlinenum	93
\linenumberlist: Added \linenumberlist mechanism	81
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	223
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	189
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	93
General: Includes edstanza and more	1
v0.3.1.	
General: Not released. Added remarks about the parallel package	1
v0.4.0.	
\@iiiminipage: Modified kernel \@iiiminipage and \endminipage to cater for critical footnotes	278
\Xarrangement@normal: Added minpage footnote setup to \footnormal	161
\edtext: Added \showlemma to \edtext	123
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	276

\mpnormalfootgroup: Added \mpnormalfootgroup	165
\mpnormalvfootnote: Added \mpnormalvfootnote	162
\showlemma: Added \showlemma	81
General: Added final/draft options	71
Added ledgroup environment	279
Added ledgroupsize environment	280
Added minipage, etc., support	1
v0.4.1.	
\do@Xfeet: Changed \do@Xfeet code for easier extensions	249
\edindex: Let eledmac take advantage of memoir's indexing	285
\print@Xnotes: Added \op@Xfeet	249
General: Added code for changing \doclearpage	252
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	192
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	193
\@xympar: Eliminated \marginpar disturbance	268
General: Added left and right side notes	269
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	147
\l@edgetsidenote@margin: Added \sidenotemargin and \sidenote@margin	269
General: Added moveable side note	269
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	280
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	104
\@nl@reg: Added \fix@page to \@nl	102
Extended \@nl to include the page number	102
\fix@page: Added \last@page@num and \fix@page	103
\get@thisfootnote: Changed \l@dbfnote and \vl@dbfnote as originals could give incorrect markers in the footnotes	193
\new@line: Extended \new@line to output page numbers	115
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	102
\@ref@reg: Added \@ref@reg	111
\affixline@num: Added skipnumering to \affixline@num	147
\do@actions@fixedcode: Added \do@actions@fixedcode	146
\do@actions@next: Added number skipping to \do@actions	145
\do@insidelinehook: Added \do@linehook for use in \do@line	143
\endnumbering: Changed \endnumbering for eledpar	84
\fix@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fix@l@cks	150
\footplitskips: Added \footplitskips for use in many footnote styles	159
\get@linelistfile: Added \get@linelistfile	100
\initnumbering@reg: Added \initnumbering@reg	83

\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	279
\l@dcsnotetext@r: Added \l@emptyd@ta	143
\l@dgetline@margin: Added \l@dgetline@margin	89
\l@dgetlock@disp: Added \l@dgetlock@disp	91
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	269
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	81
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	143
\l@dzeropenalties: Added \l@dzeropenalties	138
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	93
\line@list@stuff: Deleted \page@start from \line@list@stuff	114
\list@clearing@reg: Added \list@clearing@reg	100
\n@num: Added \n@num	109
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	199
\resumenumbering: Changed \resumenumbering for eledpar	85
\setprintendlines: Added \setprintendlines for use by \printendlines	222
\setprintlines: Added \setprintlines for use by \printlines	185
\skipnumbering: Added \skipnumbering and supports	118
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	91
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	93
Using \sublinenumrep instead of \sublinenumr@p	93
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	200
General: eledmac having been available for 2 years, deleted the commented out original edmac texts	1
Maieul Rouquette new maintainer	1
Made macros of all messages	74
Replaced all \interAfootnotelinepenalty, etc., by just \interfootnotelinepenalty	1
Tidying up for eledpar and ledarab packages	1
v0.8.0.	
General: Bug on endnotes fixed: in a // text, all endnotes will print and be placed at the ends of columns ()	1
v0.8.1.	
General: Bug on \edtext ; \critex ; \lemma fixed: we can now us non-switching commands	1
v0.9.0.	
General: No more ledpatch. All patches are now in the main file.	1
v0.9.1.	
General: Fix some bugs linked to integrating ledpatch on the main file.	1
v0.10.0.	
General: Corrections to \section and other titles in numbered sections	1
v0.11.0.	
General: Makes it possible to add a symbol on each verse's hanging, as in French typography. Redefines the command \hangingsymbol to define the character.	1
v0.12.0.	
\l@dnumpstartsL: Added \ifledRcol and \ifnumberingR for/from eledpar	81
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	18

Possibility to number the pstart with the commands <code>\numberpstarttrue</code>	1
v0.12.1.	
General: Don't number <code>\pstarts</code> of stanza.	1
The numbering of <code>\pstarts</code> restarts on each <code>\beginnumbering</code>	1
v0.13.0.	
<code>\managestanza@modulo</code> : New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	293
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses. . .	48
New <code>stanzaindentsrepetition</code> counter: to repeat stanza indents every n verses.	1
v0.13.1.	
General: <code>\thepstartL</code> and <code>\thepstartR</code> use now <code>\bfseries</code> and not <code>\bf</code> , which is deprecated and makes conflicts with memoir class.	1
v0.14.0.	
<code>\edlabel</code> : Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph.	254
General: Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph.	1
v0.15.0.	
<code>\affixline@num</code> : Line numbering can be disabled.	147
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	292
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code>	188
General: Line numbering can be reset at each <code>pstart</code>	88
Possibility to print <code>\pstart</code> number inside.	18
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	292
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	122
General: <code>\lemma</code> can contain commands.	26
Debug in lineation command	20
New generic commands to customize footnote display.	34, 239
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code>	24
Options of <code>\Xfootnotes</code>	157
Possibility to have commands in sidenotes.	56
Some compatibility break with <code>eledmac</code> . Change of name: <code>eledmac</code>	1
v1.0.1.	
General: Correction on <code>\Xnumberonlyfirstinline</code> with lineation by <code>pstart</code> or by page. . .	35
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code>	214
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	240
General: Add <code>\labelpstarttrue</code>	18
Add <code>\Xnumberonlyfirstintwolines</code>	35
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	37
New hook to add arbitrary code at the beginning of the notes	43
New options for block of notes.	45
New package option: <code>parapparatus</code>	1
New tools to change order of series	238
Sectioning commands.	63

v1.2.0.	
\Xprenotes:	Debug in familiar footnotes (bug introduced by v1.1). 214
\endquote:	Compatibility of \ledchapter with the <i>memoir</i> class. 328
v1.3.0.	
\endquote:	<i>Quotation</i> and quote environment inside numbered sections. 328
v1.4.0.	
\edtext:	Compatibility of \edtext with the right-to-left direction (with Polyglossia). 123
\ledsetnormalparstuffX:	Direction of footnotes with polyglossia. 211
\newseries@:	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. 229
\rbracket:	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). 180
General:	Compatibility with LuaTeX of RTL notes. 1
v1.4.1.	
\affixside@note:	Remove spurious spaces. 275
\endquote:	New option <i>noquotation</i> 328
\get@thisfootnote:	Compatibility of standard footnotes with eledmac when these footnotes contain any commands. 193
\labelrefsparsesubline:	Fix bug with \edlabel. 256
v1.4.2.	
General:	Debug with some special classes. 1
v1.4.3.	
General:	Add \Xnonbreakableafternumber. 38
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. 328
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. 199
v1.5.0.	
\do@insidelinehook:	Added \do@insidelinehook for use in \do@oline 143
\edindex:	Compatibility with imakeidx package, and possibility to use multiple index with \edindex. 285
General:	Correct indexing when the call is made in critical notes. 281
v1.5.1.	
\managestanza@modulo:	Correct stanzaindentsrepetition counter 293
\normalvfootnoteX:	Fix bug with normal familiar footnotes when mixing RTL and LTR text. 196
v1.6.0.	
\newverse:	Add \falseverse macro. 296
v1.6.1.	
\AtEveryPstart:	Spurious space in \pstart. 135

\ifinserthangingsymbol: Hang verse is now not automatically flush right.	292
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill inside.	140
\pend: Spurious space in \pend.	136
General: Corrects a false hanging verse when a verse is exactly the length of a line.	1
v1.7.0.	
General: New features for managing page breaks.	65
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts.	328
\get@index@command: Debug \get@index@command and compatibility with hyperref package.	284
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work.	243
\prevpage@num: Correct \parafootsep when using with ledgroup.	172
General: Compatibility with parledgroup option ofeledpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	281
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	216
v1.8.2.	
General: Debug compatibility problem with hebrew option of babel package.	1
v1.8.3.	
General: Fixes spurious spaces added by v1.7.0.	1
v1.8.5.	
General: Debug indexing in right column, witheledpar.	281
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes.	248
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup.	276
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	328
General: Add \pstartref and \xpstartref to refer to a pstart number (extension of \edlabel).	1
v1.10.1.	
General: Compatibility with cleveref.	1
v1.10.2.	
General: Compatibility of stanza with v1.8a of babel-greek.	1
v1.10.3.	
General: Debug of cross-referencing.	1
v1.10.4.	
General: Debug of critical notes in edtabular environment.	1
v1.10.5.	
General: Debug of \pausenumbering.	1
Debug of \xxref.	1
v1.10.6.	
General: Debug of interaction between \autopar and \pausenumbering.	1
v1.11.0.	
General: Add hooks to disable the font selection for lemma in footnote.	41
v1.11.1.	
General: Correct a bug when a critical note starts with plus or minus.	1

v1.12.0.

\@nl@reg: To ensure compatibility with \musixtex, \@l becomes \@l. Consequently, \@l@reg becomes \@nl@reg.	102
\AtEveryPstart: New optional argument for \pstart, to execute code before it. . .	135
\edindex: Use correctly default index when imakeidx is loaded.	285
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	328
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar. .	83
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	269
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in eledmac, including needs for eledpar case.	270
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	81
\l@dunhbox@line: \do@line is split in more little commands.	140
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	243
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	243
\pend: New optional argument for \pend, to execute code after it.	136
\stanza: &can have an optional argument: content to be printed after.	296
\Stanza can have an optional argument: content to be printed before.	296
Add \newverse macro, \falseverse deprecated.	296
General: Add \ledinnernote and \ledouternote commands.	56
Add \Xendparagraph and related settings.	47
Add hyperlink to crossref (needs hyperref package).	52
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	281
New hooks: \Xafterrule and \afterruleX	46
New options for ragged-paragraph notes	44
New sectioning commands.	63
Optional arguments for \pstart and \pend.	18

v1.12.1.

\wrap@edcrossref: Fix spurious spaces.	259
--	-----

v1.12.2.

\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	140
--	-----

v1.12.3.

\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	116
\flag@start send a error message when a \edtext is done without insert (note)	116
\reledmac@error: Replaced error messages	75
General: Add macros for new messages since v0.7	74
Correct bug with side and familiar notes in tabular environments.	1
Debug \eledxxx with some paper size	1
Debug \ledinnernote and \ledouternote commands in the top of pages.	56
Debug left and right notes (bugs added by 1.12.0)	1
Underline lemma in \eledxxx when using draft mode.	1

v1.12.4.

General: Debug spurious page breaks before \chapter (bug added in 1.12.0)	1
---	---

v1.12.5.	
\@edindex@hyperref: Debug \edindex when hyperref is not loaded	287
\@ssect: Debug \eledchapter in parallel with memoir	332
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	143
\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	84
\l@dgobblearg: \l@dgobblearg becomes \l@gobbeloptarg	312
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	312
\resumenumbering: Debug \resumenumbering	85
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	259
v1.12.8.	
\flag@end: \flag@start do not send a error message when a \edtext is done	
without insert (note) but have a endnote	116
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	242
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	206
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	240
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	47
Added widthliketwocolumns option	71
v1.13.1.	
\thepstart: Add \l@dzeropenalties in \pstart	135
General: Coming back of page and line breaking penalties's management, deleted by	
error in v0.17.	1
Debug quotation environment inside of a \pstart preceded by a sectioning command.	1
v1.13.2.	
\l@dnumpstartsl: Add \ifl@dpadding for \eledpar	81
General: Fix bug with normal footnotes, added by v1.13.0.	1
v1.13.3.	
General: Fix extra spaces with paragraphed footnotes, added by v1.13.0.	1
v1.13.4.	
General: Fix bug with index when memoir class is used without hyperref	1
v1.14.0.	
\edindex: Let \eledmac take advantage of \makeidx even when memoir class is used	285
General: Debug spurious characters before endnotes.	216
Delete previous override of \l@d@@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	288
Move gobbling command	81
Provide \@gobblefour	81
v1.14.1.	
\@ssect: Debug sectioning commands when using both <code>handout</code> and <code>hyperref</code>	
package.	334
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	332
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	122
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	206
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	117
\if@lemmacommand@: New boolean \iflemmacommand@.	127

General: Fix bug with footnotes layout when using some options of the geometry package (bug add by v1.13.0).	1
New commands <code>\AtEveryPstart</code> and <code>\AtEveryPend</code>	18
New tools to prevent ambiguous references in lemma	27
v1.15.1.	
<code>\line@list@stuff</code> : Revert modification of 1.5.2 which makes bug with numbering. Leave vertical mode to solve spurious space before minipage.	114
v1.16.0.	
General: <code>\edtext</code> is now defined only in <code>eledmac</code> , not in <code>eledpar</code> . Debug wrong numbering when using <code>\sameword + eledpar + \tag</code> command.	122
Compatibility of standard footnotes with some biblatex styles.	1
New <code>\stanzaindent</code> command.	1
v1.16.1.	
<code>\xlineref</code> : <code>\lineref</code> is not defined if defined by some other package, like <code>lineno</code> . Eledmac provides <code>\edlineref</code> instead.	259
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph. . . .	123
v1.18.0.	
<code>\@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools + hyperref</code>	287
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code> . . .	254
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes.	217
<code>\l@dnumpstartsL</code> : Add <code>\ifl@dprintingpages</code> and <code>\@dprintingcolumns</code> for <code>eledpar</code>	81
<code>\print@line</code> : Compatibility with Lua \TeX RTL languages.	140
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code>	181
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns (<code>eledpar</code>)	180
General: Add <code>\Xpstarteverytime</code>	37
Compatibility with Lua \TeX RTL languages.	1
Debug <code>\Xonlypstart</code> when using <code>\Xnumberonlyfirstinline</code> and the current line number differs from the previous.	37
v1.19.0.	
<code>\footssplitskips</code> : <code>\footssplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages.	159
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty. .	261
General: <code>\Xmaxhnotes</code> and <code>\maxhnotesX</code> work now for both two-columns and three-columns setting.	1
Compatibility with <code>eledpar</code> v1.13.0.	1
v1.19.1.	
General: Call <code>\correct@footinsX@box</code> and <code>\correct@Xfootins@box</code> directly in <code>\print@notesX@forpages</code> and <code>\print@Xnotes@forpages</code> , that is in <code>eledpar</code> . . .	1
v1.20.0.	
<code>\printlines</code> : Added <code>\ifl@d@Xmorethantwolines</code> and <code>\ifl@d@Xmorethantwolines</code> to <code>\printlines</code>	189
<code>\stanza</code> : <code>&</code> and <code>&</code> can be preceded by spaces.	296
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (fix bug added in 1.19.0).	261
General: Add <code>\Xendboxlinenum</code>	40
Add <code>\Xtwolines</code> and <code>\Xmorethantwolines</code> hooks	37
Add series option.	1

Correct <code>\Xinplaceofnumber</code> hook.	1
Explicit error message when calling <code>\Xfootnote</code> outside of <code>\edtext</code>	1
Fix bug with line number typesetting direction when using <code>\eledsection</code> and similar commands for RTL texts with <code>Lua\TeX</code>	1
Fix issues with RTL text in notes when using <code>Lua\TeX</code>	1
Options fulllines in <code>\Xfootnote</code>	24
The <code>\newifs</code> are not followed by boolean values set to false, because it is the <code>\TeX</code> default setting.	1
v1.21.0.	
<code>\@edindex@hyperref</code> : Look at the hyperindex option of hyperref before inserting hyperref	287
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	217
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code>	284
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code>	211
<code>\n@num</code> : <code>\n@num@ref</code> deleted	109
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	109
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	242
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	242
<code>\print@footnoteXrule</code> : Code refactoring: the spaces after the footnote rules are directly managed in <code>\print@Xfootnoterule</code> and <code>\print@footnoteXrule</code>	213
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	239
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	118
Correct <code>\skipnumbering</code> for stanza.	118
Delete <code>\skipnumbering@reg</code>	118
General: <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> are now compatible with <code>\autopar</code> . . .	1
<code>\Xafterrule</code> and <code>\afterruleX</code> features no longer create problems of overflowing at the bottom of the page.	1
<code>\chapter</code> inside optional argument of <code>\pstart</code> works when typesetting parallel pages	1
<code>\preXnotes</code> and <code>\prenotesX</code> features no longer create problems of overflowing at the bottom of the page.	1
<code>\seriesatbegin</code> and <code>\seriesatbegin</code> more efficient	238
Add <code>\applabel</code> and related	54
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column.	1
Add <code>\hidenumbering</code>	22
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks	43
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code>	37
Add <code>\Xparindent</code> and <code>\hangindentX</code>	42
Add <code>\Xtwolinesbutnotmore</code> and <code>\Xtwolinesonlyinsamepage</code>	1
Add <code>nocritical</code> , <code>noend</code> , <code>nofamiliar</code> and <code>noledgroup</code> options.	1
Add <code>noeledsec</code> package option	1
Debug <code>\beforenotesX</code> <code>\maxhnotesX</code> <code>\noteswidthliketwocolumnsX</code> and <code>\afterruleX</code> with footnotes set in two and three columns.	1
Fix bug when a <code>\Xfootnote</code> follows a <code>\Xendnote</code> in the second argument of <code>\edtext</code> (bug added in <code>eledmac 1.0.0</code>).	1

Fix bug with <code>\maxnotesX</code> when using <code>\foottwocolX</code> or <code>\footthreecolX</code>	1
Fix bug with space between columns with notes in two columns (bug added in v1.13.0).	1
Fix spurious space after first page number in <code>\doendnotes</code> . <code>oldprintnpnumspace</code> option allows to come back to previous setting	1
<code>parapparatus</code> option works now with familiar footnotes.	1
Provide <code>\@gobblefive</code>	81
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	284
General: Add <code>\doendnotesbysection</code> command.	25
Add option for lemma separator inside endnotes	41
Adds hyperlink for references to notes in indices.	1
Fix conflict between <code>noend</code> package option and <code>edtabularx</code> environments	1
Provides support for <code>xindy</code>	1
Standardize endnotes handbook.	25
When using <code>hyperref</code> package, internal links in index or with <code>\edlineref</code> are now targeted to the top and not longer to the bottom of the lines they refer to.	1
v1.22.1.	
<code>\prevpage@num</code> : Correct double symbol when using both <code>\parafootsep</code> and <code>\Xsymlinenum</code>	172
General: Fix bug (added on v1.22.0) with <code>\Xinplaceofnumber</code> hook.	1
v1.23.0.	
<code>\@edtext@level</code> : The boolean <code>\if@edtext@</code> becomes the counter <code>\edtext@level</code>	122
<code>\SErefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code>	265
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	127
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	229
<code>\sameword</code> : In order to allow use of <code>\sameword</code> with <code>inputenc</code> , we detokenize its mandatory argument before using it in control sequence names.	131
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code>	39
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code>	39
Allow use of <code>\sameword</code> with <code>inputenc</code> managing of UTF-8.	1
Compatibility between <code>nofamiliar/nocriticals</code> option and <code>minipage/ledgroup</code>	1
Error message when using <code>\beginnumbering... \endnumbering</code> without <code>\pstart</code>	1
Fix bug with <code>\sameword</code> when the lemma overlaps multiple line.	27
Fix bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code>	27
Fix bug with <code>\skipnumbering</code> called immediately after a <code>\pstart</code>	1
Fix error of <code>\iftrue</code> not closed.	1
Fix spurious space with <code>\skipnumbering</code> (bug added on v1.21.0).	1
New tools to ensure the line-list file uses the right version of commands when upgrading the <code>eledmac</code> version.	1
Optional argument of <code>\sameword</code> can be a comma-separated list of <code>\edtext</code> depth.	27
v1.23.1.	
General: Fix bug with <code>\lemma</code> command in the right side.	1
v1.23.2.	
General: Compatibility with L ^A T _E X's release 2015.	1

v1.24.0.	
General: We can reinitialize <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> providing to it an empty argument.	1
v1.24.1.	
General: <code>\lemma</code> is disabled when using ‘ <code>nocritical</code> ’ option.	1
v1.24.2.	
General: Fix incompatibility between ‘ <code>nofamiliar</code> ’ option and ‘ <code>memoir</code> ’ package.	1
v1.24.3.	
General: Restore marginal numbers and notes with sectioning command (bug introduced in v1.21.0)	1
v1.24.4.	
General: Fix spurious space with <code>\edindex</code> when using <code>xindy+hyperref</code> option.	1
v1.24.5.	
General: Fix bug of indent, when a added in 1.1.0, when a <code>\beginnumbering</code> immediately follow a sectioning command.	1
v2.0.0.	
<code>\@iiiminipage</code> : Patch <code>\@iiiminipage</code> instead of redefining it.	278
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	268
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	278
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	328
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some conde of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	279
<code>\newseries@</code> : One endnotes file by series.	235
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined	251
<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>	251
Add <code>\Xendinplaceofnumber</code> hook.	1
Add <code>\Xendnonumber</code> hook.	1
Add <code>nonum</code> option for endnotes.	1
Fix bug when printing only one series of endnotes, but wanted to keep endnotes for other series.	1
In order to have a more consistent name’s convention, many names has been changed.	1
Many \TeX ’s output macros are now patched and not override.	1
Package’s name becomes <code>reledmac</code>	1
Patch <code>\@footnotemark</code> instead of redefine it	192
Suppress indexing command specific to <code>memoir</code>	285
v2.0.1.	
General: Fix bug in <code>eledmac-compat</code> option	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code>	1
v2.1.0.	
General: Fix bug with <code>\advanceline</code> at the beginning of a <code>\pstart</code>	1
Fix bug with <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting with <code>scrbook</code>	1
Fix bug with <code>\eledchapter</code> in parallel typesetting with <code>scrbook</code>	1
Fix bug with <code>\setline</code> at the beginning of a <code>\pstart</code>	1
Fix spacing bug with <code>\Xbhooknote</code> and <code>\bhooknoteX</code> when using them to insert text and not to execute code.	1

New tools to number stanzas	1
v2.1.1.	
General: Fix bug with <code>\ledpbsetting{before}</code>	1
v2.1.2.	
General: Fix bug with lineation by <code>pstart</code> and <code>tabular</code> environments (added in 2.1.0).	1
v2.1.3.	
<code>\ledsetnormalparstuffX</code> : Replaced <code>\noindent</code> with <code>\parindent</code> set to 0pt.	211
General: <code>\Xhangindent</code> and <code>\hangindentX</code> work now with all the paragraphs in the note.	1
<code>\Xnoindent</code> and <code>\noindentX</code> work now again (broken in 2.0.0).	1
Change some internal code in order to provide compatibility with \TeX release of october 2015	1
Fix bug which inserted double space before paragraphed familiar notes.	1
Fix bug with <code>\edindex</code> when using not-Latin characters without UTF-8 engines	1
v2.2.0.	
General: Fix bug with combination of <code>\onehalfspacing</code> and two columns and three columns notes typeset.	1
Fix bug with some setting command and optimization option.	1
Fix spurious space with paragraphed critical notes when using $\text{Lua}\TeX$	1
Increase line list version number to ensure compatibility with new options of <code>reledpar</code> package.	1
New setting tools for endnotes: <code>\Xendnumberonlyfirstinline</code> , <code>\Xendnumberonlyfirstintwolines</code> , <code>\Xendsymmlinenumber</code> , <code>\Xendbeforenumber</code> , <code>\Xendafternumber</code> , <code>\Xendbeforemsymmlinenumber</code> , <code>\Xendaftersymmlinenumber</code> , <code>\Xendboxsymmlinenumber</code> , <code>\Xendhangindent</code> , <code>\Xendbhooklinenumber</code> , <code>\Xendahooklinenumber</code> , <code>\Xendbhookinplaceofnumber</code> , <code>\Xendahookinplaceofnumber</code>	1
v2.2.1.	
General: Compatibility with TeXformat 2015/10/01.	1
v2.2.2.	
General: Fix bug in <code>\sethangingsymbol</code>	1
Fix bug with old version of <code>etex</code>	1
v2.3.0.	
General: Disable empty lines as paragraph in stanza.	1
Fix compatibility of paragraphed footnotes with <code>bidi</code> v17.9 and following.	1
Warning message when using some setting commands inside <code>rightside</code> environment (deprecated behavior)	1
v2.3.1.	
General: Fix spurious space when using optional argument of <code>\stanza</code> (introduced in v2.3.0).	1
v2.4.0.	
<code>\footnoteoptions@</code> : First argument of <code>\footnoteoption@</code> is now mandatory, not optional.	157
General: <code>\Xbhooknote</code> and <code>\bhooknoteX</code> work with notes in columns.	1
Fix bug of <code>\parindentX</code> and <code>\Xparindent</code> with two columns and three columns notes.	1
Fix bug with <code>\sameword</code> in right side.	1
Fix spurious space in two columns and three columns notes.	1
Fix spurious space when using optional argument of <code>stanza</code> (introduced in v2.3.0).	1
New hooks: <code>\Xlinerangeseparator</code> and <code>\Xendlinerangeseparator</code>	36

Option <code>linrangesep</code> for critical footnotes and endnotes.	36
v2.4.1.	
General: Fix bug with <code>\appref</code> and <code>\apprefwithpage</code> (introduced in v2.4.0).	1
Fix bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override \LaTeX <code>\roman</code> command, like Greek language.	1
Fix bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override \LaTeX <code>\roman</code> command, like Greek.	1
v2.5.0.	
<code>\SErefwithpage</code> : Debug <code>\setapprefprefixsingle</code>	264
<code>\edlabel</code> : Fix bug when calling <code>\edlabel</code> in a footnotes of the rightside	254
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments.	217
<code>\printlines</code> : <code>\printlines</code> takes an eighth argument: the line flag	188
<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.	259
General: <code>\apprefwithpage</code> and <code>\appref</code> print double quotation mark when the label was not defined.	1
<code>\apprefwithpage</code> and <code>\appref</code> work with right side crossref.	1
<code>\apprefwithpage</code> works also when <code>noend</code> option is enabled.	1
<code>\appref</code> and <code>\apprefwithpage</code> can take <code>linrangesep</code> optional argument.	1
<code>\edlabel</code> works now in <code>Xfootnote</code>	1
<code>\lemma</code> can be used even when the <code>nocritical</code> is enabled.	1
Compatibility with new hook and tools of <code>reledpar</code> 2.6.0.	1
Fix spurious vertical space in <code>astanza</code> environment (<code>reledpar</code>)	1
Log now states ‘There were undefined references’ when using wrong references in <code>\edlineref</code> or <code>edpageref</code>	1
New hooks to customize page and line number appearance in endnotes.	1
New hooks: <code>\Xhookgroup</code> and <code>\bhookgroupX</code>	1
New tools to easily make cross-reference to a passage defined by a start and an end line	54
v2.6.0.	
General: Adds compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code> package.	1
Fix bug with footnote counter in <code>ledgroup</code> (added in v2.5.0).	1
Fix bug, introduced in v2.5.0, with footnote numbering in parallel typesetting when using <code>perpage</code> package.	1
v2.7.0.	
<code>\@k</code> : <code>\rigidbalance</code> is split in <code>\Xrigidbalance</code> and <code>\rigidbalanceX</code>	173
<code>\l@d@section</code> : <code>\endnotes</code> take seven arguments.	217
General: Add dash as default page range separator for <code>\SEonlypage</code>	1
Debug <code>\SErefonlypage</code> when referring to only one page.	1
Delete parenthesis after <code>\SErefonlypage</code>	1
Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.5.0).	1
Fix bug with <code>\SErefwithpage</code>	1
Fix bugs in compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code> package, when indexing outside of a <code>ledgroup</code>	1
New commands to make glossaries connected to page and linenumber with the <code>glossaries</code> package	1
New hooks: <code>\Xhsize</code> and <code>\hsizeX</code>	46
New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</code>	42

New setting commands: <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code>	1
Warning for duplicate and undefined labels are parsable by latexmk	1
Warning for duplicate labels does not send any more a false line and page number	1
When using <code>hyperref</code> package, add link in familiar footnotes between the footnote marks in the text and the footnote marks in the footnote	1
When using <code>hyperref</code> package, add links for <code>\SEref</code> and <code>related</code> , <code>\appref</code> and <code>related</code>	1
When using <code>hyperref</code> package, add links from critical footnotes and critical endnotes to the line of text they refers	1
v2.7.1.	
General: Debug <code>\Xhookgroup</code> hooks executed on columnar footnotes (moved to a larger group, to take effect).	1
v2.7.2.	
General: <code>\Xhsize</code> and <code>\hsizeX</code> become <code>\Xwidth</code> and <code>\widthX</code>	46
Fix problem of hyphenation when using <code>hyperref</code> package (added in v2.7.0).	1
v2.8.0.	
<code>\l@d@section</code> : <code>\Xendhangindent</code> and <code>\Xendafternote</code> can take values which are relative to the font size of the endnote.	217
General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package.	1
More <code>\edgls...</code> commands.	1
No indentation for paragraphed notes in <code>ledgroup</code> . Can be changed with <code>\Xparindent</code> and <code>\parindentX</code>	1
v2.8.1.	
General: Warnings for undefined labels are really parsable by latexmk	1
v2.8.2.	
General: Fix bug concerning indent in a paragraph immediately following a sectioning command (bug NOT fixed on <code>reledpar</code>)	1
Fix bug with <code>\AtEveryPstart</code> added in version 2.0.0.	1
Fix bug with vertical space after the between-sectioning command as optional argument of a <code>\pstart</code> and <code>\pstart</code> content	1
v2.9.0.	
General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> option.	1
Fix bug when using <code>\linenum{page}</code> and <code>\pausenumbering... \resumenumbering</code>	1
Fix bug with three- and two-column footnote setting (added in v2.4.0).	1
Fix spurious space inside three-column familiar footnote.	1
Write correct metadata in numbered files when using <code>\pausenumbering... \resumenumbering</code>	1
v2.9.1.	
General: Fix bug when notes start with “plus” or “minus”.	1
v2.9.2.	
General: Fix bug with <code>hyperref</code> package when a lemma starts with “plus” or “minus” (bug introduced in v. 2.7.0).	1
v2.9.3.	
General: Fix bug with line number position and reset added by v. 2.9.0	1
v2.10.0.	
<code>\print@lemma</code> : Code refactoring between <code>\parafootfmt</code> , <code>\twocolfootfmt</code> , <code>\threecolfootfmt</code> and <code>\normalfootfmt</code>	181

General: Add <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> .	1
Fix bug in <code>\ledlsnotefontsetup</code> and <code>\ledrsnotefontsetup</code> which could not handle <code>\color</code> command properly.	1
More specific error messages.	1
New hooks: <code>\Xwrapcontent</code> , <code>\Xendwrapcontent</code> and <code>\wrapcontentX</code> .	42
New hooks: <code>\Xwraplemma</code> and <code>\Xendwraplemma</code> .	42
v2.10.1.	
General: Add ‘nopenalties’ option.	1
Fix bug introduced in v. 1.4: not paragraphed critical footnotes could prevent marginal line number from being displayed	1
v2.11.0.	
<code>\do@actions@fixedcode</code> : Add action 1010	146
General: Add new tools to produce an apparatus of manuscripts	1
Fix bug in <code>\Xparafootsep</code> in parallel typesetting	1
Make <code>\parafootsepX</code> work	1
Prevent <code>\Xtxtbeforenotes</code> hook from causing notes to go beyond the bottom margin	1
v2.12.0.	
General: <code>\preXnotes</code> becomes <code>\Xprenotes</code> (naming convention)	1
Add <code>\hidenumberingonleftpage</code> and <code>\hidenumberingonrightpage</code>	1
Add <code>\toendnotes</code> and related.	1
Add <code>auxdir</code> option.	1
Fix bug in critical and familiar footnotes when using uppercase letters with accent mark	1
Fix bug when using <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting in combination with the <code>noeledsec</code> option.	1
Fix bug with <code>\ledinnernote</code> and <code>\ledouternote</code> in parallel typesetting	1
Fix bug with familiar footnote number in optional argument of <code>\pstart</code> or <code>\pend</code> in parallel typesetting	1
Fix spurious vertical space in <code>\chapter</code> when used as optional argument of <code>\pstart</code> in parallel typesetting.	1
Make endnote compatible with <code>\sameword</code> mechanism	1
More accurate message to control the position of <code>\Xfootnote</code> and <code>\applabel</code> in the \TeX code	1
v2.13.0.	
General: Version 2.13.0 never existed.	1
v2.13.1.	
General: In critical footnotes, the right side flag is printed only if requested explicitly with <code>\Xlineflag</code> (bug added in v. 2.5.0).	1
v2.13.2.	
General: Fix bug added in v. 11.2 which could make parallel typesetting not work.	1
v2.13.3.	
General: Makes <code>\Xendafterpagenumbe</code> affecting <code>\Serefwithpage</code>	1
v2.14.0.	
General: Hyperref with the line number inside critical footnotes is correct when using <code>\xxref</code>	1
Some internal changes for new features of <code>reledpar</code> .	1
v2.14.1.	
General: Fix bug when using <code>\footnoteX</code> in the first argument of <code>\edtext</code> .	1
v2.14.1a.	
General: Fix problematic typos in the handbook.	1

v2.15.0.	
General: Add ‘byline’ arrangement.	1
Fix <code>\Xtxtbeforenotes</code> in <code>ledgroup</code>	1
v2.15.1.	
General: Fix <code>\edindex</code> in tabular environments.	1
v2.15.2.	
General: Fix bug with <code>fancyhdr</code> package 3.8 and later.	1
v2.15.3.	
General: Fix bug with <code>\section</code> in optional argument of <code>\pstart</code> and empty line before <code>\pend</code> (bug added in v2.8.2).	1
Simplification of the sectioning command code.	1
v2.16.0.	
General: <code>\Xdo@feet</code> becomes <code>\do@Xfeet</code>	251
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1
Add <code>\Xpagelinesep</code> hook.	38
Compatibility with new features of <code>reledpar</code>	1
Deleted dead code.	1
Display a warning message if using a version of \TeX that is too old.	1
Fix bug with <code>\Xgroupbylines</code> for notes in two columns	1
Fix bug with <code>\Xtxtbeforenotes</code> for notes in three or two columns	1
Fix bug with ‘notenumber’ option of <code>indextools</code> package when indexing texts in familiar footnotes.	1
Fix potential bug when using <code>\edindex</code> in critical footnotes.	1
More explicit error message in case the stanza indentation is not defined.	251
New options for <code>\fnpos</code> and <code>\mpfnpos</code> to set a customized order for familiar and critical footnotes.	1
When <code>\edindex</code> is called outside of a <code>\beginnumbering... \endnumbering</code> structure, it is automatically switched to <code>\index</code> , with a warning message.	1
When indexing texts in familiar footnotes with <code>\edtext</code> , refer to the line number where the footnote is called.	1
When indexing texts in sidenotes with <code>\edtext</code> , refer to the line number where the sidenote is called.	1
v2.16.1.	
General: Fix bug with redefinition of the style of the footnote number (bug added in v2.12.0)	1
v2.16.2.	
General: Error message if <code>footmisc</code> is loaded after <code>reledmac</code>	1
Fix bug introduced by v2.16.1 when using non-expandable control sequence, like <code>\normalfont</code> , in the footnote number style.	1
v2.16.3.	
General: Fix bug with <code>\Seref</code> (bug added in v2.7.0).	1
v2.16.4.	
General: Fix bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.15.3).	1
v2.16.5.	
General: Fix potential spurious spaces in endnotes.	1

v2.16.6.

- General: Fix bug with the line number style in `\doenotes` when referring to right side
line in parallel typesetting. 1
- Take into account `\linenumberstyle` when using `\edlineref`. 1

v2.16.7.

- General: Fix bug with `\msdata` when using multiple
`\beginnumbering...endnumbering` 1
- Fix bug with `\numberpstarttrue` when using multiple
`\beginnumbering...endnumbering` 1

v2.16.8.

- General: Fix bug with `\edindex` in footnote in parallel typesetting 1