

reledmac

Typeset scholarly editions with L^AT_EX^{*}

Maïeul Rouquette[†]

based on the original ledmac by

Peter Wilson

Herries Press

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

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

Abstract

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

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

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

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

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

Contents

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

^{*}This file (**reledmac.dtx**) has version number v2.18.1, last revised 2017/03/20.

[†]maieul at maieul dot net

1.2.4 <code>reledmac</code>	14
1.3 Bibliography	14
2 How the package works — the problem of the number of \LaTeX runs	14
3 Compatibility warning	15
4 Options	15
4.1 Specific features	15
4.2 Optimizing package performance	16
5 Text lines and paragraphs numbering	16
5.1 Text lines numbering	16
5.2 Paragraphs	17
5.2.1 Basics	17
5.2.2 Automatically producing <code>\pstart ... \pend</code>	17
5.2.3 Content before specific <code>\pstart</code> and after specific <code>\pend</code>	18
5.2.4 Content before every <code>\pstart</code> and after every <code>\pend</code>	18
5.2.5 Numbering paragraphs (<code>\pstart</code>)	18
5.2.6 Languages written in Right to Left	19
5.2.7 Memory limits	19
5.3 Lineation commands	20
5.3.1 Disabling lineation	20
5.3.2 Setting lineation start and step	20
5.3.3 Setting lineation reset	20
5.3.4 Setting line number margin	20
5.3.5 Other settings	21
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	22
5.4.4 Line number style	22
5.4.5 Skipping and hiding number	22
5.5 Executing code at each line	23
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	28
6.3.1 Basic use	28

6.3.2 Notes about input encoding with UTF-8 processor	28
6.3.3 Use with <code>\lemma</code> command	29
6.3.4 Customizing	30
6.4 Apparatus of Manuscripts	31
6.4.1 Marking sections of text	31
6.4.2 Layout of the apparatus of manuscripts	32
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	33
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	35
7.2 Control line number printing	36
7.2.1 Print line number only at first time	36
7.2.2 Print page number only at first time	36
7.2.3 Arbitrary text before line number	37
7.2.4 Separator for line range	37
7.2.5 Abbreviate line range	37
7.2.6 Disable line number	38
7.2.7 Printing pstart number	38
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	39
7.2.11 Space around number	39
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	41
7.5.1 For footnotes	41
7.5.2 For endnotes	41
7.6 Font style	42
7.6.1 For line number	42
7.6.2 For the lemma	42
7.6.3 For all notes	42
7.7 Wrapping notes	42
7.7.1 Wrapping lemmas	42
7.7.2 Wrapping contents	43

7.8 Indent of notes content	43
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	44
7.11.1 Alignment	44
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	45
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	47
7.14 Footnotes and the <code>reledpar</code> 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	49
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	51
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	53
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	54
11.5 Normal \LaTeX 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	55
11.6.3 Settings	55
11.7 Compatibility with <code>xr</code> package	57
12 Side notes	57
12.1 Basics	57
12.2 Setting	57
12.2.1 Width	57
12.2.2 Vertical position	58
12.2.3 Distance to the main text	58
12.2.4 Font	58
12.2.5 Separator between notes	58
13 Indexing	58
13.1 Basics	58
13.2 Referring to critical notes	59
13.3 Separator between page and line numbers	59
13.4 Using <code>xindy</code>	59
13.5 Advanced setting	60
14 Glossary	60
14.1 Preamble setting	60
14.2 Commands	61
15 Tabular material	61
16 Sectioning commands	64
16.1 Sectioning commands without line numbers or critical notes	64
16.2 Sectioning commands with line numbering and critical notes	64
16.3 Optimization	65
17 Quotation environments	65
18 Page breaks	66
18.1 Control page breaking	66
18.2 Prevent page break in a long verses	66
19 Miscellaneous	66
19.1 Known and suspected limitations	67
19.1.1 Non-standard geometry	67
19.1.2 <code>floatrow</code> package compatibility	67
19.1.3 ‘No room for a new’	67
19.1.4 Marginal notes	67
19.1.5 Paragraph shape	68
19.1.6 Paragraphed footnotes	68
19.1.7 Use with other packages	68
19.2 Parallel typesetting	69

I Implementation overview	71
II Preliminaries	71
II.1 Links with original edmac	71
II.2 Package declaration	71
II.3 Package options	72
II.4 Loading packages	74
II.5 Compatibility with LuaTeX	74
II.6 Boolean flags	74
II.7 Messages	75
II.8 Gobbling	82
II.9 Miscellaneous commands	82
II.10 Prepare reledpar	82
II.11 Booleans provided by other optional packages which are required in any case	83
III Sectioning commands	83
IV List macros	87
V Line counting	89
V.1 Choosing the system of lineation	89
V.2 Line number margin	90
V.3 Line number initialization and increment	91
V.4 Line number locking	93
V.5 Line number style	94
V.6 Line number printing	94
V.7 Line number counters and lists	95
V.8 Line number locking counter	96
V.9 Line number associated to lemma	96
V.10 Reading the line-list file	100
V.11 Commands within the line-list file	102
V.12 Writing to the line-list file	115
VI Marking text for notes	121
VI.1 \edtext itself	122
VI.2 Substitute lemma	129
VI.3 Substitute line numbers	130
VI.4 Lemma disambiguation	131
VII Paragraph decomposition and reassembly	137
VII.1 Boxes, counters, \pstart and \pend	137
VII.2 Processing one line	142
VII.2.1 General process	142
VII.2.2 Process for “normal” line	143
VII.2.3 Process for line containing \eledsection command	145
VII.2.4 Hooks	145

VII.2.5 Sidenotes and marginal line number initialization	146
VIII Line and page number computation	146
IX Line number printing	150
X Pstart number printing in side	154
XI Restoring footnotes and penalties	155
XI.1 Add insertions to the vertical list	155
XI.2 Penalties	157
XI.3 Printing leftover notes	158
XI.4 Text before notes	159
XII Critical footnotes	159
XII.1 Fonts	160
XII.2 Individual note options	160
XII.3 Notes language	161
XII.4 General survey of the way we manage notes	162
XII.5 General setup	162
XII.6 Footnotes arrangement	163
XII.6.1 User level macro	163
XII.6.2 Normal footnote	163
XII.6.3 Paragraphed footnotes	168
XII.6.4 Columnar footnotes	175
XII.7 Critical notes presentation	182
XII.7.1 Font tools	182
XII.7.2 Pstart number in footnote	183
XII.7.3 Lemma printing	184
XII.7.4 Line number printing	184
XII.7.5 Footnote grouped by line	193
XIII Familiar footnotes	194
XIII.1 Adjacent footnotes	194
XIII.2 Regular footnotes for numbered texts	196
XIII.3 Footnote formats	198
XIII.4 Footnote arrangement	198
XIII.4.1 User level macro	198
XIII.4.2 Normal footnotes	199
XIII.4.3 Two columns footnotes	204
XIII.4.4 Three columns footnotes	207
XIII.4.5 Paragraphed footnotes	209
XIII.5 Wrapping footnote marks in hyperlink	213
XIV Code common to both critical and familiar footnote in normal arrangement	214

XV Footnotes' width for two columns	214
XVI Footnotes' order	216
XVII Footnotes' rule	216
XVIII Specific skip for first series of footnotes	217
XVIII.0.1 Overview	217
XVIII.0.2 User level command	217
XVIII.0.3 Internal commands	218
XIX Endnotes	219
XIX.1 Internal commands	219
XIX.2 User level commands	223
XIX.2.1 Inserting contents to endnotes	223
XIX.2.2 Printing endnotes	223
XX Generate series of notes	230
XX.1 Test if series is still existing	230
XX.2 Init specific to reledpar	230
XX.3 For critical footnotes	231
XX.3.1 Options	231
XX.3.2 Create inserts, needed to add notes in foot	232
XX.3.3 Create commands for critical apparatus, \Afootnote, \Bfootnote etc.	232
XX.3.4 Set standard display	235
XX.4 For familiar footnotes	235
XX.4.1 Options	236
XX.4.2 Create tools for familiar footnotes (\footnoteX)	236
XX.5 The endnotes	238
XX.5.1 The auxiliary file	238
XX.5.2 The main macro	238
XX.5.3 Tools	239
XX.5.4 Internal commands	239
XX.5.5 The options	240
XX.6 Init standards series (A,B,C,D,E)	241
XXI Setting series display	241
XXI.1 Change series order	241
XXI.2 Test series order	242
XXI.2.1 Get the first series	242
XXI.3 Series setting	242
XXI.3.1 General way of working	242
XXI.3.2 Tools to set options	243
XXI.3.3 Tools to generate options commands	244
XXI.3.4 Options for critical notes	246
XXI.3.5 Options for familiar notes	247

XXI.3.6 Options for endnotes	248
XXI.4 Hooks for a particular footnote	250
XXI.5 Alias	251
XXII Output routine	251
XXII.1 Extra footnotes output	251
XXII.2 Patching standard output's commands	254
XXIII Cross referencing	257
XXIII.1 Compatibility with xref	271
XXIV Side notes	272
XXV Minipages and such	280
XXVI Indexing	285
XXVI.1 Looking on package order	285
XXVI.2 Auxiliary macros for \edindex	285
XXVI.3 Code specific to \edindexin critical footnotes	286
XXVI.4 Analysis of command in indexed text	288
XXVI.5 Code for the formatted index	288
XXVI.6 Main code	289
XXVI.7 Hyperlink	290
XXVI.8 'innote' and 'notenumber' option of indextols package	293
XXVII Glossaries	293
XXVIII Verse	295
XXVIII.1 Hanging symbol management	296
XXVIII.2 Using & character	296
XXVIII.3 Code category setting	296
XXVIII.4 Stanza count and indent	297
XXVIII.5 Numbering stanza	298
XXVIII.6 Stanza number in note	299
XXVIII.7 Main work	300
XXVIII.8 Restore catcode and penalties	302
XXIX Apparatus of Manuscripts	302
XXIX.1 User level macro	302
XXIX.2 Setting macro	304
XXIX.3 Counters and lists	304
XXIX.4 Auxiliary file macros	305
XXIX.5 Action macro	306
XXIX.6 Inserting footnote	311
XXIX.7 Other	311

XXX Arrays and tables	311
XXX.1 Preamble: macro as environment	311
XXX.2 Tabular environments	315
XXX.2.1 Disabling and restoring commands	315
XXX.2.2 Counters, boxes and lengths	318
XXX.2.3 Tabular typesetting	322
XXX.2.4 Environments	333
XXXI Quotation's commands	333
XXXII Section's title commands	335
XXXII.1 Commands to disable some feature	335
XXXII.2 General overview	335
XXXII.3 <code>\beforeeledchapter</code> command	336
XXXII.4 Auxiliary commands	336
XXXII.5 Patching standard commands	337
XXXII.6 Main code of <code>\eledxxx</code> commands	342
XXXII.7 Macros written in the auxiliary file	344
XXXIII Page breaking or no page breaking depending of specific lines	347
XXXIV Long verse: prevents being separated by a page break	348
XXXV Tools for hyperref package	349
XXXVI Compatibility with <code>eledmac</code>	350
Appendix A Things to do when changing versions	352
Appendix A.1 Migrating from <code>edmac</code> to <code>ledmac</code>	352
Appendix A.2 Migration from <code>ledmac</code> to <code>eledmac</code>	353
Appendix A.3 Migration to <code>eledmac</code> 1.5.1	354
Appendix A.4 Migration to <code>eledmac</code> 1.12.0	354
Appendix A.5 Migration to <code>eledmac</code> 17.1	355
Appendix A.6 Migration to <code>eledmac</code> 1.21.0	355
Appendix A.6.1 <code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuff</code>	355
Appendix A.6.2 Endnotes	355
Appendix A.7 Migration to <code>eledmac</code> 1.22.0	355
Appendix A.8 Migration to <code>eledmac</code> 1.23.0	355
Appendix A.9 Migration from <code>eledmac</code> to <code>reledmac</code>	356
Appendix A.9.1 Risk of 'no room for a new'	356
Appendix A.9.2 Multiple indices with memoir	356
Appendix A.9.3 Deprecated commands and options	356
Appendix A.9.4 <code>\renewcommand</code> replaced by <code>command</code>	357
Appendix A.9.5 Commands the names of which have been changed	357
Appendix A.9.6 Endnotes	359
Appendix A.9.7 Z Series	359
Appendix A.9.8 Internal commands	359

Appendix A.10 Migration to <code>reledmac</code> 2.1.0	359
Appendix A.11 Migration to <code>reledmac</code> 2.1.3	359
Appendix A.12 Migration to <code>reledmac</code> 2.3.0	359
Appendix A.13 Migration to <code>reledmac</code> 2.4.0	360
Appendix A.14 Migration to <code>reledmac</code> 2.5.0	360
Appendix A.15 Migration to <code>reledmac</code> 2.7.0	360
Appendix A.16 Migration to <code>reledmac</code> 2.7.2	360
Appendix A.17 Migration to <code>reledmac</code> 2.8.0	360
Appendix A.18 Migration to <code>reledmac</code> 2.13.1	360
Appendix A.19 Migration to <code>reledmac</code> 2.18.0	361
References	362
Index	362
Change History	410

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.

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).

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

1.2.2 ledmac

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

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

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

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

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

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

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

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

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

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

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

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

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

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

¹²Being prepared at Poona and Lausanne Universities.

1.2.3 eledmac

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

To migrate from ledmac to eledmac, please read Appendix A.2 p. 353.

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. 356.

1.3 Bibliography

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

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

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

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

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

3 Compatibility warning

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

Please read 7.13.6 p. 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. 357).

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` `\pend` Within a numbered section, each paragraph of numbered text must be marked using the `\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 19.2 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` It is possible to insert a number at every `\pstart` command; you must use the
`\numberpstartfalse` `\numberpstarttrue` command to have it. You can stop the numbering with `\numberpstartfalse`.
`\thepstart` You can redefine the command `\thepstart` to change style. You can change the value
of the `pstart` number by using *after* `\beginnumbering`:

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

On each `\beginnumbering` the numbering restarts.

`\sidepstartnumtrue` With the `\sidepstartnumtrue` command, the number of `\pstart` will be printed

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

inside. In this case, the line number will be not printed.

`\labelstarttrue`

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

5.2.6 Languages written in Right to Left

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

5.2.7 Memory limits

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

`\pausenumbering`
`\resumenumbering`

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

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

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

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

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

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

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

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

5.3 Lineation commands

5.3.1 Disabling lineation

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

5.3.2 Setting lineation start and step

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

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

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

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

`\linenumberlist`

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

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

`\gdef\linenumberlist{}`

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

5.3.3 Setting lineation reset

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

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

5.3.4 Setting line number margin

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

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

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

5.3.5 Other settings

`\leftlinenum`
`\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`
`\sublinenumberstyle` Line numbers are normally printed as arabic numbers. You can use `\linenumberstyle{<style>}` 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` `\dolinehook` 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:

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

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

6.2.2 Footnotes

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

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

If you need more series of critical notes, please look at 6.6.1 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.
- `linerangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

6.2.3 Endnotes

`\Aendnote` **Inserting endnotes** The package also maintains five separate series of endnotes.
`\Bendnote`
`\Cendnote`
`\Dendnote`
`\Eendnote`

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

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

`\Aendnote[⟨option⟩]{⟨text⟩}`.

`⟨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`
`Xtoendnotes`

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

For example:

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

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

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

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

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

6.2.4 Paragraph in critical apparatus

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

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

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

6.2.5 Change lemma and line number

`\lemma`

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

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.

```

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

```

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

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

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

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

```
\newcommand{\variant}[2][1,usedefault]{\Afootnote[#1]{#2}}
```

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`.

```

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

```

6.3 Disambiguation of identical words in the apparatus

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

6.3.1 Basic use

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

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

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

6.3.2 Notes about input encoding with UTF-8 processor

If you use UTF-8 processor, like $\text{Xe}\text{L}\text{A}\text{T}\text{E}\text{X}$ or $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$, 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 $\text{Xe}\text{L}\text{A}\text{T}\text{E}\text{X}$, add this line in your preamble: `\XeTeXinputnormalization 1`.
- If you use $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$, use the `uninormalize` package of Michal Hoftich¹⁶ with the `buffer` option set to true.

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

With these tools, Xe₂TeX / LuaTeX will dynamically normalize unicode input when reading the file. Consequently, you will have no problems with the `\sameword` command.

6.3.3 Use with `\lemma` command

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

For example in the following example:

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

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

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

- In the content of `\lemma`, use `\sameword` with no optional argument.
- In the first argument of `\edtext`, use `\sameword` with the optional argument `[⟨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.

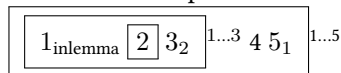
Note also that the `⟨X⟩` does not refer to the level where the `\sameword` occurs, but to the level of the `\lemma` that refers to that `\sameword`. For example:

```
\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

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

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

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

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

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

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

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

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

6.4.2 Layout of the apparatus of manuscripts

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

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

6.4.3 Settings

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

<code>\setmsdataseries</code>	The series used by default for the apparatus of manuscripts is series A. However, you can change it with <code>\setmsdataseries{<series>}</code> .
<code>\setmsdatalabel</code>	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 <code>\setmsdatalabel{<txt>}</code> .
<code>\setmsdataposition</code>	If you want the manuscript apparatus to be on the same level of critical footnotes as the other apparatuses, for each line, reledmac will first insert the manuscript apparatus, then the other footnotes. You can change it using: <code>\msdataposition{regular-msdata}</code> And restore the default behaviour using <code>\msdataposition{msdata-regular}</code>

6.5 Familiar notes

6.5.1 Basic use

<code>\footnoteA</code>	As well as the standard \LaTeX footnotes generated via <code>\footnote</code> , the package also provides five series of additional footnotes called <code>\footnoteA</code> through <code>\footnoteE</code> . 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.
<code>\footnoteB</code>	
<code>\footnoteC</code>	
<code>\footnoteD</code>	
<code>\footnoteE</code>	

6.5.2 Customizing mark

<code>\thefootnoteA</code>	Each series uses a set of macros for styling the marks. The mark numbering scheme of series A is defined by the <code>\thefootnoteA</code> macro; the default is:
<code>\bodyfootmarkA</code>	
<code>\footfootmarkA</code>	
	<code>\renewcommand*{\thefootnoteA}{\arabic{footnoteA}}</code>
	The appearance of the mark in the text is controlled by <code>\bodyfootmarkA</code> 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` `\seriesatend` However in some specific cases, you need to change the series order at some point inside the document. You can use `\seriesatbegin{<s>}` to pull up a given series `<s>` to the beginning, or `\seriesatend{<s>}` to push it down to the end.

6.7 Position of critical and familiar footnotes

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

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

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

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

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

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

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

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

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

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

7 Critical apparatus appearance

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

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

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

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

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

There is also name convention:

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

7.1 Notes arrangement in a series

`\Xarrangement`
`\arrangementX`

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

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

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

- `paragraph` formats all of the footnotes of a series as a single paragraph; if you use this arrangement, you are strongly encouraged to read 19.1.6 p. 68.
- `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

<code>\Xnumberonlyfirstinline</code>	. 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 <code>\Xnumberonlyfirstinline[⟨s⟩]</code> .
<code>\Xnumberonlyfirstintwolines</code>	Suppose you have a lemma on line 2 and a lemma between line 2 and line 3. With <code>\Xnumberonlyfirstinline</code> , the second lemma is considered to be on the same line as the first lemma. But if you use both <code>\Xnumberonlyfirstinline[⟨s⟩]</code> and <code>\Xnumberonlyfirstintwolines[⟨s⟩]</code> , a distinction is made.
<code>\Xsymlinenum</code>	For setting a particular symbol in place of the line number, you can use <code>\Xsymlinenum[⟨s⟩]{⟨symbol⟩}</code> in combination with <code>\Xnumberonlyfirstinline[⟨s⟩]</code> . From the second lemma of the same line, the symbol will be used instead of the line number. Note that any command called in <code>⟨symbol⟩</code> must be robust. Use <code>\robustify</code> to robustify a non-robust command.
<code>\Xendnumberonlyfirstinline</code> <code>\Xendnumberonlyfirstintwolines</code> <code>\Xendsymlinenum</code>	For endnotes, <code>\Xendnumberonlyfirstinline</code> ; <code>\Xendnumberonlyfirstintwolines</code> and <code>\Xendsymlinenum</code> are the equivalents of <code>\Xnumberonlyfirstinline</code> ; <code>\Xnumberonlyfirstintwolines</code> and <code>\Xsymlinenum</code> .

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

`\Xendinplaceofpagenumber` • `\Xendinplaceofpagenumber[$\langle series \rangle$]{ $\langle l \rangle$ }` to print a $\langle l \rangle$ length horizontal space in case no symbol is printed instead of the page number.

7.2.3 Arbitrary text before line number

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

7.2.4 Separator for line range

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

7.2.5 Abbreviate line range

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

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

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

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

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

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

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

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

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

`\Xendtwolines` For endnotes, use these macros: `\Xendtwolines`; `\Xendmorethantwolines`;
`\Xendmorethantwolines` `\Xendtwolinesbutnotmore`;
`\Xendtwolinesbutnotmore`

`\Xendtwolinesonlyinsamepage` instead of `\Xtwolines`; `\Xmorethantwolines`;
`\Xtwolinesbutnotmore`; `\Xtwolinesonlyinsamepage`.

7.2.6 Disable line number

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

7.2.7 Printing pstart number

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

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

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

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

7.2.8 Printing stanza number

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

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

`\Xstanzaseparator`

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

7.2.9 Separator between line and subline numbers

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

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

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

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

7.2.10 Separator between page and line numbers

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

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

7.2.12 Space around line symbol

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

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

`\Xendbeforemsymlinum` `\Xendbeforemsymlinum` and `\Xendaftersymlinum` are the equivalents of `\Xbeforemsymlinum` and `\Xaftersymlinum` for the endnotes.

7.2.13 Space in place of number

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

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

7.2.14 Boxing line number and line symbol

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

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

`\Xboxmsymlinum` `\Xboxmsymlinum[⟨s⟩]{⟨l⟩}` is the same as `\Xboxlinenum` but for the line number symbol.

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

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

L to align left (default value);

R to align right;

C to center.

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

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

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

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

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

7.3 For endnotes

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

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

7.4 Arbitrary code around line number

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

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

7.5 Separator between the lemma and the note

7.5.1 For footnotes

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

7.5.2 For endnotes

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

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

7.6 Font style

7.6.1 For line number

`\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

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

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

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

`\Xlemmafont{\bfseries}`

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

7.6.3 For all notes

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

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

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

7.7 Wrapping notes

7.7.1 Wrapping lemmas

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

mand. 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 \TeX command.

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

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

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

7.8 Indent of notes content

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

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

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

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

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

7.9 Arbitrary code at the beginning of notes

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

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

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

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

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

7.10 Arbitrary code before inserting note

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

`beforeinsertingX`

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

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

`\Xhsizethreecol` `\Xhsizethreecol[⟨s⟩]{⟨l⟩}` is used to change width of a column when critical notes are displaying in three columns. Default value is `.3 \hsizel`.

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

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

7.12 Options for paragraphed footnotes and notes grouped by line

7.12.1 Mark separation of notes

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

critical footnotes) or `\afternoteX[⟨s⟩]{⟨l⟩}` (for familiar footnotes). The default value is 1em plus .4em minus .4em.

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

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

7.12.2 Ragged text

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

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

7.13 Options for block of notes

7.13.1 Grouping notes by line

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

In many cases, you might like to use it in combination with `\Xnumberonlyfirstinline` (7.2.1 p. 36).

`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. 36).

7.13.2 Text before notes

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

7.13.3 Code before notes

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

7.13.4 Spacing

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

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

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

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

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

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

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

7.13.5 Rule

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

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

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

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

7.13.6 Maximum height

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

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

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

Be careful with the two previous commands. Actually, for technical purposes, one paragraphed note is considered as one block. Consequently, it cannot be broken between two pages, even if you used these commands. The debug is in the `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 width of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.11.2 p. 44.

7.14 Footnotes and the `reledpar` columns

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

`\noteswidthliketwocolumnsX`

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

7.15 Endnotes in one paragraph

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

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

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

8 Fonts

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

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

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

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

`\select@lemmafонт` We will briefly discuss `\select@lemmafонт` here because it is important to know

about it now, although it is not one of the macros you would expect to change in the course of a simple job. Hence it is ‘protected’ by having the @-sign in its name.

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

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

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

9 Verse

9.1 Basic

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

9.2 Define stanza indents

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

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

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

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

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

If you want the hanging verse to be flush right, you can use `\sethanginsymbol`: see p. 9.6 p. 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 `stanzaindentsrepetition` counter at n . For example:

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

is like

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

Be careful: the feature is changed in eledmac 1.5.1. See Appendix A.3 p. 354.

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

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

```
\setcounter{stanzaindentsrepetition}{0}
```

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

9.4 Manual stanza indent

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

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. 66 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` 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` envi-

ronment are typeset at the end of the environment, as with minipages, but the environment includes normal page breaks. The environment makes no change to the textwidth so it appears as normal text; it just might be that footnotes appear in the middle of a page, with text above and below.

`ledgroupsize`

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

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

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

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

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

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

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

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

11 Cross referencing

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

11.1 Basic use

`\edlabel`

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

`\edpageref`

`\edlineref`

`\sublineref`

`\pstartref`

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

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

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

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

You will be warned if you use `\edlabel{foo}` and `foo` has been used as a label before. The `ref` commands will return references to the last place in the file marked with

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

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. 55).

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. 27).

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

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

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

`\xxref`

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

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

Here, an example of use:

```
\beginnumbering

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

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

\endnumbering
```

11.4 Not automatic cross-referencing

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

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

11.5 Normal L^AT_EX 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.

<code>\edlabelS</code>	Use <code>\edlabelS{<label>}</code> to mark the start line of the passage.
<code>\edlabelE</code>	Use <code>\edlabelE{<label>}</code> to mark the end the end line of the passage. These two commands just create to label which are named <code><label>:start</code> and <code><label>:end</code> .
<code>\edlabelSE</code>	Use <code>\edlabelSE{<label>}</code> to mark just one location in the text. Contrary to a classical <code>\edlabel</code> , the <code><label></code> could be use with <code>\Seref</code> and <code>\Serefwithpage</code> .
<code>\Seref</code>	The main utility is to use them with three other commands. <code>\Seref{<label>}</code> will make a cross-reference printed as a reference in critical footnotes.
<code>\Serefwithpage</code>	<code>\Serefwithpage</code> will make a cross-reference printed as a reference in critical end-notes.
<code>\Serefonlypage</code>	<code>\Serefonlypage</code> will make a cross-reference printed only with page number.

11.6.2 References to lines that are commented on in the apparatus

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

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

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

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

11.6.3 Settings

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

For example, you may use:

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

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

`\setSerefprefixsingle` `\setSerefprefixmore` `\setSerefonlypageprefixsingle` `\setSerefonlypageprefixmore` `\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`
- `\Xendlinerangeseparator`
- `\Xendmorethantwolines`
- `\Xendsublinesep`
- `\Xendtwolines`
- `\Xendtwolinesbutnotmore`
- `\Xendtwolinesonlyinsamepage`

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

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

11.7 Compatibility with xr package

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

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

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

12 Side notes

12.1 Basics

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

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

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

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

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

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

12.2 Setting

12.2.1 Width

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

12.2.2 Vertical position

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

12.2.3 Distance to the main text

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

12.2.4 Font

`\ledlsnotefontsetup` These macros specify how the sidenote texts are to be typeset. The initial definitions are:
`\ledrsnotefontsetup`

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

These can of course be changed to suit.

12.2.5 Separator between notes

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

13 Indexing

13.1 Basics

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

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

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

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

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

1. Load `imakeidx` or `indextools`.
2. Load `reledmac`.

3. Declare the index with the macro `\makeindex` of `imakeidx` and `indextools`.

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

13.2 Referring to critical notes

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

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

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

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

13.3 Separator between page and line numbers

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

- is the default separator used by the MAKEINDEX program.

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

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

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

```
page_compositor ":"
```

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

13.4 Using xindy

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

```
pagenumber-linenum
```

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

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:

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

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

14 Glossary

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

14.1 Preamble setting

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

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`.

- 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. 59).

14.2 Commands

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

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

15 Tabular material

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

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

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

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

1	2	3
a	bb	ccc
AAA	BB	C

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

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

For example:

```
\beginnumbering
\pstart
\begin{edtabularl}
\textbf{\Large I} & wish I was a little bug\edindex{bug} &
\textbf{\Large I} & eat my peas with honey\edindex{honey} \\
& With whiskers \edtext{round}{\Afootnote{around}} my tummy &
& I've done it all my life. \\
& I'd climb into a honey\edindex{honey} pot &
```


$$\backslash newcommand*{\upbracketfill}{\%$$
$$\begin{array}{c} \end{array}$$

`\edatleft` `\edatleft[\langle math \rangle]{\langle symbol \rangle}{\langle halfheight \rangle}` typesets the math $\langle symbol \rangle$ as $\left\{\langle symbol \rangle\right\}$

$$\begin{array}{c}$$
$$\begin{array}{ccccccc} & \& 1 & \& 2 & \& 3 & \& \\ & & & & & & & \backslash \\ & & & & & & & \end{array}$$

edbeforetab `\edbeforetab{text}{entry}`, where *entry* is an entry in the leftmost column,

`\edatertab` typesets $\langle text \rangle$ left justified before the $\langle entry \rangle$. Similarly `\edatertab{\langle entry \rangle}{\langle text \rangle}`,

$$\begin{array}{l} \end{array}$$
$$\frac{A \& 1 \& 2 \& 5}{\backslash \text{edgefromtab}[Refers][P] \& 1 \& 2 \& 6}$$
$$\end{edarray}$$

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

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

`\edvertdots`

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

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

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

16 Sectioning commands

16.1 Sectioning commands without line numbers or critical notes

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

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

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

16.2 Sectioning commands with line numbering and critical notes

You have to use the following commands:

- `\eledchapter[⟨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.

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. $n + 1$.

`\ledpbsetting`

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

18.2 Prevent page break in a long verses

`\lednopbinversettrue`

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

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

19 Miscellaneous

`\extensionchars`

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

`\ifledfinal`

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

`\showlemma`

The lemma within the text is printed via `\showlemma{lemma}`. Normally, or with the ‘final’ option, the definition of `\showlemma` is:

```
\newcommand*{\showlemma}[1]{#1}
```

so it just produces its argument. With the ‘draft’ option it is defined as

```
\newcommand*{\showlemma}[1]{\textit{#1}}
```

so that its argument is typeset in an italic font, which may make it easier to check that all lemmas have been treated.

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

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

19.1 Known and suspected limitations

19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.13.6 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` `\TeX` is a three-pass system, but even after a document has been processed three times, there are some tricky situations in which the page breaks decided by `\TeX` never settle down. At each successive run, `reledmac` may oscillate between two different sets of page decisions. To stop this happening, should it arise, Wayne Sullivan suggested the inclusion of the quantity `\ballast`. The amount of `\ballast` will be subtracted from the penalties which apply to the page breaks calculated on the *previous* run through `\TeX`, thus reinforcing these breaks. So if you find your page breaks oscillating, insert `\setcounter{ballast}{100}` or some such figure, and with any luck the page breaks will settle down. Luckily, this problem does not crop up at all often.

19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.1 p. 35, and described in more detail on XII.6.3 p. 171, 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{...}}}
```

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

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

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

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

19.2 Parallel typesetting

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

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

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

I Implementation overview

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

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

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

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

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

II Preliminaries

II.1 Links with original `edmac`

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

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

II.2 Package declaration

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

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledmac}[2017/03/20 v2.18.1 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 $\text{Lua}\text{\TeX}$ or $\text{X}\text{\TeX}$ 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 $\text{Lua}\text{\TeX}$

Here, we enable some primitives for $\text{Lua}\text{\TeX}$.

```

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 TeX's memory when the same message is repeated and also lets them be edited easily.

\reledmac@warning Write a warning message.

```

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

```

`\reledmac@error` Write an error message.

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

```
\led@err@NumberingStarted30 \newcommand*\led@err@NumberingStarted{%
\led@err@NumberingNotStarted31 \reledmac@error{Numbering has already been started}{\@ehc}}
\led@err@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 %
```

```
\led@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 %
```

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

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

```

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

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

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

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

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

```

```

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

```

```

\led@warn@DuplicateLabel204 \newcommand*\led@warn@DuplicateLabel}[1]{%
\led@warn@AppLabelOutSecondArgEdtext205 \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined206 \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined207 }%
208 \newcommand*\led@warn@AppLabelOutSecondArgEdtext}[1]{%
209 \reledmac@warning{\string\applabel\space outside of the second argument
of an \string\edtext\space `#1' on page \thepage.}}%
210 \newcommand*\led@warn@RefUndefined}[1]{%
211 \G@refundefinedtrue%
212 \reledmac@warning{Reference `#1' on page \thepage\space undefined.%
213 Using `000'.}%
214 \@latex@warning{Reference `#1' undefined\on@line}%
215 }%
216 \newcommand*\led@warn@pairRefUndefined}[1]{%
217 \G@refundefinedtrue%
218 \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \thepage\
space undefined.
219 Using `??'.}%
220 \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
221 }
222 %

```

```

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

```

```

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

```

```

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

```

```

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

```

```

\led@err@BadAction36 \newcommand*{\led@err@StanzaIndentNotDefined}{%
237 \reledmac@error{You have not defined the indentation for the line \number
\stanza@count}{\@ehc}}%
238 %

\led@err@ManySidenotes39 \newcommand{\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes40 \ifledRcol{%
\led@err@ManyRightnotes41 \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@TooManyColumns61 \newcommand*{\led@err@TooManyColumns}{%
\led@err@UnequalColumns62 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn63 \newcommand*{\led@err@UnequalColumns}{%
\led@err@HighEndColumn64 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns65 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 %

```

```

\led@err@toendnotes@outsidenumbering73 \newcommand{\led@err@toendnotes@outsidenumbering}{%
274   \reledmac@error{\string\toendnotes\space and related commands must be
      called inside a numbered texte (\string\...beginnumbering\string\endnumbering
      )}{\@ehc}%
275 }%
276 %

```

```

\led@err@EdtextWithoutFootnote77 \newcommand{\led@err@EdtextWithoutFootnote}{%
278   \reledmac@error{edtext without Xfootnote. Check syntaxis}{\@ehc}%
279 }%
280 %

```

```

\led@err@FootnoteNotInSecondArgEdtext81 \newcommand{\led@err@FootnoteNotInSecondArgEdtext}[1]{%
282   \reledmac@error{#1footnote outside of the second argument of an edtext.
      Check syntax}{\@ehc}%
283 }%
284 %

```

```

\led@error@PackageAfterEledmac85 \newcommand{\led@error@PackageAfterEledmac}[1]{%
286   \reledmac@error{#1 must be loaded before reledmac}{\@ehc}%
287 }%
288 %

```

```

\led@error@fail@patch@@makecol89 \newcommand{\led@error@fail@patch@@makecol}{%
290   \reledmac@error{Fail to patch \string\@makecol\space command}{\@ehc}%
291 }%
292 %

```

```

\led@error@fail@patch@@reinserts93 \newcommand{\led@error@fail@patch@@reinserts}{%
294   \reledmac@error{Fail to patch \string\@reinserts\space command}{\@ehc}%
295 }%
296 %

```

```

\led@error@fail@patch@@docclearpage97 \newcommand{\led@error@fail@patch@@docclearpage}{%
298   \reledmac@error{Fail to patch \string\@docclearpage\space command}{\@ehc}%
299 }%
300 %

```

```

\led@error@fail@patch@@iiiminipage01 \newcommand{\led@error@fail@patch@@iiiminipage}{%
302   \reledmac@error{Fail to patch \string\@iiiminipage\space command}{\@ehc}%
303 }%
304 %

```



```

or@fail@patch@endminipage 305 \newcommand{\led@error@fail@patch@endminipage}{%
306 \reledmac@error{Failed to patch the \string\endminipage\space command}{\
@ehc}%
307 }%
308 %

```

```

or@fail@patch@endminipage 309 \newcommand{\led@error@fail@patch@makeindex}{%
310 \reledmac@error{Failed to patch the \string\makeindex\space command}{\
@ehc}%
311 }%
312 %

```

```

n@edinde@outsidenumbering 313 \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 %

```

```

warning@hsizeX@deprecated 317 \newcommand{\led@warning@hsizeX@deprecated}{%
318 \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
319 }%
320 %

```

```

warning@Xhsize@deprecated 321 \newcommand{\led@warning@Xhsize@deprecated}{%
322 \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
323 }%
324 %

```

```

warning@msdatawithoutstop 325 \newcommand{\led@warning@msdatawithoutstop}{%
326 \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
327 }%
328 %

```

```

ning@preXnotes@deprecated 329 \newcommand{\led@warning@preXnotes@deprecated}{%
330 \reledmac@warning@preXnotes@deprecated%
331 }%
332 %

```

II.8 Gobbling

Here, we define some commands which gobble their arguments.

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

II.9 Miscellaneous commands

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

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

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

II.10 Prepare reledpar

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

```
\ifl@dprintingcolumns349 \newif\ifl@dpairing
\ifpst@rtedL350 \newif\ifl@dpaging%
\l@dnumpstartsL351 \newif\ifl@dprintingpages%
352 \newif\ifl@dprintingcolumns%
353 \newif\ifpst@rtedL
```

```

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

```

`\endnumbering` `\endnumbering` must follow the last text for a numbered section. It takes care of notifying you when changes have been noted in the input that require running the file through again to move everything to the right place.

```

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

```

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

`\resumenumbering`

```

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

```

The `\resumenumbering` macro is a bit more involved, but not much. It does most of the same things as `\beginnumbering`, but without resetting the various counters. Note

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

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

```

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

```

IV List macros

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

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

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

one operation. They also expands the items add to the list.

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

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

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

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

\list@clear The `\list@clear` macro just initializes a list to the empty list; it is no different from `\list@create` in its effect, but it is in its semantic .

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

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

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

\xleft@appenditem `\xleft@appenditem` expands an item and appends it to the left end of a list macro; it is otherwise identical to `\xright@appenditem`.

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

\gl@p The `\gl@p` macro removes the leftmost item from a list and places it in a control sequence. You type `\gl@p\l\to\z` (where `\l` is the list macro, and `\z` receives the left item). `\l` is assumed nonempty: use `\ifx\l\empty` to test for an empty `\l`. The control sequences created by `\gl@p` are all global.


```

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

```

V Line counting

V.1 Choosing the system of lineation

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

If the argument is page.

```

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

```

If the argument is pstart.

```

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

```

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

```

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

```

In other case, it is an error.

```

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

```

V.2 Line number margin

`\linenummargin` `\linenummargin{<word>}` specify which margin line numbers are in; it takes one argument, a string, which value can be left ; right; inner or outer.

`\line@margin` The selection is recorded in the count `\line@margin`: 0 for left, 1 for right, 2 for outer, and 3 for inner.

`\ldgetline@margin`

```

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

```

```

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

```

V.3 Line number initialization and increment

`\c@firstlinenum` 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; `\c@linenumincrement` `linenumincrement` is the difference between successive numbered lines. The initial values of these counters produce labels on lines 5, 10, 15, etc. `linenumincrement` must be at least 1.

```

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

```

```

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

```

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

```

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

```

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

```

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

```

655 %

V.4 Line number locking

`\lockdisp` When line locking is being used, the `\lockdisp{⟨word⟩}` macro specifies whether a line number—if one is due to appear—should be printed on the first printed line or on the last, or by all of them. Its argument is a word, either `first`, `last`, or `all`. Initially, it is set to `first`.

`\lock@disp` encodes the selection: 0 for first, 1 for last, 2 for all.

```

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

```

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

```

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

```

```
691
692 %
```

V.5 Line number style

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

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

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

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

`\sublinenumrep`

`\sublinenumr@p`

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

Initialise the number styles to arabic.

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

V.6 Line number printing

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

`\rightlinenum`

`\linenumsep`

`\numlabfont`

`\ledlinenum`

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

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

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

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

```

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

```

V.7 Line number counters and lists

Footnote references using line numbers rather than symbols can't be generated in one pass, because we do not know the line numbers till we ship out the pages. It would be possible if footnotes were never keyed to more than one line; but some footnotes gloss passages that may run for several lines, and they must be tied to the first line of the passage glossed. And even one-line passages require two passes if we want line-per-page numbering rather than line-per-section numbering.

So we run \LaTeX over the text several times, and each time save information about page and line numbers in a 'line-list file' to be used during the next pass. At the start of each section—whenever `\beginnumbering` is executed—the line-list file for that section is read, and the information from it is encoded into a few list macros.

We need first to define the different line numbers that are involved in these macros, and the associated counters.

\line@num The count `\line@num` stores the line number that is used in marginal line numbering and in notes: counting either by section, page or pstart, depending on your choice for this section. This may be qualified by `\subline@num`.

```

731 \newcount\line@num
732 %

```

`\subline@num` The count `\subline@num` stores a sub-line number that qualifies `\line@num`. For example, line 10 might have sub-line numbers 1, 2 and 3, which might be printed as lines 10.1, 10.2, 10.3.

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

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

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

`\sublines@false`

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

`\absline@num` The count `\absline@num` stores the absolute number of lines since the start of the section: that is, the number we have actually printed, no matter what numbers we attached to them. This value is never printed on an output page, though `\line@num` will often be equal to it. It is used internally to keep track of where notes are to appear and where new pages start: using this value rather than `\line@num` is a lot simpler, because it does not depend on the lineation system in use.

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

We will call `\absline@num` numbers “absolute” numbers, and `\line@num` and `\subline@num` numbers “visible” numbers.

V.8 Line number locking counter

`\@lock` The counts `\@lock` and `\sub@lock` tell us the state of line-number and sub-line-number locking. 0 means we are not within a locked set of lines; 1 means we are at the first line in the set; 2, at some intermediate line; and 3, at the last line.

`\sub@lock`

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

V.9 Line number associated to lemma

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

`\insertlines@list`

`\actionlines@list`

`\actions@list`

- `\line@list`: the page and line numbers for every lemma marked by `\edtext`. There are seven pieces of information, separated by vertical bars:

1. the starting page,
2. line, and
3. sub-line numbers, followed by the
4. ending page,
5. line, and
6. sub-line numbers, and then the
7. font specifier for the lemma.

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

```
23|35|0|24|3|0|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:

```

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

`\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`

```

748 \newcount\page@num
749 \newcount\endpage@num
750 \newcount\endline@num
751 \newcount\endsubline@num
752 %

```

`\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.

```

753 \newif\ifnoteschanged@
754 %

```

`\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@55 \newcommand*\resetprevline@{%
756 \def\do##1{\global\csundef{prevline##1}}%
757 \dolistloop{\@series}%
758 }
759 %

```

`\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@50 \newcommand*\resetprevpage@num{%
761 \def\do##1{%
762 \ifcsdef{prevpage##1@num}{%
763 \global\csname prevpage##1@num\endcsname=\z@%
764 \global\csname prevpage##1@numR\endcsname=\z@%
765 }%
766 {}%
767 \ifcsdef{##1prevpage@num}{%

```

```

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

```

V.10 Reading the line-list file

`\read@linelist` `\read@linelist{⟨file⟩}` is the control sequence that is called by `\beginnumbering` (via `\line@list@stuff`) to open and process a line-list file; its argument is the name of the file. . First, it clear all previous line's list.

```

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

```

When using `reledpar`, make sure that the `\maxlinesinpar@list` is empty (otherwise things will be thrown out of kilter if there is any old stuff still hanging in there).

```

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

```

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

Our line, page, and line-locking counters were already zeroed by `\line@list@stuff` if this is being called from within `\beginnumbering`; sub-lineation will be turned off as well in that case. On the other hand, if this is being called from `\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.

```

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

```

```

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

```

When the reading is done, we are all through with the line-list file. All the information we needed from it will now be encoded in our list macros.

Finally, we initialize the `\next@actionline` and `\next@action` macros, which specify where and what the next action to be taken is.

```

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

```

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

```

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

```

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

```

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

```

This version of `\read@linelist` creates list macros containing data for the entire section, so they could get rather large. It would be no more difficult to read the line-list file incrementally rather than all at once: we could read, at the start of each paragraph, only the commands relating to that paragraph. But this would require that we have two line-lists open at once, one for reading, one for writing, and on systems without version numbers we would have to do some file renaming outside of \TeX for that to work. We have retained this slower approach to avoid that sort of hacking about, but have provided the `\pausenumbering` and `\resumenumbering` macros to help you if you run into macro memory limitations (see 5.2.7 p. 19 above).

V.11 Commands within the line-list file

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

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

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

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

```

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

```

```

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

```

\@nl \@nl does everything related to the start of a new line of numbered text.

\@nl@reg In order to get the \setlinenum to work Peter Wilson had to slip in some new code at the start of the macro, to get the timing of the actions correct. The problem was that his original naive implementation of \setlinenum had a unfortunate tendency to change the number of the last line of the *preceding* paragraph. The new code is sort of based on the page number handling and \setline. It seems that a lot of fiddling with the line number internals is required.

In November 2004 in order to accurately determine page numbers Peter Wilson added these to the macro. It is now:

`\@nl{<page counter number>}{<printed page number>}`

We do not (yet) use the printed number (i.e., the \thepage) but it may come in handy later. The macro \fix@page checks if a new page has started.

Exactly what \@nl does depends on whether right text is being processed. That's why many code is defined in \@nl@reg or \nl@regR.

```

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

```

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```



```

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

```

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

```

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

```

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

\sub@action \sub@action adds an entry to the action-code list to turn sub-lineation on or off, according to the current value of the \ifsublines@ flag.

```

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

```

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

\do@lockon

\do@lockonL

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

```

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

```

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

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

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

```

1226 \newcount\insert@count
1227 %

```

- #2, a sequence of other line-list-file commands, executed to determine the ending line-number. (This may also include other \@ref commands, corresponding to uses of \edtext within the first argument of another instance of \edtext.)

\dummy@ref When nesting of \@ref commands does occur, it is necessary to temporarily redefine \@ref within \@ref, so that we are only doing one of these at a time.

```

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

```

\@ref@reg The first thing \@ref (i.e. \@ref@reg) itself does is to add the specified number of items to the \insertlines@list list.

```

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

```



```

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

```

Next, process the second argument to determine the page and line numbers for the end of this lemma. We temporarily equate `\@ref` to a different macro that just executes its argument, so that nested `\@ref` commands are just skipped this time. Some other macros need to be temporarily redefined to suppress their action.

```

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

```

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

```

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

```

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

```

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

```

Decrease edtext level counter.

```

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

```

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

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

```

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

```

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

```

1276   #1%
1277   %

```

Now, we store the list of \@sw of this current \edtext as an element of the global list of list of \@sw for a \edtext depth.

```

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

```

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

```

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

```

V.12 Writing to the line-list file

We have now defined all the counters, lists, and commands involved in reading the line-list file at the start of a section. Now we will cover the commands that `reledmac` uses within the text of a section to write commands out to the line-list.

`\linenum@out` The file will be opened on output stream `\linenum@out`.

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

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

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

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

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

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

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

`\line@list@stuff` The `\line@list@stuff{<file>}` macro, which is called by `\beginnumbering`, performs all the line-list operations needed at the start of a section. Its argument is the name of the line-list file.

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

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

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

Now close the current output line-list file, if any, and open a new one. The first time we open a line-list file for output, we do it using `\immediate`, and clear the `\iffirst@linenum@out@` flag.

```

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

```

If we get here, then this is not the first line-list we have seen, so we do not open or close the files immediately.

```

1321 \if@minipage%
1322 \leavevmode%
1323 \fi%
1324 \closeout\linenum@out%
1325 \openout\linenum@out=\l@auxdir#1\relax%
1326 \write\linenum@out{\string\line@list@version{\this@line@list@version}}%
1327 %
1328 \ifl@dpaging%
1329 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}%
1330 %
1331 \fi%
1332 \fi}
1333 %

```

\new@line The `\new@line` macro sends the `\@nl` command to the line-list file, to mark the start of a new text line, and its page number.

```

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

```

```

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

```

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

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

\flag@end

```

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

```

```

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

```

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

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

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

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

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

```

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

```

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

```

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

```

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

\flag@end@later

```

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

```

\startsub \startsub and \endsub turn sub-lineation on and off, by writing appropriate instructions to the line-list file. When sub-lineation is in effect, the line number counter is frozen and the sub-line counter advances instead. If one of these commands appears in the middle of a line, it does not take effect until the next line; in other words, a line is counted as a line or sub-line depending on what it started out as, even if that changes in the middle.

We tinker with \lastskip because a command of either sort really needs to be attached to the last word preceding the change, not the first word that follows the change. This is because sub-lineation will often turn on and off in mid-line—stage directions, for example, often are mixed with dialogue in that way—and when a line is mixed we want to label it using the system that was in effect at its start. But when sub-lineation begins at the very start of a line we have a problem, if we don't put in this code.

```

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

```

```

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

```

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

```

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

```

\setline You can use `\setline{<num>}` in running text (i.e., within `\pstart... \pend`) to set the current visible line-number to a specified positive value.

```

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

```

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

```

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

```


`\startlock` You can use `\startlock` or `\endlock` in running text to start or end line number locking at the current line. They decide whether line numbers or sub-line numbers are affected, depending on the current state of the sub-lineation flags.

```

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

```

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

```

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

```

VI Marking text for notes

The `\edtext` macro is used to create all footnotes and endnotes, as well as to print the portion of the main text to which a given note or notes is keyed. The idea is to have that lemma appear only once in the `.tex` file: all instances of it in the main text and in the notes are copied from that one appearance.

The `\edtext` macro takes two arguments.

```
\edtext{#1}{#2}
```

- #1 is the piece of the main text being glossed; it gets added to the main text, and is also used as a lemma for notes to it.
- #2 is a series of subsidiary macros that generate various kinds of notes.

The `\edtext` macro may be used (somewhat) recursively; that is, `\edtext` may be used within its own first argument. The code would be much simpler without this feature, but nested notes will commonly be necessary: it is quite likely that we will have an explanatory note for a long passage and notes on variants for individual words within that passage. The situation we can't handle is overlapping notes that are not nested: for example, one note covering lines 10–15, and another covering 12–18. You can handle such cases by using the `\lemma` and `\linenum` macros within #2: they alter the copy of the lemma and the line numbers that are passed to the notes, and hence allow you to overcome any limitations of this system, albeit with extra effort.

The recursive operation of `\edtext` will fail if you try to use a copy that is called something other than `\edtext`. In order to handle recursion, `\edtext` needs to redefine its own definition temporarily at one point, and that does not work if the macro you are calling is not actually named `\edtext`. There is no problem as long as `\edtext` is not invoked in the first argument. If you want to call `\edtext` something else, it is best to create instead a macro that expands to an invocation of `\edtext`, rather than copying `\edtext` and giving it a new name; otherwise you will need to add an appropriate definition for your new macro to `\morenoexpands`.

Side effects of our line-numbering code make it impossible to use the usual footnote macros directly within a paragraph whose lines are numbered (see comments to `\do@line`, VII.2.1 p. 142). Instead, the appropriate note-generating command is appended to the list macro `\inserts@list`, and when `\pend` completes the paragraph it inserts all the notes at the proper places.

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

VI.1 `\edtext` itself

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

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

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

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

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

`\dummy@edtext` We now need to define a number of macros that allow us to weed out nested instances of `\edtext`, and other problematic macros, from our lemma. This is similar to what we did in reading the line-list file using `\dummy@ref` and various redefinitions—and that is because nested `\edtexts` macros create nested `\@ref` entries in the line-list file.

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

`\dummy@edtext@showlemma` Some time, we want to obtain only the first argument of `\edtext`, while also wrapping it in `\showlemma`. For example, when printing a `\eledsection`.

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

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

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

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

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

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

get confused about the difference between alignment parameters and macro parameters; those that use temporary control sequences; and those that look carefully at what the current font is.

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

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

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

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

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

```

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

`\@tag` Now, we define an empty `\@tag` command. It will be redefine by `\edtext`: its value is the first argument. It will be used by the `\Xfootnote` commands.

```

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

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

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

\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.

```
1535 \newif\if@edtext@secondarg%
1536 %
```

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

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

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

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

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

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

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

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

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

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

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

By default, we do not use `\lemma`

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

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

We get the next series of samewords data in the list of samewords data for the current edtext level. We push them inside `\sw@inthisedtext`.

```

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

```

`\@tag` Our normal lemma is just argument #1; but that argument could have further invocations of `\edtext` within it. We get a copy of the lemma without any `\edtext` macros within it by temporarily redefining `\edtext` to just copy its first argument and ignore the other, and then expand #1 into `\@tag`, our lemma.

This is done within a group that starts here, in order to get the original `\edtext` restored; within this group we have also turned off the expansion of those control sequences commonly found within text that can cause trouble for us.

```

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

```

`\l@d@nums` Prepare more data for the benefit of note-generating macros: the line references and font specifier for this lemma go to `\l@d@nums`.

```

1567     \set@line%
1568 %

```

`\insert@count` will be altered by the note-generating macros: it counts the number of deferred footnotes or other insertions generated by this instance of `\edtext`. If we are in a right column (`reledpar`), we use `\insert@countR` instead of `\insert@count`.

```

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

```

Now process the note-generating macros in argument #2 (i.e., `\Afootnote`, `\lemma`, etc.). `\ignorespaces` is here to skip over any spaces that might appear at the start of #2; otherwise they wind up in the main text. Footnote and other macros that are used within #2 should all end with `\ignorespaces` as well, to skip any spaces between macros when several are used in series.

```

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

```

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

```

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

```

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

```

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

```

Finally, we are ready to admit the first argument into the current paragraph.

It is important that we generate and output all the notes for this chunk of text *before* putting the text into the paragraph: notes that are referenced by line number should generally be tied to the start of the passage they gloss, not the end. That should all be done within the expansion of `#2` above, or in `\aftergroup` commands within that expansion.

```

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

```

Finally, we add any insertions that are associated with the *end* of the lemma. Footnotes that are identified by symbols rather than by where the lemma begins in the main text need to be done here, and not above.

```

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

```

We switch some flags to false.

- The one that checks having footnotes inside a `\edtext`.
- The one that says we are inside a `\edtext`. In fact, it is not a flag, but a counter which is increased to 1 in each level of `\edtext`.
- The one that says we are inside a `\@lemma`.

```

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

```

We also reset `\@beforeinsertofthisedtext`

```

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

```

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

```

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

```

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

```

1625 \let\@beforeinsertofthisedtext\relax
1626 %

```

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


```

1627 \newif\ifnumberline
1628 \numberlinetrue
1629 %

```

\set@line The `\set@line` macro is called by `\edtext` to put the line-reference field and font specifier for the current block of text into `\l@d@nums`.

One instance of `\edtext` may generate several notes, or it may generate none — it is legitimate for argument #2 to `\edtext` to be empty. But `\flag@start` and `\flag@end` induce the generation of a single entry in `\line@list` during the next run, and it is vital to also remove one and only one `\line@list` entry here.

If no more lines are listed in `\line@list`, something is wrong — probably just some change in the input. We set all the numbers to zeros, following an old publishing convention for numerical references that have not yet been resolved.

```

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

```

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

```

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

```

VI.2 Substitute lemma

\lemma The `\lemma{<text>}` macro allows you to change the lemma that is passed on to the notes. Read about `\@tag` in normal `\edtext` macro for more details about `\sw@list@inedtext` and `\no@expands` (VI.1 p. 126).

```

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

```

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

```

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

```

\if@lemmacommand@ This boolean is set to TRUE inside a \edtext (or \critext) when a \lemma command is called. That is useful for some commands which can have a different behavior if the lemma in the note is different from the lemma in the main text.

```

1667 \newif\if@lemmacommand@%
1668 %

```

VI.3 Substitute line numbers

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

As argument \linenum takes a set of seven parameters separated by vertical bars, in the format used internally for \l@d@nums (see V.9 p. 97): the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma. However, you can omit any parameters you do not want to change, and you can omit a string of vertical bars at the end of the argument. Hence \linenum{18|4|0|18|7|1|0} is an invocation that changes all the parameters, but \linenum{|3} only changes the starting line number, and leaves the rest unaltered.

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

```

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

```

\line@set \linenum calls \line@set to do the actual work; it looks at the first number in the argument to \linenum, sets the corresponding value in \l@d@nums, and then calls itself to process the next number in the \linenum argument, if there are more numbers in \l@d@nums to process.

```

1674 \def\line@set#1|#2\|#3|#4\|{%
1675   \gdef\@tempb{#1}%
1676   \ifx\@tempb\empty
1677     \l@d@add{#3}%
1678   \else
1679     \l@d@add{#1}%
1680   \fi
1681   \gdef\@tempb{#4}%
1682   \ifx\@tempb\empty\else
1683     \l@d@add{|\}\line@set#2\|#4\|}%
1684   \fi}
1685 %

```

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

```

1686 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1687
1688 %

```

VI.4 Lemma disambiguation

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

The general mechanism is the following:

- **At the first run**, each `\sameword` command increments an `etoolbox` counter the name of which contains the argument of the `\sameword` commands.
- Then this counter, associated with the argument of `\sameword` is stored with the `\@sw` command in the auxiliary file of the current `reledmac` section (the `.1`, `.2...` file).
- **When this auxiliary file is read at the second run**, different operations are achieved:
 1. Get the rank of each `\sameword` in a line (relative rank) from the rank of each `\sameword` in all the numbered section (absolute rank):
 - For each paired `\sameword` argument and absolute line number, a counter is defined. Its value corresponds to the number of times `\sameword{⟨argument⟩}` is called from the beginning of the lineation to the end of the current line. We also store the same data for the preceding absolute line number, if it does not have `\sameword{⟨argument⟩}`.
 - For each `\sameword` having the same argument, we subtract from its absolute rank the number stored for the paired `\sameword` argument and previous absolute line number. Consequently, we obtain the relative rank.

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

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

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

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

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

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

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

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

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

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

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

After theory, implementation.

`\get@sw@txt` As the argument of `\sameword` can contain an active character if we use `inputenc` with `utf8` option instead of native UTF-8 engine, we store its detokenized content in a macro in order to allow the dynamic name of macro with `\csname`.²⁷

Because there is a bug with `\detokenize` and \LaTeX when using non BMP characters²⁸, we detokenize only for non- \LaTeX engines. In any case, in \LaTeX a `\csname` construction can contain UTF-8 characters without a problem, as UTF-8 characters are not managed with category codes, but instead read directly as UTF-8 characters.

```

1689 \newcommand{\get@sw@txt}[1]{%
1690   \ifxetex%
1691     \xdef\sw@txt{#1}%
1692   \else%
1693     \expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize{#1}}%
1694   \fi%
1695 }%
1696 %

```

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

```

1697 \newcommandx{\sameword}[2][1,usedefault]{%
1698   \leavevmode%
1699   \get@sw@txt{#2}%
1700 %

```

²⁷See <http://tex.stackexchange.com/q/244538/7712>.

²⁸<http://sourceforge.net/p/xetex/bugs/108/>

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

```
1701 \unless\ifledRcol%
1702 \csnumgdef{sw@sw@txt}{\csuse{sw@sw@txt}+\@ne}%
1703 %
```

Then, write its value to the numbered file.

```
1704 \protected@write\linenum@out{}{\string\@sw{sw@txt}{\csuse{sw@sw@txt}
1705 }}{#1}}%
1706 %
```

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

```
1706 \else%
1707 \csnumgdef{sw@sw@txt}{\csuse{sw@sw@txt}+\@ne}%
1708 \protected@write\linenum@outR{}{\string\@sw{sw@txt}{\csuse{sw@sw@txt}
1709 }}{#1}}%
1710 \fi%
1711 %
```

And print the word.

```
1711 #2%
1712 }%
1713 %
```

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

```
\if@addsw14 \newif\if@addsw%
1715 %
```

\@sw The command printed in the auxiliary files.

```
1716 \newcommand{\@sw}[3]{%
1717 \get@sw@txt{#1}%
1718 \unless\ifledRcol%
1719 %
```

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

```
1720 \csxdef{sw@sw@txt @\the\absline@num @\the\section@num}{#2}%
1721 %
```

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

```
1722 \numdef{\prev@line}{\the\absline@num-1}%
1723 \ifcsundef{sw@sw@txt @\prev@line @\the\section@num}{%
1724 \csnumgdef{sw@sw@txt @\prev@line @\the\section@num}{#2-1}%
1725 }{}%
1726 %
```

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

```
1727 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1728 %
```

And do the same thing for the right side.

```
1729 \else%
1730 \csxdef{sw@sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1731 \numdef{\prev@line}{\the\absline@numR-1}%
1732 \ifcsundef{sw@sw@txt @\prev@line @\the\section@numR @R}{%
1733 \csnumgdef{sw@sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1734 }{}%
1735 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R
}}%
1736 \fi%
1737 %
```

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

```
1738 \@tempcnta=\@edtext@level
1739 \@whilenum{\@tempcnta>0}\do{%
1740 \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1741 {%
1742 \@addswfalse%
1743 \notbool{lemmacommand@\the\@tempcnta}%
1744 {\@addswtrue}%
1745 {\IfStrEq{#3}{inlemma}%
1746 {\@addswtrue}%
1747 {%
1748 \def\do##1{%
1749 \ifnumequal{##1}{\the\@tempcnta}%
1750 {\@addswtrue\listbreak}%
1751 }%
1752 }%
1753 \docsvlist{#3}%
1754 }%
1755 }%
1756 \if@addsw%
1757 \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1758 \ifledRcol%
1759 \xright@appenditem{\the@sw}{\the\absline@numR}\to\@tmp%
1760 \else%
1761 \xright@appenditem{\the@sw}{\the\absline@num}\to\@tmp%
1762 \fi%
1763 \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
1764 \fi%
1765 }%
1766 {%
1767 \advance\@tempcnta by -1%
1768 }%
1769 }%
```

```
1770 %
```

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

```
1771 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1772   \get@sw@txt{#2}%
1773   \unless\ifledRcol@%
1774 %
```

Just a precaution.

```
1775   \ifx\sw@list@inedtext\empty%
1776     \def\the@sw{999}%
1777     \def\this@absline{-99}%
1778   \else%
1779 %
```

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

```
1780   \gl@p\sw@list@inedtext\to\@tmp%
1781   \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1782   \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1783   \fi%
1784 %
```

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

```
1785   \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
1786     \numdef{\prev@line}{\this@absline-1}%
1787     \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1788     }%
1789     {\numdef{\sw@atthisline}{0}}%
1790 %
```

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

```
1791   \ifnumgreater{\sw@atthisline}{1}%
1792     {\showwordrank{#2}{\the@sw}}%
1793     {#2}%
1794 %
```

And the same for right side.

```
1795   \else%
1796     \ifx\sw@list@inedtext\empty%
1797       \def\the@sw{999}%
1798       \def\this@absline{-99}%
1799     \else%
1800       \gl@p\sw@list@inedtext\to\@tmp%
1801       \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1802       \edef\this@absline{\expandafter\@secondoftwo\@tmp}%

```



```

1803 \fi%
1804 \ifcsdef{sw@\sw@txt @\this@absline @\the\section@numR @R}{%
1805 \numdef{\prev@line}{\this@absline-1}%
1806 \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
1807 }%
1808 {\numdef{\sw@atthisline}{0}}%
1809 \ifnumgreater{\sw@atthisline}{1}%
1810 {\showwordrank{#2}{\the@sw}}%
1811 {#2}%
1812 \fi%
1813 }%
1814 %

```

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

```

1815 \newcommand{\showwordrank}[2]{%
1816 #1\textsuperscript{#2}%
1817 }%
1818 %

```

VII Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than straight onto the vertical list, and when the paragraph ends we slice the paragraph into its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

VII.1 Boxes, counters, `\pstart` and `\pend`

`\raw@text` Here are numbers and flags that are used internally in the course of the paragraph decomposition.

`\ifnumberedpar@`

`\numberedpar@true` When we first form the paragraph, it goes into a box register, `\raw@text`, instead

`\numberedpar@false` 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

`\num@lines` paragraph when it is complete. When we chop it up into lines, each line in turn goes

`\one@line` into the `\one@line` register, and `\par@line` will be the number of that line within the

`\par@line` paragraph.

```

1819 \newbox\raw@text
1820 \newif\ifnumberedpar@
1821 \newcount\num@lines
1822 \newbox\one@line
1823 \newcount\par@line
1824 %

```

`\pstart` `\pstart` starts the paragraph by clearing the `\inserts@list` list and other relevant variables, and then arranges for the subsequent text to go into the `\raw@text` box.
`\AtEveryPstart`
`\numberpstarttrue` `\pstart` needs to appear at the start of every paragraph that is to be numbered; the
`\numberpstartfalse` `\autopar` command below may be used to insert these commands automatically.
`\labelpstarttrue`
`\labelpstartfalse` Beware: everything that occurs between `\pstart` and `\pend` is happening within
`\thepstart` a group; definitions must be global if you want them to survive past the end of the paragraph.

```

1825
1826 \newcommand{\AtEveryPstart}[1]{%
1827   \ifstrempy{#1}%
1828   {\xdef\at@every@pstart{}}%
1829   {\gdef\at@every@pstart{\noindent#1}}%
1830 }%
1831 \xdef\at@every@pstart{}%
1832
1833 \newcounter{pstart}
1834 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
1835 \newif\ifnumberpstart
1836 \numberpstartfalse
1837 \newif\iflabelpstart
1838 \labelpstartfalse
1839 \newcommandx*\pstart[1][1]{%
1840   \normal@pars%
1841   \ifstrempy{#1}{\at@every@pstart}{\noindent#1}%
1842   \ifautopar%
1843     \autopar%
1844   \fi%
1845   \ifluatex%
1846     \edef\l@luatextextdir@L{\the\textdir}%
1847   \fi%
1848   \@nobraektrue%
1849   \ifnumbering \else%
1850     \led@err@PstartNotNumbered%
1851     \beginnumbering%
1852   \fi%
1853   \ifnumberedpar@%
1854     \led@err@PstartInPstart%
1855     \pend%
1856   \fi%
1857   \list@clear{\inserts@list}%
1858   \global\let\next@insert=\empty%
1859   \begingroup\normal@pars%
1860   \global\advance \l@dnumpstartsL\@ne
1861   \global\setbox\raw@text=\vbox\bgroup%
1862     \if@nobraek%
1863       \if@afterindent\else%
1864         \noindent%
1865       \fi%
1866     \fi%

```

```

1867 \ifautopar\else%
1868 \ifnumberpstart%
1869 \ifistanza\else%
1870 \ifsidepstartnum\else%
1871 \thepstart%
1872 \fi%
1873 \fi%
1874 \fi%
1875 \fi%
1876 \numberedpar@true%
1877 \iflabelpstart\protected@edef\@currentlabel%
1878   {\p@pstart\thepstart}
1879 \fi%
1880 \l@dzeropenalties%
1881 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
1882 }
1883 %

```

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

```

1884 \newcommand*{\pend}[1][1]{\ifnumbering \else%
1885   \led@err@PendNotNumbered%
1886 \fi%
1887 \global\l@dskipversenumberfalse%
1888 \ifnumberedpar@ \else%
1889   \led@err@PendNoPstart%
1890 \fi%
1891 %

```

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.

```

1892 \l@dzeropenalties%
1893 \endgraf\global\num@lines=\prevgraf\egroup%
1894 \global\par@line=0%
1895 %

```

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`.

```

1896 \csnumdef{pstartline}{0}%
1897 \loop\ifvbox\raw@text%
1898   \csnumdef{pstartline}{\pstartline+\@ne}%
1899   \do@line%

```

```

1900 \ifbypstart@%
1901 \ifnumequal{\pstartline}{1}{%
1902   \bgroup%
1903   \let\leavevmode\relax%
1904   \setline{1}%
1905   \egroup%
1906   \resetprevline@{}{}%
1907   \fi%
1908 \repeat%
1909 %

```

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

```

1910 \flush@notes%
1911 \endgroup%
1912 \ignorespaces%
1913 %

```

Increase `pstart` counter.

```

1914 \ifnumberpstart%
1915   \global\pstartnumtrue%
1916 \fi%
1917 \addtocounter{pstart}{1}%
1918 %

```

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

```

1919 \normal@pars%
1920 \ifstrempy{#1}{\at@every@pend}{\noindent#1}%
1921 %

```

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.

```

1922 \@nobreakfalse%
1923 \ifautopar%
1924   \autopar%
1925 \fi%
1926 }
1927 %

```

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.

```

\AtEveryPend28
\at@every@pend29 \newcommand{\AtEveryPend}[1]{%
1930   \ifstrempy{#1}%
1931     {\xdef\at@every@pend{}}%
1932     {\gdef\at@every@pend{\noindent#1}}%
1933 }%

```

```

1934 \xdef\at@every@pend{}\%
1935
1936 %

```

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

```

1937 \newcommand*\l@dzeropenalties{%
1938   \brokenpenalty \z@ \clubpenalty \z@
1939   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
1940   \postdisplaypenalty \z@ \widowpenalty \z@}
1941
1942 %

```

`\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.

```

1943 \newif\ifautopar
1944 \autoparfalse
1945 \newcommand*\autopar{%
1946   \ifledRcol
1947     \ifnumberingR \else
1948       \led@err@AutoparNotNumbered
1949     \beginnumberingR
1950     \fi
1951   \else
1952     \ifnumbering \else
1953       \led@err@AutoparNotNumbered

```

```

1954 \beginnumbering
1955 \fi
1956 \fi
1957 \autopartrue
1958 \everypar{\setbox0=\lastbox
1959 \endgraf \vskip-\parskip
1960 \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\
fi\fi
1961 \let\par=\pend}%
1962 \ignorespaces}
1963 %

```

\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.

```

1964 \newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}
1965
1966 %

```

\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.

```

1967 \newif\ifautopar@pause
1968 %

```

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

```

1969 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
1970 \newcommand*{\do@line}{%
1971 {\vbadness=10000
1972 \splittopskip=\z@
1973 \do@linehook
1974 \l@emptyd@ta
1975 \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
1976 \unvbox\one@line \global\setbox\one@line=\lastbox
1977 \getline@num
1978 \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{\}
1979 \ifnum\@lock>\@ne
1980 \inserthangingsymboltrue
1981 \else
1982 \inserthangingsymbolfalse
1983 \fi
1984 \check@pb@in@verse

```

```

1985 \ifl@dhidenumber%
1986 \global\l@dhidenumberfalse%
1987 \f@x@l@cks%
1988 \else%
1989 \affixline@num%
1990 \fi%
1991 %

```

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

```

1992 \xifinlist{\the\l@dnumstartsl}{\eled@sections@@}%
1993 {\print@eledsection}%
1994 {\print@line}%
1995 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{%}
1996 }%
1997 %

```

VII.2.2 Process for “normal” line

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

```

1998 \def\print@line{
1999 %

```

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

```

2000 \affixpstart@num%
2001 %

```

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

```

2002 \hb@xt@ \linewidth{%
2003 %

```

User hook.

```

2004 \do@insidelinehook%
2005 %

```

Left line number

```

2006 \l@dld@ta%
2007 %

```

Prepare text to be inserted before notes.

```

2008 \if@firstlineofpage%
2009 \set@Xtxtbeforenotes%
2010 \global\@firstlineofpagefalse%
2011 \fi%
2012 %

```

Insert footnotes made of manuscripts data and critical footnotes.

```

2013 \ifdefstring{\ms@data@position}{msdata-regular}{%
2014 \insert@msdata%
2015 \add@inserts%
2016 \add@Xgroupbyline%
2017 }{%
2018 \add@inserts%
2019 \add@Xgroupbyline%
2020 \insert@msdata%
2021 }%
2022 %

```

Insert marginal notes.

```

2023 \affixside@note%
2024 %

```

Print left notes.

```

2025 \l@dlsn@te
2026 %

```

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

```

2027 {\ledllfill\hb@xt@ \wd\one@line{\new@line%
2028 %

```

If we use Lua \TeX then restore the direction.

```

2029 \ifluatex%
2030 \textdir\l@luatextextdir@L%
2031 \fi%
2032 %

```

Insert, if needed, the hanging symbol.

```

2033 \inserthangingsymbol%
2034 %

```

And so, print the line.

```

2035 \l@dunhbox@line{\one@line}}%
2036 %

```

Right line number

```

2037 \ledrlfill\l@drd@ta%
2038 %

```

Print right notes.

```

2039 \l@drsn@te
2040 }}%
2041 %

```

And reinsert penalties (for page breaking)...

```

2042 \add@penalties%
2043 }
2044 %

```


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.

```

2045 \def\print@eledsection{%
2046   \if@firstlineofpage%
2047     \set@Xtxtbeforenotes%
2048     \global\@firstlineofpagefalse%
2049   \fi%
2050   \ifdefstring{\ms@data@position}{msdata-regular}{%
2051     \insert@msdata%
2052     \add@inserts%
2053     \add@Xgroupbyline%
2054   }{%
2055     \add@inserts%
2056     \add@Xgroupbyline%
2057     \insert@msdata%
2058   }%
2059   \affixside@note%
2060   \numdef{\temp@}{\l@dnumpstartsL-1}%
2061   \xifinlist{\temp@}{\eled@sections@}{\@nbreaktrue}{\@nbreakfalse}%
2062   \@eled@sectioningtrue%
2063   \csuse{eled@sectioning@the\l@dnumpstartsL}%
2064   \@eled@sectioningfalse%
2065   \global\csundef{eled@sectioning@the\l@dnumpstartsL}%
2066   \if@RTL%
2067     \hspace{-3\paperwidth}%
2068     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2069   \else%
2070     \hspace{3\paperwidth}%
2071     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2072   \fi%
2073   \vskip-\baselineskip%
2074 }
2075 %

```

VII.2.4 Hooks

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

```

2076 \newcommand*{\do@linehook}{}
2077 \newcommand*{\do@insidelinehook}{}
2078 %

```

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

```
2079 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
2080 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
2081
2082 %
```

VII.2.5 Sidenotes and marginal line number initialization

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

```
2083 \l@dcsnotetext \newcommand*{\l@emptyd@ta}{%
2084 \l@dcsnotetext@l \gdef\l@dld@ta{}%
2085 \l@dcsnotetext@r \gdef\l@drd@ta{}%
2086 \gdef\l@dcsnotetext@l{}%
2087 \gdef\l@dcsnotetext@r{}%
2088 \gdef\l@dcsnotetext{}%
2089
2090 %
```

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

```
2091 \l@drsn@te \newcommand{\l@dlsn@te}{%
2092 \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep}}
2093 \newcommand{\l@drsn@te}{%
2094 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
2095
2096 %
```

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

```
2097 \newcommand*{\ledllfill}{\hfil}
2098 \newcommand*{\ledrlfill}{}
2099
2100 %
```

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.

```
2101 \newcommand*{\getline@num}{%
2102 \global\advance\absline@num \@ne%
2103 \do@actions
2104 \do@ballast
```

```

2105 \ifnumberline
2106     \ifsublines@
2107         \ifnum\sub@lock<\tw@
2108             \global\advance\subline@num \@ne
2109         \fi
2110     \else
2111         \ifnum\@lock<\tw@
2112             \global\advance\line@num \@ne
2113             \global\subline@num \z@
2114         \fi
2115     \fi
2116 \fi
2117 }
2118 %

```

\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. 157).

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

\c@ballast

```

2119 \newcount\ballast@count
2120 \newcounter{ballast}
2121 \setcounter{ballast}{0}
2122 %

```

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.

```

2123 \newcommand*{\do@ballast}{\global\ballast@count \z@
2124 \begingroup
2125     \advance\absline@num \@ne
2126     \ifnum\next@actionline=\absline@num
2127         \ifnum\next@action>-1001\relax
2128             \global\advance\ballast@count by -\c@ballast
2129         \fi
2130     \fi
2131 \endgroup}
2132 %

```

\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`.

```

2133 \newcommand*{\do@actions}{%
2134   \global\let\do@actions@next=\relax
2135   \ifnum\absline@num<\next@actionline\else
2136 %

```

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.

```

2137   \ifnum\next@action>-1001
2138     \global\page@num=\next@action
2139     \global\@firstlineofpagetrue%
2140     \ifbypage@
2141       \global\line@num=\z@ \global\subline@num=\z@
2142       \resetprevline@
2143     \fi
2144     \add@msdata@firstlineofpage%
2145 %

```

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`.)

```

2146   \else
2147     \ifnum\next@action<-4999
2148       \@l@tempcnta=-\next@action
2149       \advance\@l@tempcnta by -5001
2150       \ifsublines@
2151         \global\subline@num=\@l@tempcnta
2152       \else
2153         \global\line@num=\@l@tempcnta
2154       \fi
2155 %

```

We rescale the value in `\@l@tempcnta` so that we can use a case statement.

```

2156   \else
2157     \@l@tempcnta=-\next@action
2158     \advance\@l@tempcnta by -1000
2159     \do@actions@fixedcode
2160   \fi
2161 \fi
2162 %

```

Now we get information about the next action off the list, and then set `\do@actions@next` so that we will call ourselves recursively: the next action might also be for this line.

There is no warning if we find `\actionlines@list` empty, since that will always happen near the end of the section.

```

2163   \ifx\actionlines@list\empty
2164     \gdef\next@actionline{1000000}%
2165   \else
2166     \gl@p\actionlines@list\to\next@actionline
2167     \gl@p\actions@list\to\next@action

```

```

2168 \global\let\do@actions@next=\do@actions
2169 \fi
2170 \fi
2171 %

```

Make the recursive call, if necessary.

```

2172 \do@actions@next}
2173
2174 %

```

`\do@actions@fixedcode` This macro handles the fixed codes for `\do@actions`. It is one big case statement.

```

2175 \newcommand*{\do@actions@fixedcode}{%
2176 \ifcase\@l@dttempcnta
2177 \or% % 1001 = starting sublineation
2178 \global\sublines@true
2179 \or% % 1002 = ending sublineation
2180 \global\sublines@false
2181 \or% % 1003 = starting locking number
2182 \global\@lock=\@ne
2183 \or% % 1004 = ending locking number
2184 \ifnum\@lock=\tw@
2185 \global\@lock=\thr@@
2186 \else
2187 \global\@lock=\z@
2188 \fi
2189 \or% % 1005 = starting locking subnumber
2190 \global\sub@lock=\@ne
2191 \or% % 1006 = ending locking subnumber
2192 \ifnum\sub@lock=\tw@
2193 \global\sub@lock=\thr@@
2194 \else
2195 \global\sub@lock=\z@
2196 \fi
2197 \or% % 1007 = skipping numbering
2198 \l@dskipnumbertrue
2199 \or% % 1008 = skipping numbering in stanza
2200 \l@dskipversenumbertrue%
2201 \or% % 1009 = hiding number
2202 \l@dhidenumbertrue
2203 \or% % 1010 = inserting msdata
2204 \add@msdata%
2205 \else
2206 \led@warn@BadAction
2207 \fi}
2208
2209
2210 %

```

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` \leq `\firstlinenum`, we compare the two directly instead of making these calculations.

We compute, in the scratch counter `\@l@tempcnta`, the number of the next line that should be printed with a number (*m* in the above discussion), and move the current line number into the counter `\@l@tempcntb` for comparison.

First, the case when we are within a sub-line range.

```
2211 \newcommand*{\affixline@num}{%
2212 %
```

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).

```
2213 \ifledgroupnotesL@else
2214 \ifnumberline
2215 \ifl@dskipnumber
2216 \global\l@dskipnumberfalse
2217 \else
2218 \ifsublines@
2219 \l@tempcntb=\subline@num
2220 \ifnum\subline@num>\c@firstsublinenum
2221 \l@tempcnta=\subline@num
2222 \advance\l@tempcnta by-\c@firstsublinenum
2223 \divide\l@tempcnta by\c@sublinenumincrement
2224 \multiply\l@tempcnta by\c@sublinenumincrement
2225 \advance\l@tempcnta by\c@firstsublinenum
2226 \else
2227 \l@tempcnta=\c@firstsublinenum
2228 \fi
2229 %
```

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.

```
2230 \ch@cksub@l@ck
2231 %
```

Now the line number case, which works the same way.

```

2232     \else
2233         \@l@tempcntb=\line@num
2234     %

```

Check on the `\linenumberlist` If it is `\empty` use the standard algorithm.

```

2235         \ifx\linenumberlist\empty
2236             \ifnum\line@num>\c@firstlinenum
2237                 \@l@tempcnta=\line@num
2238                 \advance\@l@tempcnta by-\c@firstlinenum
2239                 \divide\@l@tempcnta by\c@linenumincrement
2240                 \multiply\@l@tempcnta by\c@linenumincrement
2241                 \advance\@l@tempcnta by\c@firstlinenum
2242             \else
2243                 \@l@tempcnta=\c@firstlinenum
2244             \fi
2245         \else
2246     %

```

The `\linenumberlist` was not `\empty`, so here is Wayne's numbering mechanism. This takes place in `TEX`'s mouth.

```

2247         \@l@tempcnta=\line@num
2248         \edef\rem@inder{,\linenumberlist,\number\line@num,}%
2249         \edef\sc@n@list{\def\noexpand\sc@n@list
2250             ###1,\number\@l@tempcnta,###2|{\def\noexpand\rem@inder
2251 {####2}}}%
2252         \sc@n@list\expandafter\sc@n@list\rem@inder|
2253         \ifx\rem@inder\empty%
2254             \advance\@l@tempcnta\@ne
2255         \fi
2256     %

```

A locking check for lines, just like the version for sub-line numbers above.

```

2257         \check@l@ck
2258     \fi
2259 %

```

The following tests are true if we need to print a line number.

```

2260         \ifnum\@l@tempcnta=\@l@tempcntb
2261             \ifl@dskipversenumber\else
2262     %

```

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`

```

2263         \if@twocolumn
2264             \if@firstcolumn
2265                 \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2266             \else
2267                 \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2268             \fi
2269         \else
2270             \ifboolexpr{bool {l@dprintingcolumns} and test {\line@margin@columns}{\m@ne}}%
2271             {\@l@tempcntb=\line@margin@columns}%
2272             {\@l@tempcntb=\line@margin}%
2273             \ifnum\@l@tempcntb>\@ne
2274                 \advance\@l@tempcntb \page@num
2275             \fi
2276             \ifodd\@l@tempcntb
2277                 \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
2278             \else
2279                 \gdef\l@dld@ta{\llap{\leftlinenum}}}%
2280             \fi
2281         \fi
2282     \fi
2283 \fi
2284 %

```

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

```

2285         \f@x@l@cks
2286     \fi
2287 \fi
2288 \fi
2289 }
2290
2291 %

```

`\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.

```

2292 \newcommand*{\ch@cksub@l@ck}{%
2293     \ifcase\sub@lock
2294     \or
2295     \ifnum\sublock@disp=\@ne

```



```

2296         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2297     \fi
2298 \or
2299     \ifnum\sublock@disp=\tw@ \else
2300         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2301     \fi
2302 \or
2303     \ifnum\sublock@disp=\z@
2304         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2305     \fi
2306 \fi}
2307 %

```

Similarly for line numbers.

```

2308 \newcommand*{\check@lock}{%
2309     \ifcase\@lock
2310     \or
2311         \ifnum\lock@disp=\@ne
2312             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2313         \fi
2314     \or
2315         \ifnum\lock@disp=\tw@ \else
2316             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2317         \fi
2318     \or
2319         \ifnum\lock@disp=\z@
2320             \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2321         \fi
2322 \fi}
2323 %

```

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

```

2324 \newcommand*{\fix@locks}{%
2325     \ifcase\@lock
2326     \or
2327         \global\@lock=\tw@
2328     \or \or
2329         \global\@lock=\z@
2330     \fi
2331     \ifcase\sub@lock
2332     \or
2333         \global\sub@lock=\tw@
2334     \or \or
2335         \global\sub@lock=\z@
2336     \fi}
2337 %
2338 %

```

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.

```

\leftpstartnum39
\rightpstartnum40 \newif\ifsidepstartnum
\ifsidepstartnum41 \newcommand*{\affixpstart@num}{%
2342     \ifsidepstartnum
2343         \if@twocolumn
2344             \if@firstcolumn
2345                 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2346             \else
2347                 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2348             \fi
2349         \else
2350             \l@dttempcntb=\line@margin%
2351             \ifnum\l@dttempcntb>\@ne
2352                 \advance\l@dttempcntb \page@num
2353             \fi
2354             \ifodd\l@dttempcntb
2355                 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2356             \else
2357                 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2358             \fi
2359         \fi
2360     \fi
2361 }
2362 %
2363
2364
2365 \newif\ifpstartnum
2366 \pstartnumtrue
2367 \newcommand*{\leftpstartnum}{
2368     \ifpstartnum\thepstart
2369     \kern\linenumsep\fi
2370     \global\pstartnumfalse
2371 }
2372 \newcommand*{\rightpstartnum}{
2373     \ifpstartnum

```

```

2374 \kern\linenumsep
2375 \thepstart
2376 \fi
2377 \global\pstartnumfalse
2378 }
2379 %

```

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.

```

2380 \list@create{\inserts@list}
2381 %

```

`\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`.

```

2382 \newcommand*{\add@inserts}{%
2383 \global\let\add@inserts@next=\relax
2384 %

```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```

2385 \ifx\inserts@list\empty \else
2386 %

```

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.

```

2387 \ifx\next@insert\empty
2388 \ifx\insertlines@list\empty
2389 \global\noteschanged@true
2390 \gdef\next@insert{100000}%
2391 \else
2392 \gl@p\insertlines@list\to\next@insert
2393 \fi
2394 \fi
2395 %

```

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.

```

2396 \ifnum\next@insert=\absline@num
2397 \gl@p\inserts@list\to\@insert
2398 \@insert
2399 \global\let\@insert=\undefined
2400 \global\let\next@insert=\empty
2401 \global\let\add@inserts@next=\add@inserts
2402 \fi
2403 \fi
2404 %

```

Make the recursive call, if necessary.

```

2405 \add@inserts@next}
2406
2407 %

```

\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`.

```

2408 \newcommand{\add@Xgroupbyline}{%
2409 \unless\ifnocritical%
2410 \def\do##1{%Looping on the series
2411 \def\do####1{%Looping on the ##1@forinserting command
2412 \ifcsdef{##1@forinserting@####1}{%
2413 \X@beforeinsertion{##1}%
2414 \if@ledgroup%
2415 \global\setbox\@nameuse{mp##1footins}=\vbox%
2416 \else%
2417 \insert\csname ##1footins\endcsname%
2418 \fi%
2419 {%
2420 \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
2421 {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
2422 }%
2423 \if@ledgroup%
2424 \unvbox\@nameuse{mp##1footins}%

```

```

2425         \fi%
2426         \X@atbegininsertion{##1}%
2427         \ifcsstring{series@display##1}%
2428             {%
2429                 \Xledsetnormalparstuff{##1}%
2430                 \rule{z@}\splittopskip%
2431             }%
2432         {%
2433             \csuse{##1@forinserting@###1}%
2434             \strut\par%
2435         }%
2436         \global\csundef{##1@forinserting@###1}%
2437     }%
2438     {%
2439     }%
2440     \ifcsdef{##1@forinserting}{%
2441         \dolistcsloop{##1@forinserting}%
2442     }{%
2443         \global\csundef{##1@forinserting}%
2444     }%
2445     \dolistloop{\@series}%
2446     \fi%
2447 }%
2448
2449
2450 %

```

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. 147). Finally, the penalty is checked to see that it does not go below -10000 .

```

2451 \newcommand*{\add@penalties}{\@l@dttempcnta=\ballast@count
2452   \ifnum\num@lines>\@ne
2453     \global\advance\par@line \@ne
2454     \ifnum\par@line=\@ne
2455       \advance\@l@dttempcnta \clubpenalty
2456     \fi
2457     \@l@dttempcntb=\par@line \advance\@l@dttempcntb \@ne
2458     \ifnum\@l@dttempcntb=\num@lines
2459       \advance\@l@dttempcnta \widowpenalty

```

```

2460 \fi
2461 \ifnum\par@line<\num@lines
2462   \advance\@l@tempcnta \interlinepenalty
2463 \fi
2464 \fi
2465 \ifnum\@l@tempcnta=\z@
2466   \relax
2467 \else
2468   \ifnum\@l@tempcnta>-10000
2469     \penalty\@l@tempcnta
2470   \else
2471     \penalty -10000
2472   \fi
2473 \fi}
2474
2475 %

```

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.

```

2476 \newcommand*{\flush@notes}{%
2477   \@xloop
2478   \ifx\inserts@list\empty \else
2479     \glp\inserts@list\to\@insert
2480     \@insert
2481     \global\let\@insert=\undefined
2482   \repeat}
2483
2484 %

```

\@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.

```

2485 \def\@xloop#1\repeat{%

```

```

2486 \def\body{#1\expandafter\body\fi}%
2487 \body}
2488
2489 %

```

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.

```

2490 \newcommand{\set@Xtxtbeforenotes}{%
2491   \unless\ifnocritical{%
2492     \def\do##1{%
2493       \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2494     }%
2495     \dolistloop{\@series}%
2496   \fi%
2497 }%
2498 %

```

\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.

```

2499 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2500   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2501     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2502     \ifcsvoid{Xtxtbeforenotes@#1}{-}{%
2503       \ifcsstring{series@display#1}{paragraph}%
2504       {\noindent\csuse{Xtxtbeforenotes@#1}}%
2505       {\expandafter\insert\csname#1footins\endcsname%
2506         \bgroup%
2507           \noindent%
2508           \csuse{\csuse{series@display#1}@begin@insert}{#1}%
2509           \strut\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}%
2510         \egroup%
2511       }%
2512     }%
2513   }%
2514 }%
2515 }%
2516 %

```

XII Critical footnotes

The footnote macros are adapted from those in `PLAIN TEX`, but they differ in these respects: the outer-level commands must add other commands to a list macro rather

than doing insertions immediately; there are many separate levels of the footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

XII.1 Fonts

Before getting into the details of formatting the notes, we set up some font macros. It is the notes that present the greatest challenge for our font-handling mechanism, because we need to be able to take fragments of our main text and print them in different forms: it is common to reduce the size, for example, without otherwise changing the fonts used.

`\select@lemmafont` `\select@lemmafont` is provided to set the right font for the lemma in a note. This macro extracts the font specifier from the line and page number cluster, and issues the associated font-changing command, so that the lemma is printed in its original font.

```

2517 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@@lemmafont#7|}
2518 \def\select@@lemmafont#1/#2/#3/#4|{%
2519   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2520    \selectfont}
2521
2522 %

```

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.

```

2523 \newcommand*{\footnoteoptions@}[3]{%
2524   \def\do##1{%
2525     \ifstrequal{#1}{L}{% In Leftside
2526       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@list%
2527         \global\advance\insert@count \@ne% Increment the left insert
counter.
2528       }%
2529     }%
2530     \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@listR%
2531       \global\advance\insert@countR \@ne% Increment the right insert
counter insert.
2532     }%
2533   }%
2534   \notblank{#2}{\docsvlist{#2}}}% Parsing all options
2535 }
2536 %

```


XII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua¹TeX is used.

```

2537 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
2538   \ifstrequal{#1}{L}{%
2539     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
inserts@list%Know the dir of lemma
2540     \global\advance\insert@count \@ne%
2541     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@list%Know the dir of lemma
2542     \global\advance\insert@count \@ne%
2543   }%
2544   {%
2545     \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}{\to\
inserts@listR%Know the dir of lemma
2546     \global\advance\insert@countR \@ne%
2547     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@listR%Know the dir of lemma
2548     \global\advance\insert@countR \@ne%
2549   }%
2550 }
2551 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

2552 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2553   \ifstrequal{#1}{L}{%
2554     \if@RTL%
2555       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@list%Know the language used in the lemma
2556       \global\advance\insert@count \@ne%
2557     }%
2558     \else
2559       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@list%Know the language of lemma
2560       \global\advance\insert@count \@ne%
2561     }%
2562     \fi%
2563     \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}{\to\
inserts@list%Know the language of lemma
2564     \global\advance\insert@count \@ne%
2565   }%
2566   {%
2567     \if@RTL
2568       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@listR%Know the language of lemma
2569       \global\advance\insert@countR \@ne%
2570     }%
2571     \else
2572       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@listR%Know the language of lemma

```

```

2570 \global\advance\insert@countR \@ne%
2571 \fi
2572 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}\}\}\to\inserts@listR%Know the language of lemma
2573 \global\advance\insert@countR \@ne%
2574 }%
2575 }
2576 %

```

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`.

```

2577 \newcommand*{\footsplitskips}{%
2578 \interlinepenalty=\interfootnotelinepenalty
2579 \unless\ifl@dprintingpages%
2580 \floatingpenalty=\@MM%
2581 \fi%

```

```

2582 \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2583 \leftskip=\z@skip \rightskip=\z@skip
2584
2585 %

```

\normalfootnoterule \normalfootnoterule is a standard footnote-rule macro, for use by a footstart macro: just the same as the PLAIN T_EX footnote rule.

```

2586 \let\normalfootnoterule=\footnoterule
2587 %

```

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.

```

2588 \newcommand{\Xarrangement}[2][1,usedefault]{%
2589   \def\do##1{%
2590     \csname Xarrangement@#2\endcsname{##1}%
2591   }%
2592   \ifstrempy{#1}%
2593     {%
2594       \dolistloop{\@series}%
2595     }%
2596     {
2597       \docsvlist{#1}%
2598     }%
2599 }%
2600 %

```

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.

```

2601 \newcommand*{\Xarrangement@normal}[1]{%
2602   \csgdef{series@display#1}{normal}
2603   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
2604   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
2605   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
2606   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
2607   \expandafter\let\csname #1footnoterule\endcsname=%
2608                                     \normalfootnoterule
2609   \count\csname #1footins\endcsname=1000
2610   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2611   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2612   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2613   %

```

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.

```

2614   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
2615   side only
2616   %

```

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).

```

2616   \ifnoledgroup@else%
2617     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2618     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2619     \count\csname mp#1footins\endcsname=1000
2620     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2621     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2622     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2623   \fi
2624 }
2625
2626 %

```

`\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.

```

2627 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2628   \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
2629   \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
2630 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
2631   \X@beforeinsertion{#1}%
2632   \insert\csname #1footins\endcsname{%
2633     \X@atbegininsertion{#1}%
2634     \normalvfootnote@inserted{#1}{#2}%
2635   }%
2636 }%
2637 }%
2638 %

```

\normalvfootnote@inserted The `\normalvfootnote@inserted` macro is expanded to the content to be add to a `\insert` for normal critical footnote.

```

2639 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote@inserted}[2]{%
2640   \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
2641   \csuse{Xnotefontsize@#1}%
2642   \footssplitskips
2643   \ifl@dpairing\ifl@dpaging\else%
2644     \setXnoteswidthliketwocolumns@{#1}%
2645   \fi\fi%
2646   \setXnotespositionliketwocolumns@{#1}%
2647   \spaceskip=\z@skip \xspaceskip=\z@skip%
2648   \csname #1footfmt\endcsname #2{#1}%
2649 }%
2650 %

```

```

\X@beforeinsertion51 \newcommand{\X@beforeinsertion}[1]{%
2652   \if@ledgroup\else%
2653     \insert@Xtxtbeforenotes{#1}%
2654   \fi%
2655   \csuse{Xbeforeinserting@#1}%
2656 }%
2657 %

```

```

\X@atbegininsertion58 \newcommand{\X@atbegininsertion}[1]{%
2659   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2660 }%
2661 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote 2662 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
2663   \iftoggle{Xgroupbyline@#1}{%
2664     \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
2665   }%
2666   {%
2667     \global\setbox\@nameuse{mp#1footins}%
2668     \vbox{%
2669       \unvbox\@nameuse{mp#1footins}%
2670       \mpnormalvfootnote@inserted{#1}{#2}%
2671     }%
2672   }%
2673 }%
2674
2675 %

```

```

\mpnormalvfootnote@inserted 2676 \newcommand{\mpnormalvfootnote@inserted}[2]{%
2677   \noindent\csuse{Xhooknote@#1}%
2678   \csuse{Xnotefontsize@#1}%
2679   \hsize\columnwidth%
2680   \@parboxrestore%
2681   \color@begingroup%
2682   \csname #1footfmt\endcsname #2{#1}\color@endgroup%
2683 }%
2684 %

```

\normalfootfmt `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 97), 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.

```

2685 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
2686   \Xstorelineinfo{#1}{#4}%
2687   \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}{%
2688     \hangindent=\csuse{Xhangindent@#4}%
2689     \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2690     \nottoggle{Xgroupbyline@#4}{\rule\z@\splittopskip}{}%
2691     {\printlinefootnote{#1}{#4}}%
2692     \print@lemma{#1}{#2}{#4}%
2693     \csuse{Xwrapcontent@#4}{#3}%
2694     \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
2695   }%
2696 %

```

\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`.

```
2697 \newcommand*{\normalfootstart}[1]{%
2698 %
```

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. 217. Here is part of this algorithm, when the block of notes are ready to be printed.

```
2699 \ifdimequal{0pt}{\Xprenotes@}{}%
2700 {%
2701   \iftoggle{Xprenotes@}{%
2702     \togglefalse{Xprenotes@}%
2703     \skip\csname #1footins\endcsname=%
2704     \glueexpr\csuse{Xprenotes@}+\csuse{Xafterterrule@#1}\relax%
2705   }%
2706   }%
2707 }%
2708 \vskip\skip\csname #1footins\endcsname%
2709 %
```

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. 214 for the general description of the problem.

```
2710 \leftskip0pt \rightskip0pt
2711 \ifl@dpairing\else%
2712   \hsize=\old@hsize%
2713 \fi%
2714 \setXnoteswidthliketwocolumns@{#1}%
2715 \setXnotespositionliketwocolumns@{#1}%
2716 %
```

And now, print the footnote's rule to finish the footnote's introduction.

```
2717 \print@Xfootnoterule{#1}%
2718 }%
2719 %
```

`\normalfootgroup` `\normalfootgroup` is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

2720 \newcommand*{\normalfootgroup}[1]{%
2721   \csuse{Xhookgroup@#1}%
2722   \unvbox\csname #1footins\endcsname%
2723   \hsize=\old@hsize%
2724   }%
2725
2726 %

```

`\mpnormalfootgroup` A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the `\Xfootgroup` and `\Xfootstarts` macros.

```

2727 \unless\ifnoledgroup@
2728 \newcommand*{\mpnormalfootgroup}[1]{%
2729   \vskip\skip\@nameuse{mp#1footins}
2730   \ifl@dpairing\ifparledgroup%
2731     \leavevmode\marks\parledgroup@{begin}%
2732     \marks\parledgroup@series{#1}%
2733     \marks\parledgroup@type{Xfootnote}%
2734   \fi\fi\normalcolor%
2735   \ifparledgroup%
2736     \ifl@dpairing%
2737     \else%
2738       \setXnoteswidthliketwocolumns@{#1}%
2739       \setXnotespositionliketwocolumns@{#1}%
2740       \print@Xfootnoterule{#1}%
2741     \fi%
2742   \else%
2743     \setXnoteswidthliketwocolumns@{#1}%
2744     \setXnotespositionliketwocolumns@{#1}%
2745     \print@Xfootnoterule{#1}%
2746   \fi%
2747   \setlength{\parindent}{0pt}
2748   \csuse{Xhookgroup@#1}%
2749   \unvbox\csname mp#1footins\endcsname}}
2750 \fi
2751 %

```

XII.6.3 Paragraphed footnotes

The paragraphed-footnote option reformats all the footnotes of one series for a page into a single paragraph; this is especially appropriate when the notes are numerous and brief. The code is based on *The TeXbook*, pp. 398–400, with alterations for our environment. This algorithm uses a considerable amount of save-stack space: a \TeX of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

`\Xarrangement@paragraph` The `\Xarrangement@paragraph` macro sets up everything for one series of the footnotes so that they will be paragraphed; it takes the series letter as argument. We include

the setting of `\count\footins` to 1000 for the footnote series just in case user is switching to paragraphed footnotes after having columnar ones, since they change this value (see below).

The argument of `\Xarrangement@footparagraph` is the letter denoting the series of notes to be paragraphed.

```

2752 \newcommand*{\Xarrangement@paragraph}[1]{%
2753   \csgdef{series@display#1}{paragraph}
2754   \expandafter\let\csname #1footstart\endcsname=\parafootstart
2755   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
2756   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
2757   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
2758   \count\csname #1footins\endcsname=1000
2759   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2760   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2761   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2762   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2763   \para@footsetup{#1}
2764   %

```

And the extra setup for minipages.

```

2765 \ifnoledgroup@else
2766   \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
2767   \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
2768   \count\csname mp#1footins\endcsname=1000
2769   \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2770   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2771   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2772   \fi
2773 }
2774 %

```

`\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.

```

2775 \providecommand{\footfudgefiddle}{64}
2776 %

```

`\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`.

```

2777 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}

```

```

2778 \setXnoteswidthliketwocolumns@{#1}%
2779 \ifcsempy{Xwidth@#1}%
2780 {}%
2781 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
2782 \dimen0=\baselineskip
2783 \multiply\dimen0 by 1024
2784 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
2785 \csxdef{#1footfudgefactor}{%
2786 \expandafter\strip@pt\dimen0 }}
2787
2788 %

```

\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.

```

2789 \newcommand*{\parafootstart}[1]{%
2790 \rightskip=0pt \leftskip=0pt%
2791 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2792 \ifdimequal{0pt}{\Xprenotes@}{}%
2793 {%
2794 \iftoggle{Xprenotes@}{%
2795 \togglefalse{Xprenotes@}%
2796 \skip\csname #1footins\endcsname=%
2797 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
2798 }%
2799 }%
2800 }%
2801 \vskip\skip\csname #1footins\endcsname%
2802 \setXnoteswidthliketwocolumns@{#1}%
2803 \setXnotespositionliketwocolumns@{#1}%
2804 \print@Xfootnoterule{#1}%
2805 \let\bidir@RTL@everypar\@empty%
2806 \noindent\leavevmode}
2807 %

```

\paravfootnote \paravfootnote is a version of the \vfootnote command that is used for paragraphed notes. It gets appended to the \inserts@list list by an outer-level footnote command like \Afootnote. The first argument is the note series letter; the second is the full text of the printed note itself, including line numbers, lemmata, and footnote text.

The initial model for this insertion is, of course, the \insert\footins definition in *The TeXbook*, p.398. There, the footnotes are first collected up in hboxes, and these hboxes are later unpacked and stuck together into a paragraph.

However, Michael Downes has pointed out that because text in `hboxes` gets typeset in restricted horizontal mode, there are some undesirable side-effects if you later want to break such text across lines. In restricted horizontal mode, where \TeX does not expect to have to break lines, it does not insert certain items like `\discretionary`s. If you later unbox these `hboxes` and stick them together, as the *TeXbook* macros do to make these footnotes, you lose the ability to hyphenate after an explicit hyphen. This can lead to overfull `hboxes` when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.²⁹

Wayne Sullivan pointed out to us another subtle problem that arises from the same cause: \TeX also leaves the `\language` whatsit nodes out of the horizontal list.³⁰ So changes from one language to another will not invoke the proper hyphenation rules in such footnotes. Since critical editions often do deal with several languages, especially in a footnotes, we really ought to get this bit of code right.

To get around these problems, Wayne suggested emendations to the *TeXbook* versions of these macros which are broadly the same as those described by Michael: the central idea (also suggested by Donald Knuth in a letter to Michael) is to avoid collecting the text in an `hbox` in the first place, but instead to collect it in a `vbox` whose width is (virtually) infinite. The text is therefore typeset in unrestricted horizontal mode, as a paragraph consisting of a single long line. Later, there is an extra level of unboxing to be done: we have to unpack the `vbox`, as well as the `hboxes` inside it, but that is not too hard. For details, we refer you to Michael's article, where the issues are clearly explained.³¹ Michael's unboxing macro is called `\Xunvzx`: `unvbox`, extract the last line, and `unhbox` it.

Doing things this way has an important consequence: as Michael pointed out, you really can't put an explicit line-break into a note built in a `vbox` the way we are doing.³² In other words, be very careful not to use `\break`, or `\penalty-10000`, or any equivalent inside your para-footnote. If you do, most of the note will probably disappear. You *are* allowed to make strong suggestions; in fact `\penalty-9999` will be quite okay. Just do not make the break mandatory. We have not applied any of Michael's solutions here, since we feel that the problem is exiguous, and `reledmac` is quite baroque enough already. If you think you are having this problem, look up Michael's solutions.

One more thing; we set `\leftskip` and `\rightskip` to zero. This has the effect of neutralizing any such skips which may apply to the main text (cf. XII.6.2 p. 167 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.

```
2808 \newcommand*{\paravfootnote}[2]{%
2809   \csuse{Xbeforeinserting@#1}%
2810   \insert\csname #1footins\endcsname
2811   \bgroup
```

²⁹Michael Downes, 'Line Breaking in `\unhboxed` Text', *TUGboat* **11** (1990), pp. 605–612.

³⁰See *The TeXbook*, p. 455 (editions after January 1990).

³¹Wayne supplied his own macros to do this, but since they were almost identical to Michael's, Peter Wilson have used the latter's `\Xunvzx` macro since it is publicly documented.

³²'Line Breaking', p. 610.

```

2812 \csuse{Xnotefontsize@#1}
2813 \footsplitskips
2814 \setbox0=\vbox{\hsize=\maxdimen%
2815   \let\bidir@RTL@everypar\@empty%
2816   \insert@Xtxtbeforenotes{#1}%
2817   \noindent\csuse{Xhooknote@#1}%
2818   \csname #1footfmt\endcsname #2{#1}}%
2819 \setbox0=\hbox{\Xunvxh{0}{#1}}%
2820 \dp0=0pt
2821 \ht0=\csname #1footfudgefactor\endcsname\wd0
2822 %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

2823 \if@RTL\noindent \leavevmode\fi\box0%
2824 \penalty0
2825 \egroup}
2826
2827 %

```

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.

```

2828 \newcommand*{\mpparavfootnote}[2]{%
2829   \global\setbox\@nameuse{mp#1footins}\vbox{%
2830     \unvbox\@nameuse{mp#1footins}%
2831     \csuse{Xnotefontsize@#1}
2832     \footsplitskips
2833     \setbox0=\vbox{\hsize=\maxdimen%
2834       \let\bidir@RTL@everypar\@empty%
2835       \insert@Xtxtbeforenotes{#1}%
2836       \noindent\color@begingroup%
2837       \csuse{Xhooknote@#1}%
2838       \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
2839     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2840     \dp0=\z@
2841     \ht0=\csname #1footfudgefactor\endcsname\wd0
2842     \box0
2843     \penalty0
2844   }}

```

2845
2846 %

\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`.

```
2847 \newcommand*{\Xunvxh}[2]{%
2848   \setbox0=\vbox{\unvbox#1%
2849     \global\setbox1=\lastbox}%
2850   \unhbox1
2851   \unskip           % remove \rightskip,
2852   \unskip           % remove \parfillskip,
2853   \unpenalty        % remove \penalty of 10000,
2854   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
2855
2856 %
```

\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).

```
2857 \newcommand*{\parafootfmt}[4]{%
2858   \Xstorelineinfo{#1}{#4}%
2859   \Xinsertparafootsep{#4}%
2860   \ledsetnormalparstuff@common%
2861   \printlinefootnote{#1}{#4}%
2862   \print@lemma{#1}{#2}{#4}%
2863   \csuse{Xwrapcontent@#4}{#3}%
2864   \penalty-10 }
2865 %
```

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.

```
2866 \newcommand*{\parafootgroup}[1]{%
2867   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
```

```

2868 \unvbox\csname #1footins\endcsname
2869 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2870 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2871 \makehboxofhboxes
2872 \setbox0=\hbox{\unhbox0 \removehboxes}%
2873 \csuse{Xbhookgroup@#1}%
2874 \csuse{Xnotefontsize@#1}%
2875 \unhbox0\par%
2876 \global\hsize=\old@hsize%
2877 }%
2878
2879 %

```

`\mpparafootgroup` The minipage version.

```

2880 \newcommand*\mpparafootgroup}[1]{%
2881 \setXnoteswidthliketwocolumns@{#1}%
2882 \vskip\skip\@nameuse{mp#1footins}
2883 \ifl@dpairing\ifparledgroup%
2884 \leavevmode\marks\parledgroup@{begin}%
2885 \marks\parledgroup@series{#1}%
2886 \marks\parledgroup@type{Xfootnote}%
2887 \fi\fi\normalcolor
2888 \ifparledgroup%
2889 \ifl@dpairing%
2890 \else%
2891 \setXnoteswidthliketwocolumns@{#1}%
2892 \setXnotespositionliketwocolumns@{#1}%
2893 \print@Xfootnoterule{#1}%
2894 \fi%
2895 \else%
2896 \setXnoteswidthliketwocolumns@{#1}%
2897 \setXnotespositionliketwocolumns@{#1}%
2898 \print@Xfootnoterule{#1}%
2899 \fi%
2900 \unvbox\csname mp#1footins\endcsname
2901 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2902 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2903 \makehboxofhboxes
2904 \setbox0=\hbox{\unhbox0 \removehboxes}%
2905 \csuse{Xbhookgroup@#1}%
2906 \csuse{Xnotefontsize@#1}%
2907 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2908 \unhbox0\par}}
2909
2910 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes11 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes12 \loop
2913 \unpenalty
2914 \setbox2=\lastbox
2915 \ifhbox2
2916 \setbox0=\hbox{\box2\unhbox0}%
2917 \repeat}
2918
2919 \newcommand*{\removehboxes}{\setbox0=\lastbox
2920 \ifhbox0{\removehboxes}\unhbox0 \fi}
2921
2922 %

```

Insertion of the footnotes separator The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num23 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep24 \iflabeledRcol{%
2925 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
2926 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2927 {\ifcsequal{prevline#1}{lineinfo@}%
2928 {\ifcseempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
2929 {\csuse{Xparafootsep@#1}}}%
2930 }%
2931 {\csuse{Xparafootsep@#1}}%
2932 }%
2933 {}%
2934 \global\csname #1prevpage@numR\endcsname=\page@numR%
2935 \else%
2936 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
2937 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2938 {\ifcsequal{prevline#1}{lineinfo@}%
2939 {\ifcseempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
2940 {\csuse{Xparafootsep@#1}}}%
2941 }%
2942 {\csuse{Xparafootsep@#1}}%
2943 }%
2944 {}%
2945 \global\csname #1prevpage@num\endcsname=\page@num%
2946 \fi%
2947 }
2948 %

```

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 \splitoff, 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 \@@line.

We do not call directly \rigidbalance, but we call \Xrigidbalance for critical notes and \rigidbalanceX for familiar notes. Both of them call \rigidbalance.

```

2949 \newcount\@k \newdimen\@h
2950 \newcommand*\Xrigidbalance}[3]{%
2951   \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
2952   \rigidbalance{#1}{#2}{#3}%
2953 }%
2954
2955 \newcommand*\rigidbalanceX}[3]{%
2956   \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
2957   \rigidbalance{#1}{#2}{#3}%
2958 }%
2959
2960 \newcommand*\rigidbalance}[3]{%
2961   \setbox0=\box#1 \@k=#2 \@h=#3%
2962   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
2963     \valign{##\vfil\cr\dosplits}}}%
2964
2965 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splitoff
2966   \global\advance\@k-1\cr\dosplits\fi}
2967
2968 \newcommand*\splitoff{\dimen0=\ht0
2969   \divide\dimen0 by\@k \advance\dimen0 by\@h
2970   \setbox2 \vsplit0 to \dimen0
2971   \unvbox2 }
2972
2973 %

```

Three columns

```

\Xarrangement@threecol 2974 \newcommand*\Xarrangement@threecol}[1]{%
2975   \csgdef{series@display#1}{threecol}
2976   \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
2977   \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
2978   \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
2979   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2980   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2981   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2982   \threecolfootsetup{#1}
2983 %

```


The additional setup for minipages.

```

2984 \ifnoledgroup@else
2985   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2986   \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
2987   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2988   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2989   \mpthreecolfootsetup{#1}
2990 \fi
2991 }
2992
2993 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XII.6.2 p. 166 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.

```

2994 \newcommand*{\threecolfootsetup}[1]{%
2995   \count\csname #1footins\endcsname 333
2996   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
2997   \multiply\dimen\csname #1footins\endcsname \thr@@}
2998 %

```

`\mpthreecolfootsetup` The setup for minipages.

```

2999 \newcommand*{\mpthreecolfootsetup}[1]{%
3000   \count\csname mp#1footins\endcsname 333
3001   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3002
3003 %

```

`\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).

```

3004 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnote}[2]{%
3005   \iftoggle{Xgroupbyline@#1}{%
3006     \prepare@Xgroupbyline{#1}{#2}{\threecolvfootnote@inserted}%
3007   }%
3008   {%
3009     \X@beforeinsertion{#1}%
3010     \insert\csname #1footins\endcsname{%
3011       \threecolvfootnote@inserted{#1}{#2}%
3012     }%
3013   }%
3014 }%
3015 %

```

```

\threecolvfootnote@inserted 16 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
threecolvfootnote@inserted}[2]{%
3017   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3018   \noindent\csuse{Xhooknote@#1}%
3019   \csuse{Xnotefontsize@#1}%
3020   \footplitskips%
3021   \csname #1footfmt\endcsname #2{#1}%
3022 }%
3023 %

```

\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.

```

3024 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
3025   \threecol@begin@insert{#4}%
3026   \hspace{\parindent}%
3027   \printlinefootnote{#1}{#4}%
3028   \print@lemma{#1}{#2}{#4}%
3029   \csuse{Xwrapcontent@#4}{#3}%
3030   \nottoggle{Xgroupbyline@#4}%
3031   {\strut\par\allowbreak}%
3032   {}%
3033 }%
3034 %

```

\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`.

```

3035 \newcommand{\threecol@begin@insert}[1]{%
3036   \normal@pars%
3037   \nottoggle{Xgroupbyline@#1}%
3038   {\hspace \csuse{Xhsizethreecol@#1}}%
3039   }%
3040   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3041   \tolerance=5000%
3042   \hangindent=\csuse{Xhangindent@#1}%
3043   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3044   \@tempdima=\parindent%
3045   \csuse{Xcolalign@#1}%
3046   \parindent=\@tempdima%
3047   \strut%
3048 }%
3049 %

```

\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.

```

3050 \newcommand*\threecolfootgroup[1]{%
3051   \csuse{Xhookgroup@#1}\par%
3052   \splittopskip=\ht\strutbox
3053   \expandafter
3054   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
3055 %

```

\mpthreecolfootgroup The setup for minipages.

```

3056 \newcommand*\mpthreecolfootgroup[1]{%
3057   \vskip\skip\@nameuse{mp#1footins}
3058   \ifl@dpairing\ifparledgroup%
3059     \leavevmode\marks\parledgroup@{begin}%
3060     \marks\parledgroup@series{#1}%
3061     \marks\parledgroup@type{Xfootnote}%
3062   \fi\fi\normalcolor
3063   \ifparledgroup%
3064     \ifl@dpairing%
3065     \else%
3066       \setXnoteswidthliketwocolumns@{#1}%
3067       \setXnotespositionliketwocolumns@{#1}%
3068       \print@Xfootnoterule{#1}%
3069     \fi%

```

```

3070 \else%
3071 \setXnoteswidthliketwocolumns@{#1}%
3072 \setXnotespositionliketwocolumns@{#1}%
3073 \print@Xfootnoterule{#1}%
3074 \fi%
3075 \csuse{Xbhookgroup@#1}\par%
3076 \splittopskip=\ht\strutbox
3077 \expandafter
3078 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3079
3080 %

```

Two columns

```

\Xarrangement@twocol 3081 \newcommand*\Xarrangement@twocol}[1]{%
3082 \csgdef{series@display#1}{twocol}
3083 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3084 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3085 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3086 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3087 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3088 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3089 \twocolfootsetup{#1}
3090 %

```

The additional setup for minipages.

```

3091 \ifnoledgroup@else
3092 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3093 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
3094 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3095 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3096 \mptwocolfootsetup{#1}
3097 \fi
3098 }
3099
3100 %

```

`\twocolfootsetup` Here is a series of macros which are very similar to their three-column counterparts. In this case, each note is assumed to contribute only a half a line of text. And the notes are set in columns giving a gap between them of one tenth of the `\hsize`.

```

\Xarrangement@twocol 3101 \newcommand*\Xarrangement@twocol}[1]{%
\Xarrangement@twocol 3102 \count\csname #1footins\endcsname 500
\Xarrangement@twocol 3103 \csxdef{default@#1footins}{500}%
\Xarrangement@twocol 3104 \multiply\dimen\csname #1footins\endcsname \tw@}
\Xarrangement@twocol 3105 %

```

```

3106 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnote}[2]{%
3107   \iftoggle{Xgroupbyline@#1}{%
3108     \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3109   }{%
3110     \X@beforeinsertion{#1}%
3111     \insert\csname #1footins\endcsname{%
3112       \twocolvfootnote@inserted{#1}{#2}%
3113     }%
3114   }%
3115 }%
3116 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
twocolvfootnote@inserted}[2]{%
3117   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3118   \noindent\csuse{Xhooknote@#1}%
3119   \csuse{Xnotefontsize@#1}%
3120   \footsplitskips%
3121   \csname #1footfmt\endcsname #2{#1}%
3122 }%
3123 %

3124 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmt}[4]{% 4th
arg is optional, for backward compatibility
3125   \twocol@begin@insert{#4}%
3126   \hspace{\parindent}%
3127   \printlinefootnote{#1}{#4}%
3128   \print@lemma{#1}{#2}{#4}%
3129   \csuse{Xwrapcontent@#4}{#3}%
3130   \nottoggle{Xgroupbyline@#4}%
3131     {\strut\par\allowbreak}%
3132   {}%
3133 }%
3134 %

3135 \newcommand{\twocol@begin@insert}[1]{%
3136   \normal@pars%
3137   \hsize \csuse{Xhsizetwocol@#1}%
3138   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3139   \tolerance=5000%
3140   \hangindent=\csuse{Xhangindent@#1}%
3141   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3142   \@tempdima=\parindent%
3143   \csuse{Xcolalign@#1}%
3144   \parindent=\@tempdima%
3145   \strut%
3146 }%
3147
3148 \newcommand*{\twocolfootgroup}[1]{%
3149   \csuse{Xhookgroup@#1}\par%
3150   \splittopskip=\ht\strutbox
3151   \expandafter

```

```

3152 \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
3153
3154 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3155 \newcommand*\mptwocolfootsetup}[1]{%
3156   \count\csname mp#1footins\endcsname 500
3157   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3158 %

3159 \newcommand*\mptwocolfootgroup}[1]{%
3160   \vskip\skip\@nameuse{mp#1footins}
3161   \ifl@dpairing\ifparledgroup%
3162     \leavevmode\marks\parledgroup@{begin}%
3163     \marks\parledgroup@series{#1}%
3164     \marks\parledgroup@type{Xfootnote}%
3165   \fi\fi\normalcolor
3166   \ifparledgroup%
3167     \ifl@dpairing%
3168     \else%
3169       \setXnoteswidthliketwocolumns@{#1}%
3170       \setXnotespositionliketwocolumns@{#1}%
3171       \print@Xfootnoterule{#1}%
3172     \fi%
3173   \else%
3174     \setXnoteswidthliketwocolumns@{#1}%
3175     \setXnotespositionliketwocolumns@{#1}%
3176     \print@Xfootnoterule{#1}%
3177   \fi%
3178   \csuse{Xbhookgroup@#1}\par%
3179   \splittopskip=\ht\strutbox
3180   \expandafter
3181   \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3182
3183 %

```

XII.7 Critical notes presentation

Here, we define some commons macro which are used in order to print a critical notes, that is a note with 1) line number 2) lemma 3) lemma separator 4) text associated to the lemma.

XII.7.1 Font tools

`\endashchar`

`\fullstop`

`\rbracket`

The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals does not contain an en-dash or square brackets, and its period and comma are in odd locations.

To allow use of the standard footnote macros with such fonts, we use the following macros for certain characters.

The `\endashchar` macro is simply an en-dash from the normal font and is immune to changes in the surrounding font. The same goes for the full stop. These two are used in `\printlines`. The right bracket macro is the same again; it crops up in `\normalfootfmt` and the other footnote macros for controlling the format of the footnotes.

Note that these commands are not directly called by `reledmac`, but are enclosed as default value of specific hooks. Consequently, people should not redefine them, but use instead the `\Xlinerangeseparator`, `\Xendlinerrangeseparator`, `\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.

```

3184 \def\endashchar{\textnormal{--}}
3185
3186 \newcommand*{\fullstop}{\textnormal{.}}
3187 \def\Xsublinesep@side{\fullstop}
3188
3189 \newcommand*{\rbracket}{\textnormal{%
3190   \csuse{text}\csuse{footnote@lang}}{%
3191     \ifluatex%
3192       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]}{\thinspace
3193     }}%
3194     \else%
3195       \thinspace}%
3196     \fi}%
3197 }
3198
3199 %

```

XII.7.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

3200 \newcommand{\printpstart}[0]{%
3201   \ifbool{expr}{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3202     l@dprintingcolumns}}{%
3203     \ifledRcol%
3204       \thepstartR%
3205     \else%
3206       \thepstartL%
3207     \fi%
3208   }{%
3209     \thepstart%
3210   }%
3211 }

```

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).

```

3212 %
3213 \newcommand{\print@lemma}[3]{%
3214   \bgroup%
3215   \nottoggle{Xlemmadisablefontselection@#3}%
3216   {\select@lemmafnt#1|}%
3217   }%
3218   \bgroup%
3219   \csuse{Xlemmafnt@#3}%Deprecated
3220   \csuse{Xwraplemma@#3}{#2}%
3221   \egroup%
3222   \egroup%
3223   \iftoggle{nosep@}{%
3224     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3225     \relax%
3226   }%
3227   {\ifcsemtty{Xlemmaseparator@#3}%
3228     {%
3229       \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3230       \relax%
3231     }%
3232     {%
3233       \nobreak%
3234       \hskip\csuse{Xbeforelemmaseparator@#3}%
3235       \csuse{Xlemmaseparator@#3}%
3236       \hskip\csuse{Xafterlemmaseparator@#3}%
3237       \relax%
3238     }%
3239   }%
3240 }%
3241 %

```

XII.7.4 Line number printing

`\Xstorelineinfo` The `\Xstorelineinfo` macro is used to store some data about line number of the current critical footnote, data which will be reused later for the `\Xnumberonlyfirstinline` and related setting.

#1 footnote specification for the current footnote ; #2 footnote series.

```

3242 \newcommand{\Xstorelineinfo}[2]{%
3243   \l@dp@rsefootspec#1|}%
3244   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3245     \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3246     \l@dparsedendline - \l@dparsedendsub}%
3247   }%
3248   {%

```



```

3248 \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3249 }%
3250 }%
3251 %

```

\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`.

```

3252 \newcommand{\printlinefootnote}[2]{%
3253   \iftoggle{nonum@}{%Try if the line number must printed for this specific
not (by default, yes)
3254     \hspace{\csuse{Xinplaceofnumber@#2}}%
3255   }%
3256   {%
3257     {%
3258       \iftoggle{Xnonumber@#2}%Try if the line number must printed (by
default, yes)
3259       {%
3260         \hspace{\csuse{Xinplaceofnumber@#2}}%
3261       }%
3262       {%
3263         {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the
line number must be printed only in the first time.
3264           {%
3265             \ifcsdef{prevline#2}%
3266               {%Be sure the \prevline exists.
3267               \ifcsequal{prevline#2}{\lineinfo@}%Try it
3268               {%
3269                 \ifcseempty{Xsymlinenum@#2}%
3270                 {%
3271                   \hspace{\csuse{Xinplaceofnumber@#2}}%
3272                 }%
3273                 {\printsymlinefootnotearea{#2}}%
3274               }%
3275               {%
3276                 \printlinefootnotearea{#1}{#2}%
3277               }%
3278             }%
3279             {%
3280               \printlinefootnotearea{#1}{#2}%
3281             }%
3282           }%
3283           {%
3284             \printlinefootnotearea{#1}{#2}%
3285           }%
3286           \csxdef{prevline#2}{\lineinfo@}%
3287         }%
3288       }%

```

```

3289 }%
3290 }%
3291 }
3292 %

```

\printsymlinefootnotearea This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

3293 \newcommand{\printsymlinefootnotearea}[1]{%
3294   \hspace{\csuse{Xbeforeysmlinenumber@#1}}%
3295   \csuse{Xnotenumfont@#1}%
3296   \ifdimequal{\csuse{Xboxsymmlinenumber@#1}}{\z@}%
3297     {\csuse{Xsymmlinenumber@#1}}%
3298     {\hbox to \csuse{Xboxsymmlinenumber@#1}%
3299       {\csuse{Xsymmlinenumber@#1}\hfill}%
3300     }%
3301   \hspace{\csuse{Xaftersymmlinenumber@#1}}%
3302 }%
3303 %

```

\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.)

```

3304 \newcommand{\printlinefootnotearea}[2]{%
3305   \printXbeforenumber{#2}%
3306   \csuse{Xnotenumfont@#2}%
3307   \boxfootnotenumbers{#1}{#2}%
3308   \printXafternumber{#2}%
3309 }%
3310 %

```

\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.

```

3311 \newcommand{\boxfootnotenumbers}[2]{%
3312   \ifdimequal{\csuse{Xboxlinenum@#2}}{\Opt}{%
3313     \printlinefootnotenumbers{#1}{#2}%
3314   }%
3315   {%
3316     \hbox to \csuse{Xboxlinenum@#2}%
3317     {%
3318       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3319       \printlinefootnotenumbers{#1}{#2}%
3320       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3321     }%
3322   }%
3323 }%
3324 %

```

`\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.

```

3325 \newcommand{\printlinefootnotenumbers}[2]{%
3326   \xdef\@currentseries{#2}%
3327   \ifboolexpr{%
3328     (togl{Xpstart@#2} and bool{numberpstart})%
3329     or togl{Xpstarteverytime@#2}}%
3330   {\printpstart}{}%
3331   \iftoggle{Xstanza@#2}{%
3332     \ifnumberstanza%
3333       \printstanza%
3334       \csuse{Xstanzaseparator@#2}%
3335     \fi%
3336   }{}%
3337   \iftoggle{Xonlypstart@#2}{}{%
3338     \csuse{Xtxtbeforenumber@#2}%
3339     \printlines#1|\ifledRcol@\@Rlineflag\fi}%
3340   }%
3341   %

```

`\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.)

```

3342 \newcommand{\printXbeforenumber}[1]{%
3343   \hspace{\csuse{Xbeforenumber@#1}}%
3344   }%
3345   %

```

`\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

```

3346 \newcommand{\printXafternumber}[1]{%
3347   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3348   \hspace{\csuse{Xafternumber@#1}}%
3349   }%
3350   %

```

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. 97: the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma.

edmac' creator have defined six boolean in order to know which component of line number description we have to print:

- `\ifl@d@pnum` for page numbers;

- `\ifl@d@ssub` for starting sub-line;
- `\ifl@d@elin` for ending line;
- `\ifl@d@esl` for ending sub-line; and
- `\ifl@d@dash` for the dash between the starting and ending groups.

There is no boolean for the line number because it is always printed.

Maïeul Rouquette has added `\ifl@d@xtwoline` and `\ifl@d@xmorethantwoline` to print a symbol which stands for “and subsequent” when there are two, three or more lines.

```

\ifl@d@pnum51 \newif\ifl@d@pnum
\ifl@d@ssub52 \newif\ifl@d@ssub
\ifl@d@elin53 \newif\ifl@d@elin
\ifl@d@esl54 \newif\ifl@d@esl
\ifl@d@dash55 \newif\ifl@d@dash
\ifl@d@xtwoline56 \newif\ifl@d@xtwoline%
\ifl@d@xmorethantwoline57 \newif\ifl@d@xmorethantwoline%
58 %

```

```

\l@dp@rsefootsspec \l@dp@rsefootsspec{<spec>}{<lemma>}{<text>} parses a footnote specification. <lemma>
\l@dp@rsefootsspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@dp@rsefootsspec ber and lemma font specifier in \l@d@nums style format. The real work is done by
\l@dp@rsefootsspec \l@dp@rsefootsspec which defines macros holding the numeric values. In many cases,
\l@dp@rsefootsspec this last command is called directly. Just a reminder of the arguments:
\l@dp@rsefootsspec \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dp@rsefootsspec \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@dp@rsefootsspec
59 \newcommand*{\l@dp@rsefootsspec}[3]{\l@dp@rsefootsspec#1|}
60 \def\l@dp@rsefootsspec#1|#2|#3|#4|#5|#6|#7|{%
61 \gdef\l@dp@rsefootsspec#1}%
62 \gdef\l@dp@rsefootsspec#2}%
63 \gdef\l@dp@rsefootsspec#3}%
64 \gdef\l@dp@rsefootsspec#4}%
65 \gdef\l@dp@rsefootsspec#5}%
66 \gdef\l@dp@rsefootsspec#6}%
67 }
68 %

```

Initialise the several number value macros.

```

69 \def\l@dp@rsefootsspec{0}%
70 \def\l@dp@rsefootsspec{0}%
71 \def\l@dp@rsefootsspec{0}%
72 \def\l@dp@rsefootsspec{0}%
73 \def\l@dp@rsefootsspec{0}%
74 \def\l@dp@rsefootsspec{0}%
75
76 %

```

`\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`.

```
3377 \newcommand*\setprintlines}[6]{%
3378   \l@dpnumfalse \l@ddashfalse
3379   %
```

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

```
3380   \ifbypage@
3381     \ifnum#4=#1 \else
3382       \l@dpnumtrue
3383       \l@ddashtrue
3384     \fi
3385   \fi
3386   %
```

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.

```
3387   \ifl@dpnum \l@d@elintrue \else \l@d@elinfalse \fi
3388   \ifnum#2=#5 \else
3389     \l@d@elintrue
3390     \l@ddashtrue
3391   \fi
3392   %
```

We print the starting sub-line if it is nonzero.

```
3393   \l@d@ssubfalse
3394   \ifnum#3=0 \else
3395     \l@d@ssubtrue
3396   \fi
3397   %
```

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.

```
3398   \l@d@eslfalse
3399   \ifnum#6=0 \else
3400     \ifnum#6=#3
3401       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3402     \else
3403       \l@d@esltrue
3404       \l@ddashtrue
3405     \fi
3406   \fi%
3407   %
```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

3408 \ifl@dash%
3409 \ifbool{expr{togl{fulllines@} or test{\ifcsempy{Xtwolines@
@currentseries}}}%
3410 {}%
3411 {%
3412 \setistwofollowinglines{#1}{#2}{#4}{#5}%
3413 \ifbool{expr{%
3414 (%
3415 togl {Xtwolinesbutnotmore@}\@currentseries}%
3416 and not%
3417 (%
3418 bool {istwofollowinglines@}%
3419 )%
3420 }%
3421 or%
3422 (%
3423 (not test{\ifnumequal{#1}{#4}})%
3424 and togl{Xtwolinesonlyinsamepage@}\@currentseries}%
3425 )%
3426 }%
3427 {}%
3428 {%
3429 \l@d@dashfalse%
3430 \l@d@Xtwolinestrue%
3431 \l@d@elinfalse%
3432 \l@d@eslfalse%
3433 \ifcsempy{Xmorethantwolines@}\@currentseries}%
3434 {}%
3435 {\ifistwofollowinglines@}\else%
3436 \l@d@Xmorethantwolinestrue%
3437 \fi%
3438 }%
3439 }%
3440 }%
3441 \fi%
3442 %

End of \setprintlines.

3443 }%
3444 %

```

\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.

```

3445 \newif\ifistwofollowinglines@%
3446 \newcommand{\setistwofollowinglines}[4]{%
3447   \ifcsdef{lastlinenumberon@#1}%
3448     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
3449     {\numdef{\tmp}{0}}}%
3450   \istwofollowinglines@false%
3451   \ifnumequal{#4-#2}{1}%
3452   {\istwofollowinglines@true}%
3453   {\ifbypage@%
3454     \ifnumequal{#3-#1}{1}%
3455     {%
3456       \ifnumequal{#2}{\tmp}%
3457       {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
3458     }%
3459   }%
3460   {%
3461     \fi%
3462   }%
3463 }%
3464 %

```

`\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

```

3465 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3466   \begingroup%
3467   %

```

If we use LuaTeX, ensure we use good text's direction.

```

3468   \ifluatex%
3469     \edef\@tmp{\the\textdir}%
3470     \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
    spurious space (bug #397)
3471   \fi%
3472   %

```

Decide which part of line number components we will print.

```

3473 \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3474 %

```

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.

```

3475 \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3476 {\bgroup}%
3477 {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
hfill}%
3478 \ifl@d@pnum%
3479 \wrap@edcrossref{\@this@crossref@start}{#1}%
3480 \csuse{Xpagelinesep@\@currentseries}%
3481 \fi%
3482 \wrap@edcrossref{\@this@crossref@start}{%
3483 \linenumrep{#2}%
3484 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3485 }%
3486 \ifl@d@ssub%
3487 \csuse{Xsublinesep@\@currentseries}%
3488 \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3489 \fi
3490 \egroup%
3491 %

```

Then print the dash + end line number, or the range symbol.

```

3492 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3493 {\bgroup}%
3494 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
3495 \ifl@d@Xtwolines%
3496 \ifl@d@Xmorethantwolines%
3497 \csuse{Xmorethantwolines@\@currentseries}%
3498 \else%
3499 \csuse{Xtwolines@\@currentseries}%
3500 \fi%
3501 \else%
3502 \ifl@d@dash%
3503 \ifdefined\linerangesep%
3504 \linerangesep%
3505 \else%
3506 \csuse{Xlinerangeseparator@\@currentseries}%
3507 \fi%
3508 \fi%
3509 \ifl@d@pnum%
3510 \wrap@edcrossref{\@this@crossref@end}{#4}%
3511 \csuse{Xpagelinesep@\@currentseries}%
3512 \fi%
3513 \ifl@d@elin%

```



```

3514 \wrap@edcrossref{\@this@crossref@end}{%
3515 \linenumrep{#5}%
3516 \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3517 }%
3518 \fi%
3519 \ifl@d@esl%
3520 \ifl@d@elin%
3521 \csuse{Xsublinesep@\@currentseries}%
3522 \fi%
3523 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3524 \fi%
3525 \fi%
3526 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3527 {}%
3528 {\hfill}%Prevent underfull hbox
3529 \egroup%
3530 \endgroup%
3531 }%
3532 %

```

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`.

```

3533 \newcommand{\prepare@Xgroupbyline}[3]{%
3534 \iftoggle{Xgroupbylineseparatetwolines@#1}{%
3535 \l@dparsedefootspec#2%
3536 \ifcsdef{#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3537 {%
3538 \csgappto%
3539 {#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3540 {%
3541 \ifcsemt{Xsymlinenum@#1}%
3542 {\csuse{Xparafootsep@#1}}%
3543 {}%
3544 #3{#1}{#2}%
3545 \hskip\csuse{Xafternote@#1}\relax%
3546 }%
3547 }%
3548 {%
3549 \csdef%
3550 {#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3551 {%
3552 #3{#1}{#2}%
3553 \hskip\csuse{Xafternote@#1}\relax%
3554 }%

```

```

3555 }%
3556 \listcsxadd{#1@forinserting}{\l@dparsedendpage-\l@dparsedendline-\l@dparsedendsub}%
3557 }{%
3558 \ifcsdef{#1@forinserting@all}{%
3559 \csgappto%
3560 {#1@forinserting@all}%
3561 {%
3562 \ifcseempty{Xsymlinenum@#1}%
3563 {\csuse{Xparafootsep@#1}}%
3564 }%
3565 #3{#1}{#2}%
3566 \hskip\csuse{Xafternote@#1}\relax%
3567 }%
3568 }%
3569 {%
3570 \csdef%
3571 {#1@forinserting@all}%
3572 {%
3573 #3{#1}{#2}%
3574 \hskip\csuse{Xafternote@#1}\relax%
3575 }%
3576 }%
3577 \listcsgadd{#1@forinserting}{all}%
3578 }%
3579 }%
3580 %

```

XIII Familiar footnotes

XIII.1 Adjacent footnotes

The original edmac provided users with five series of critical footnotes (`\Afootnote` `\Bfootnote` `\Cfootnote` `\Dfootnote` `\Efootnote`), and \LaTeX provides a single numbered footnote. The `reledmac` package uses the edmac mechanism to provide six series of numbered footnotes.

First, though, the `footmisc` package has an option whereby two or more consecutive `\footnotes` have their marks separated by commas. This seemed to Peter Wilson such a useful ability that it was provided automatically by `eledmac`.

Maïeul Rouquette has maintained this feature in `reledmac`, despite he thought that is not directly in relationship with the aim of `reledmac`.

`\multiplefootnotemarker` These macros may have been defined by the `memoir` class, are provided by the `footmisc` package and perhaps by other footnote packages. That is why we use `\providecommand` and not `\newcommand`.

```

3581 \providecommand*{\multiplefootnotemarker}{3sp}
3582 \providecommand*{\multfootsep}{\textsuperscript{\normalfont,}}

```

```
3583
3584 %
```

\m@mmf@prepare A pair of self-cancelling kerns. This may have been defined in the memoir class.

```
3585 \providecommand*\m@mmf@prepare{%
3586   \kern-\multiplefootnotemarker
3587   \kern\multiplefootnotemarker\relax}
3588 %
```

\m@mmf@check This may have been defined in the memoir class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```
3589 \providecommand*\m@mmf@check{%
3590   \ifdim\lastkern=\multiplefootnotemarker\relax
3591     \edef\x@sf{\the\spacefactor}%
3592     \unkern
3593     \multfootsep
3594     \spacefactor\x@sf\relax
3595   \fi}
3596
3597 %
```

We have to modify `\@footnotetext` and `\@footnotemark`. However, if memoir is used the modifications have already been made.

```
3598 \@ifclassloaded{memoir}{%
3599 %
```

\@footnotetext Add `\m@mmf@prepare` at the end of `\@footnotetext`.

```
3600 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
3601 %
```

\@footnotemark Modify `\@footnotemark` to cater for adjacent footnotes.

```
3602
3603 \patchcmd{\@footnotemark}
3604   {\nobreak}
3605   {\m@mmf@check
3606    \nobreak
3607   }
3608   {}{}
3609 \patchcmd{\@footnotemark}
3610   {\@makefnmark}
3611   {\@makefnmark
3612    \m@mmf@prepare
3613   }
3614   {}{}
3615 %
```

Finished the modifications for the non-memoir case.

```
3616 }
3617
3618 %
```

XIII.2 Regular footnotes for numbered texts

`\l@doldold@footnotetext` In order to enable the regular `\footnotes` in numbered text we have to play around
`\@footnotetext` with its `\@footnotetext`, using different forms for when in numbered or regular text.

```
3619 \pretocmd{\@footnotetext}{%
3620   \ifnumberedpar@
3621   \edtext{\l@dbfnote{#1}}%
3622   \else
3623   }{}{}
3624 \apptocmd{\@footnotetext}{\fi}{}{}%
3625 %
```

```

\l@dbfnote \l@dbfnote adds the footnote to the insert list, and \v1@dbfnote calls the original
\v1@dbfnote \@footnotetext. We also patch \footnote in order to get the correct footnote
\v1@dbfnote numbers when typesetting parallel texts. This is moved into a \get@fnmark command.
\footnote
\get@fnmark
\get@thisfootnote
3626
3627 \patchcmd%
3628   {\footnote}%
3629   {\stepcounter\@mpfn}%
3630   {%
3631     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3632       l@dprintingcolumns}}{%
3633       \global\advance\footnote@reading by \one%
3634       \get@thisfootnote%
3635       \get@fnmark{\thisc@footnote}%
3636       \ifcsdef{footnotereading\the\footnote@reading=typeset}%
3637         {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
3638           typeset}}}%
3639         {\setcounter{\@mpfn}{\footnote@reading}}%
3640       }{}%
3641       \stepcounter\@mpfn%
3642     }{}%
3643     {}
3644
3645 \newcommand{\get@thisfootnote}{%
3646   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3647     l@dprintingcolumns}}{%
3648     \protected@xdef\thisc@footnote{\the\footnote@reading}%
3649   }%
```

```

3648 }{%
3649   \protected@xdef\thisc@footnote{\the\c@footnote}%
3650 }%
3651 }%
3652
3653 \newcommand{\l@dbfnote}[1]{%
3654   \get@thisfootnote%
3655   \gdef\@tag{#1\relax}%
3656   \ifledRcol%
3657     \xright@appenditem{%
3658       \ifdefined\Hy@footnote@currentHref%
3659         \noexpand\def\noexpand\Hy@footnote@currentHref{
Hy@footnote@currentHref}%
3660       \fi%
3661       \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
3662     }%
3663     \to\inserts@listR
3664     \global\advance\insert@countR \@ne%
3665   \else%
3666     \xright@appenditem{%
3667       \ifdefined\Hy@footnote@currentHref%
3668         \noexpand\def\noexpand\Hy@footnote@currentHref{
Hy@footnote@currentHref}%
3669       \fi%
3670       \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
3671     }%
3672     \to\inserts@list
3673     \global\advance\insert@count \@ne%
3674   \fi
3675   \ignorespaces%
3676 }%
3677
3678 \newcommand{\get@fnmark}[1]{%
3679   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}%
3680     {%
3681       \stepcounter{footnote@typeset}%
3682       \setcounter{footnote}{\c@footnote@typeset}%
3683       \immediate\write\@mainaux{%
3684         \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
3685       }%
3686       \def\@thefnmark{\thefootnote}%
3687     }%
3688   {%
3689     \setcounter{footnote}{#1}%
3690     \def\@thefnmark{\thefootnote}%
3691   }%
3692 }%
3693
3694 \newcommand{\vl@dbfnote}[2]{%

```

```

3695 \get@fnmark{#2}%
3696 \@footnotetext{#1}%
3697 }%
3698 %

```

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`

```

3699 \newcommand*\prebodyfootmark{%
3700 \leavevmode
3701 \ifhmode
3702 \edef\x@sf{\the\spacefactor}%
3703 \m@mmf@check
3704 \nobreak
3705 \fi}
3706 \newcommand*\postbodyfootmark{%
3707 \m@mmf@prepare
3708 \ifhmode\spacefactor\x@sf\fi\relax}
3709 %
3710 %

```

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.

```

3711 \newcommandx*\arrangementX}[2][1,usedefault]{%
3712 \def\do##1{%
3713 \csname arrangementX@#2\endcsname{##1}%
3714 }%
3715 \ifstrepty{#1}%
3716 {%
3717 \dolistloop{\@series}%
3718 }%
3719 {
3720 \docsvlist{#1}%
3721 }%
3722 }%
3723 %

```

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.

```

3724 \newcommand*{\normal@footnotemarkX}[1]{%
3725   \prebodyfootmark
3726   \wrapped@bodyfootmarkX{#1}%
3727   \postbodyfootmark}
3728
3729 %

```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{<series>}` *really* typesets the in-text marker. The style is the normal superscript.

```

3730 \newcommand*{\normalbodyfootmarkX}[1]{%
3731   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}%
3732 %

```

`\normalvfootnoteX` `\normalvfootnoteX{<series>}{<text>}` does the `\insert` for the `<series>` and calls the series' `\footfmt...` to format the `<text>`.

```

3733 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalvfootnoteX}[2]{%
3734   \csuse{beforeinsertingX@#1}%
3735   \insert\@nameuse{footins#1}\bgroup
3736     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3737     \noindent\csuse{bhooknoteX@#1}%
3738     \csuse{notefontsizeX@#1}%
3739     \footsplitskips
3740     \ifl@dpairing\ifl@dpadding\else%
3741       \setnoteswidthliketwocolumnsX@{#1}%
3742     \fi\fi%
3743     \setnotesXpositionliketwocolumns@{#1}%
3744     \spaceskip=\z@skip \xspaceskip=\z@skip
3745     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
3746
3747 %

```

`\mpnormalvfootnoteX` The minipage version.

```

3748 \newcommand*{\mpnormalvfootnoteX}[2]{%
3749   \get@thisfootnoteX{#1}%
3750   \get@fnmarkX{#1}{\thisc@footnote}%
3751   \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3752   \global\setbox\@nameuse{mpfootins#1}\vbox{%
3753     \unvbox\@nameuse{mpfootins#1}
3754     \noindent\csuse{bhooknoteX@#1}%
3755     \csuse{notefontsizeX@#1}%
3756     \hsize\columnwidth
3757     \@parboxrestore

```

```

3758 \color@begingroup
3759 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
3760
3761 %

```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

3762 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmtX}[2]{%
3763 \ifluatex%
3764 \textdir\footnote@luatexttexdir%
3765 \pardir\footnote@luatexpardir%
3766 \par%
3767 \fi%
3768 \protected@edef\@currentlabel{%
3769 \@nameuse{@thefnmark#1}%
3770 }%
3771 \ledsetnormalparstuffX{#1}%
3772 \hangindent=\csuse{hangindentX@#1}%
3773 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3774 \rule{z@}{\splittopskip}%
3775 {\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}}%
3776 \csuse{wrapcontentX@#1}{#2}%
3777 \strut\par}}
3778
3779 %

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

3780 \newcommand*\normalfootfootmarkX[1]{%
3781 \textsuperscript{\@nameuse{@thefnmark#1}}}
3782
3783 %

```

`\normalfootstartX` `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

3784 \newcommand*\normalfootstartX[1]{%
3785 \ifdimequal{Opt}{\prenotesX@}{}%
3786 {%
3787 \iftoggle{prenotesX@}{%
3788 \togglefalse{prenotesX@}%
3789 \skip\csname footins#1\endcsname=%
3790 \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3791 }%
3792 }%
3793 }%
3794 \vskip\skip\csname footins#1\endcsname%
3795 \leftskip=\z@
3796 \rightskip=\z@

```



```

3797 \ifl@dpairing\else%
3798   \hsize=\old@hsize%
3799 \fi%
3800 \setnoteswidthliketwocolumnsX@{#1}%
3801 \setnotesXpositionliketwocolumns@{#1}%
3802 \print@footnoteXrule{#1}%
3803 }%
3804
3805 %

```

\normalfootnoteruleX The rule drawn before the footnote series group.

```

3806 \let\normalfootnoteruleX=\footnoterule
3807
3808 %

```

\normalfootgroupX `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

3809 \newcommand*{\normalfootgroupX}[1]{%
3810   \csuse{bhookgroupX@#1}%
3811   \unvbox\@nameuse{footins#1}%
3812   \hsize=\old@hsize%
3813 }%
3814
3815 %

```

\mpnormalfootgroupX The minipage version.

```

3816 \newcommand*{\mpnormalfootgroupX}[1]{%
3817   \vskip\skip\@nameuse{mpfootins#1}
3818   \ifl@dpairing\ifparledgroup%
3819     \leavevmode\marks\parledgroup@{begin}%
3820     \marks\parledgroup@series{#1}%
3821     \marks\parledgroup@type{footnoteX}%
3822   \fi\fi\normalcolor
3823   \ifparledgroup%
3824     \ifl@dpairing%
3825     \else%
3826       \setnoteswidthliketwocolumnsX@{#1}%
3827       \setnotesXpositionliketwocolumns@{#1}%
3828       \print@footnoteXrule{#1}%
3829     \fi%
3830   \else%
3831     \setnoteswidthliketwocolumnsX@{#1}%
3832     \setnotesXpositionliketwocolumns@{#1}%
3833     \print@footnoteXrule{#1}%
3834   \fi%
3835   \csuse{bhookgroupX@#1}%
3836   \unvbox\@nameuse{mpfootins#1}}

```

```

3837
3838 %

\normalbfnoteX%
3840 \newcommand{\normalbfnoteX}[2]{%
3841   \get@thisfootnoteX{#1}%
3842   \ifledRcol%
3843     \ifluatex
3844       \footnotelang@lua[R]%
3845     \fi
3846     \ifundefined{xpg@main@language}%if polyglossia
3847       {}%
3848       {\footnotelang@poly[R]}%
3849     \xright@appenditem{%
3850       \noexpand\led@set@index@fornote{#1}%
3851       \noexpand\prepare@edindex@fornote{\led@nums}%
3852       \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3853       \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
3854       \noexpand\led@reinit@index@fornote%
3855       \unexpanded{\advance\@edindex@fornote@m@ne}%
3856     }%
3857       \to\inserts@listR
3858     \global\advance\insert@countR \@ne%
3859   \else%
3860     \ifluatex
3861       \footnotelang@lua%
3862     \fi
3863     \ifundefined{xpg@main@language}%if polyglossia
3864       {}%
3865       {\footnotelang@poly}%
3866     \xright@appenditem{%
3867       \noexpand\led@set@index@fornote{#1}%
3868       \noexpand\prepare@edindex@fornote{\led@nums}%
3869       \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3870       \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
3871       \noexpand\led@reinit@index@fornote%
3872       \unexpanded{\advance\@edindex@fornote@m@ne}%
3873     }%
3874       \to\inserts@list
3875     \global\advance\insert@count \@ne%
3876   \fi
3877   \ignorespaces}
3878
3879 %

```

`\get@thisfootnoteX` The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

3880 \newcommand{\get@thisfootnoteX}[1]{%
3881   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3882     \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
endcsname}%
3883   }{%
3884     \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}%
3885   }%
3886 }%
3887 %

```

\vbfnoteX This command calls the correct footnote-inserting commands.

```

3888 \newcommand{\vbfnoteX}[3]{%
3889   \get@fnmarkX{#1}{#3}%
3890   \@nameuse{regvfootnote#1}{#1}{#2}%
3891 }%
3892 %
3893 %

```

\get@fnmarkX This command gets the correct footnote number when typesetting parallel texts.

```

3894 \newcommand{\get@fnmarkX}[2]{%
3895   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3896     {%
3897       \stepcounter{footnote#1@typeset}%
3898       \setcounter{footnote#1}{\value{footnote#1@typeset}}%
3899       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3900       \immediate\write\@mainaux{%
3901         \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
@typeset\endcsname}%
3902       }%
3903     }%
3904     {%
3905       \setcounter{footnote#1}{#2}%
3906       \@namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3907     }%
3908   }
3909   %
3910   %

```

```

\vnunfootnoteX_{11} \newcommand{\vnunfootnoteX}[2]{%
3912   \ifnumberedpar@
3913   \edtext{}{\normalbfnoteX{#1}{#2}}%
3914   \else
3915     \def\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3916     \get@thisfootnoteX{#1}%
3917     \get@fnmarkX{#1}{\expandonce\thisc@footnote}%

```

```

3918 \@nameuse{regvfootnote#1}{#1}{#2}%
3919 \fi}
3920
3921 %

```

`arrangementX@normal` `\arrangementX@normal{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```

3922 \newcommand*{\arrangementX@normal}[1]{%
3923 \csgdef{series@displayX#1}{normal}
3924 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
3925 \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
3926 \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
3927 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
3928 \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
3929 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
3930 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
3931 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
3932 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3933 \count\csname footins#1\endcsname=1000
3934 \csxdef{default@footins#1}{1000}%Use to have note only for one side
3935 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3936 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3937 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3938 %

```

Additions for minipages.

```

3939 \ifnoledgroup@else%
3940 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3941 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
3942 \count\csname mpfootins#1\endcsname=1000
3943 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
3944 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3945 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3946 \fi
3947 }
3948
3949 %

```

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%50 \newcommand*{\arrangementX@twocol}[1]{%
3951 \csgdef{series@displayX#1}{twocol}
3952 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
3953 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
3954 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX

```

```

3955 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
3956 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3957 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
3958 \twocolfootsetupX{#1}
3959 \ifnoledgroup@ \else%
3960   \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3961   \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
3962   \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3963   \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
3964   \mptwocolfootsetupX{#1}
3965 \fi%
3966 }
3967
3968 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
3969 \newcommand*{\twocolfootsetupX}[1]{%
3970   \count\csname footins#1\endcsname 500
3971   \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
3972   \multiply\dimen\csname footins#1\endcsname by \tw@}
3973 \newcommand*{\mptwocolfootsetupX}[1]{%
3974   \count\csname mpfootins#1\endcsname 500
3975   \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
3976
3977 %

```

```

\twocolvfootnoteX \twocolvfootnoteX{<series>}
3978 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
3979   \csuse{beforeinsertingX@#1}%
3980   \insert\csname footins#1\endcsname\bgroup%
3981     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3982     \noindent\csuse{bhooknoteX@#1}%
3983     \csuse{notefontsizeX@#1}%
3984     \footsplitskips%
3985     \spaceskip=\z@skip \xspaceskip=\z@skip%
3986     \@nameuse{footfmt#1}{#1}{#2}\egroup}
3987
3988 %

```

```

\twocolfootfmtX \twocolfootfmtX{<series>}
3989 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
3990   \protected@edef\@currentlabel{%
3991     \@nameuse{@thefnmark#1}%
3992   }%
3993   \normal@pars%
3994   \hangindent=\csuse{hangindentX@#1}%

```

```

3995 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3996 \hsize \csuse{hsizetwocolX@#1}%
3997 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
3998 \tolerance=5000\relax%
3999 \par%
4000 \@tempdima=\parindent%
4001 \csuse{colalignX@#1}%
4002 \parindent=\@tempdima%
4003 {\hspace{\parindent}%
4004 \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4005 \csuse{wrapcontentX@#1}{#2}%
4006 \strut\par}%
4007 \allowbreak%
4008 }%
4009
4010 %

```

```

\twocolfootgroupX \twocolfootgroupX{(series)}
\mptwocolfootgroupX
4011 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4012 \splittopskip=\ht\strutbox
4013 \expandafter
4014 \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4015
4016 \newcommand*{\mptwocolfootgroupX}[1]{%
4017 \vskip\skip\@nameuse{mpfootins#1}
4018 \ifl@dpairing\ifparledgroup%
4019 \leavevmode\marks\parledgroup@{begin}%
4020 \marks\parledgroup@series{#1}%
4021 \marks\parledgroup@type{footnoteX}%
4022 \fi\fi\normalcolor
4023 \ifparledgroup%
4024 \ifl@dpairing%
4025 \else%
4026 \setnoteswidthliketwocolumnsX@{#1}%
4027 \setnotesXpositionliketwocolumns@{#1}%
4028 \print@footnoteXrule{#1}%
4029 \fi%
4030 \else%
4031 \setnoteswidthliketwocolumnsX@{#1}%
4032 \setnotesXpositionliketwocolumns@{#1}%
4033 \print@footnoteXrule{#1}%
4034 \fi%
4035 \csuse{bhookgroupX@#1}%
4036 \splittopskip=\ht\strutbox
4037 \expandafter
4038 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4039
4040 %

```

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]{%
4042   \csgdef{series@displayX#1}{threecol}
4043   \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4044   \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4045   \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4046   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4047   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4048   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4049   \threecolfootsetupX{#1}
4050   \ifnoledgroup@ \else%
4051     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4052     \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4053     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4054     \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4055     \mpthreecolfootsetupX{#1}
4056   \fi%
4057 }
4058
4059 %

\threecolfootsetupX  \threecolfootsetupX{<series>}
\mpthreecolfootsetupX
4060 \newcommand*{\threecolfootsetupX}[1]{%
4061   \count\csname footins#1\endcsname 333
4062   \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4063   \multiply\dimen\csname footins#1\endcsname by \thr@@
4064   \newcommand*{\mpthreecolfootsetupX}[1]{%
4065     \count\csname mpfootins#1\endcsname 333
4066     \multiply\dimen\csname mpfootins#1\endcsname by \thr@@
4067   }
4068   %

\threecolvfootnoteX  \threecolvfootnoteX{<series>}{<text>}
4069 \notbool{parapparatus@}{\newcommand*{\newcommand}{\threecolvfootnoteX}[2]{
%
4070   \csuse{beforeinsertingX@#1}%
4071   \insert\csname footins#1\endcsname\bgroup%
4072     \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4073     \noindent\csuse{hooknoteX@#1}%
4074     \csuse{notefontsizeX@#1}%
4075     \footsplitskips%
4076     \@nameuse{footfmt#1}{#1}{#2}\egroup}
4077

```

```

4078 %

\threecolfootfmtX \threecolfootfmtX{<series>}

4079 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4080   \protected@edef\@currentlabel{%
4081     \@nameuse{@thefnmark#1}%
4082   }%
4083   \hangindent=\csuse{hangindentX@#1}%
4084   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4085   \normal@pars%
4086   \hsize \csuse{hsizethreecolX@#1}%
4087   \nottoggle{parindentX@#1}{\parindent=z@}{}%
4088   \tolerance=5000\relax%
4089   \@tempdima=\parindent%
4090   \csuse{colalignX@#1}%
4091   \parindent=\@tempdima%
4092   {\hspace{\parindent}%
4093    \csuse{notenumberfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4094    \csuse{wrapcontentX@#1}{#2}%
4095    \strut\par}\allowbreak}
4096
4097 %

\threecolfootgroupX \threecolfootgroupX{<series>}
\mpthreecolfootgroupX
4098 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4099   \splittopskip=\ht\strutbox
4100   \expandafter
4101   \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4102
4103 \newcommand*{\mpthreecolfootgroupX}[1]{%
4104   \vskip\skip\@nameuse{mpfootins#1}
4105   \ifl@dpairing\ifparledgroup
4106     \leavevmode\marks\parledgroup@{begin}%
4107     \marks\parledgroup@series{#1}%
4108     \marks\parledgroup@type{footnoteX}%
4109     \fi\fi\normalcolor
4110     \ifparledgroup%
4111       \ifl@dpairing%
4112       \else%
4113         \setnoteswidthliketwocolumnsX@{#1}%
4114         \setnotesXpositionliketwocolumns@{#1}%
4115         \print@footnoteXrule{#1}%
4116       \fi%
4117     \else%
4118       \setnoteswidthliketwocolumnsX@{#1}%
4119       \setnotesXpositionliketwocolumns@{#1}%
4120       \print@footnoteXrule{#1}%

```



```

4121 \fi%
4122 \csuse{bhookgroupX@#1}%
4123 \splittopskip=\ht\strutbox
4124 \expandafter
4125 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4126
4127 %

```

XIII.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@threecol` `\footparagraphX{<series>}`

```

4128 \newcommand*{\arrangementX@paragraph}[1]{%
4129 \csgdef{series@displayX#1}{paragraph}%
4130 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4131 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4132 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4133 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4134 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4135 \count\csname footins#1\endcsname=1000
4136 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
4137 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4138 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4139 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4140 \para@footsetupX{#1}
4141 \ifnoledgroup@else
4142 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4143 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4144 \count\csname mpfootins#1\endcsname=1000
4145 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4146 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4147 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4148 \fi
4149 }
4150
4151 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

4152 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4153 \setnoteswidthliketwocolumnsX@{#1}%
4154 \ifcsempy{widthX@#1}%
4155 {}%
4156 {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4157 \dimen0=\baselineskip
4158 \multiply\dimen0 by 1024

```

```

4159 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
4160 %
4161 \expandafter
4162 \xdef\csname footfudgefactor#1\endcsname{%
4163   \expandafter\strip@pt\dimen0 }}
4164 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

4165 \newcommand*{\parafootstartX}[1]{%
4166   \ifdimequal{Opt}{\prenotesX@}{}%
4167   {%
4168     \iftoggle{prenotesX@}{%
4169       \togglefalse{prenotesX@}%
4170       \skip\csname footins#1\endcsname=%
4171       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4172     }%
4173   }%
4174 }%
4175 \leftskip=\z@
4176 \rightskip=\z@
4177 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4178 \vskip\skip\@nameuse{footins#1}%
4179 \setnoteswidthliketwocolumnsX@{#1}%
4180 \setnotesXpositionliketwocolumns@{#1}%
4181 \print@footnoteXrule{#1}%
4182 }
4183 %
4184 %

```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`
`\mppara@vfootnoteX`

```

4185 \newcommand*{\para@vfootnoteX}[2]{%
4186   \csuse{beforeinsertingX@#1}%
4187   \insert\csname footins#1\endcsname%
4188   \bgroup
4189     \csuse{notefontsizeX@#1}
4190     \footsplitskips
4191     \setbox0=\vbox{\hsize=\maxdimen%
4192       \let\bidir@RTL@everypar\@empty%
4193       \noindent\csuse{bhooknoteX@#1}%
4194       \@nameuse{footfmt#1}{#1}{#2}}%
4195     \setbox0=\hbox{\unvXH{0}{#1}}%
4196     \dp0=\z@
4197     \ht0=\csname footfudgefactor#1\endcsname\wd0
4198     \box0
4199     \penalty0
4200   \egroup}
4201 \newcommand*{\mppara@vfootnoteX}[2]{%

```

```

4202 \get@thisfootnoteX{#1}%
4203 \get@fnmarkX{#1}{\thisc@footnote}%
4204 \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4205 \global\setbox\@nameuse{mpfootins#1}\vbox{%
4206   \unvbox\@nameuse{mpfootins#1}
4207   \csuse{notefontsizeX@#1}
4208   \footsplitskips
4209   \setbox0=\vbox{\hsize=\maxdimen%
4210     \let\bidir@RTL@everypar\@empty%
4211     \noindent\color@begingroup%
4212     \csuse{bhooknoteX@#1}%
4213     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
4214   \setbox0=\hbox{\unvxhX{0}{#1}}%
4215   \dp0=\z@
4216   \ht0=\csname footfudgefactor#1\endcsname\wd0
4217   \box0
4218   \penalty0}}
4219
4220 %

```

`\unvxhX`₂₁ `\newcommand*{\unvxhX}[2]{%` 2th is optional for retro-compatibility

```

4222 \setbox0=\vbox{\unvbox#1%
4223   \global\setbox1=\lastbox}%
4224 \unhbox1
4225 \unskip           % remove \rightskip,
4226 \unskip           % remove \parfillskip,
4227 \unpenalty        % remove \penalty of 10000,
4228 \hskip\csuse{afternoteX@#2}%
4229 \relax}% but add the glue to go between the notes
4230
4231 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

4232 \newcommand*{\parafootfmtX}[2]{%
4233   \protected@edef\@currentlabel{%
4234     \@nameuse{@thefnmark#1}%
4235   }%
4236   \insertparafootsepX{#1}%
4237   \ledsetnormalparstuff@common%
4238   {\csuse{notenumfontX@#1}%
4239     \csuse{notenumfontX@#1}%
4240     \wrapped@footfootmarkX{#1}%
4241     \strut%
4242     \csuse{wrapcontentX@#1}{#2}%
4243     \penalty-10}}
4244
4245 %

```

```

\para@footgroupX \para@footgroupX{\series}
\mppara@footgroupX
4246 \newcommand*\para@footgroupX}[1]{%
4247   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4248   \unvbox\csname footins#1\endcsname
4249   \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4250   \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4251   \makehboxofhboxes
4252   \setbox0=\hbox{\unhbox0 \removehboxes}%
4253   \csuse{bhookgroupX@#1}
4254   \csuse{notefontsizeX@#1}
4255   \unhbox0\par}
4256
4257 \newcommand*\mppara@footgroupX}[1]{%
4258   \setnoteswidthliketwocolumnsX@{#1}%
4259   \vskip\skip\@nameuse{mpfootins#1}
4260   \ifl@dpairing\ifparledgroup
4261     \leavevmode%
4262     \leavevmode\marks\parledgroup@{begin}%
4263     \marks\parledgroup@series{#1}%
4264     \marks\parledgroup@type{footnoteX}%
4265   \fi\fi\normalcolor
4266   \ifparledgroup%
4267     \ifl@dpairing%
4268     \else%
4269       \setnoteswidthliketwocolumnsX@{#1}%
4270       \setnotesXpositionliketwocolumns@{#1}%
4271       \print@footnoteXrule{#1}%
4272     \fi%
4273   \else%
4274     \setnoteswidthliketwocolumnsX@{#1}%
4275     \setnotesXpositionliketwocolumns@{#1}%
4276     \print@footnoteXrule{#1}%
4277   \fi%
4278   \unvbox\csname mpfootins#1\endcsname
4279   \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4280   \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4281   \makehboxofhboxes
4282   \setbox0=\hbox{\unhbox0 \removehboxes}%
4283   \csuse{bhookgroupX@#1}%
4284   \csuse{notefontsizeX@#1}%
4285   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4286   \unhbox0\par}}
4287
4288 %

```

Insertion of the footnotes separator The command `\insertparafootsepX{\series}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX89 \newcommand{\insertparafootsepX}[1]{%
4290   \iflabeledRcol{%
4291     \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
4292       {\csuse{Xparafootsep@#1}}%
4293       {}%
4294     \global\csname prevpage#1@numR\endcsname=\page@numR%
4295   \else%
4296     \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
4297       {\csuse{Xparafootsep@#1}}%
4298       {}%
4299     \global\csname prevpage#1@num\endcsname=\page@num%
4300   \fi%
4301 }
4302 %

```

XIII.5 Wrapping footnote marks in hyperlink

`\wrapped@footfootmarkX` `\wrapped@footfootmarkX` prints the footnote mark of the footpage, wrapped in `hyperref` package's commands, if needed.

```

4303 \newcommand{\wrapped@footfootmarkX}[1]{%
4304   \ifdefined\hypertarget%
4305     \hyperlink%
4306       {@bodyfootmark#1@\this@footnoteX@reading}%
4307       {@nameuse{footfootmark#1}}%
4308     \Hy@raisedlink{%
4309       \hypertarget%
4310       {@footnotemark#1@\this@footnoteX@reading}%
4311       {}%
4312     }%
4313   \else%
4314     \@nameuse{footfootmark#1}%
4315   \fi%
4316 }%
4317 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package's commands, if needed.

```

4318 \newcommand{\wrapped@bodyfootmarkX}[1]{%
4319   \ifdefined\hypertarget%
4320     \hyperlink%
4321       {@footnotemark#1\expandafter\the\csname footnote#1@reading\endcsname}%
4322       {@nameuse{bodyfootmark#1}}%
4323     \Hy@raisedlink{%
4324       \hypertarget%
4325       {@bodyfootmark#1\expandafter\the\csname footnote#1@reading\endcsname}%

```

```

4326     {}%
4327   }%
4328   \else%
4329     \@nameuse{bodyfootmark#1}%
4330   \fi%
4331 }%
4332 %

```

XIV Code common to both critical and familiar footnote in normal arrangement

`\par` should always be redefined to `\endgraf` within the format macro (this is what `\normal@pars` does), to override tricky material in the main text to get the lines numbered automatically (as set up by `\autopar`, for example).

In the case of footnote arranged in a “normal” way, we also must set some setting for paragraph indent and text direction when using Lua \TeX .

That why we have defined `\ledsetnormalparstuff@common` in order to make this setting for both familiar and critical notes. This command is called by command to make specific setting to critical or familiar footnote.

```

\ledsetnormalparstuff@common33 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff434   \ifluatex%
\ledsetnormalparstuffX435   \ifdefstring{\footnote@luatextextdir}{TLT}{}%
4336     {\textdir\footnote@luatextextdir}%
4337   \pardir\footnote@luatexpardir%
4338   \fi%
4339   \csuse{\csuse{footnote@dir}}%
4340   \normal@pars%
4341   \parfillskip \z@ \@plus 1fil}%
4342
4343 \newcommand*{\Xledsetnormalparstuff}[1]{%
4344   \ledsetnormalparstuff@common%
4345   \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
4346 }%
4347
4348 \newcommand*{\ledsetnormalparstuffX}[1]{%
4349   \ledsetnormalparstuff@common%
4350   \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}%
4351 }%
4352 %

```

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.
`noteswidthliketwocolumns@` They set, if the option is enabled, the `\hsize`. They are also called at the on the setup
`oteswidthliketwocolumnsX@` for paragraphed notes.

```

4353
4354 \newdimen\old@hsize%
4355 \AtBeginDocument{\old@hsize=\hsize}%
4356
4357 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
4358   \global\let\hsize@fornote=\hsize%
4359   \global\old@hsize=\hsize%
4360   \let\old@columnwidth=\columnwidth%
4361   \iftoggle{Xnoteswidthliketwocolumns@#1}%
4362     {%
4363       \csuse{setwidthliketwocolumns@\columns@position}%
4364       \global\let\hsize@fornote=\hsize%
4365     }%
4366     {}%
4367   \let\hsize=\hsize@fornote%
4368   \let\columnwidth=\old@columnwidth%
4369 }%
4370
4371 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4372   \global\let\hsize@fornote=\hsize%
4373   \global\old@hsize=\hsize%
4374   \let\old@columnwidth=\columnwidth%
4375   \iftoggle{noteswidthliketwocolumnsX@#1}%
4376     {%
4377       \csuse{setwidthliketwocolumns@\columns@position}%
4378       \global\let\hsize@fornote=\hsize%
4379     }%
4380     {}%
4381   \let\hsize=\hsize@fornote%
4382   \let\columnwidth=\old@columnwidth%
4383 }%
4384
4385 %

```

`espositionliketwocolumns@` These two commands set the position of the critical / familiar footnotes, depending on
`sXpositionliketwocolumns@` the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They
 call commands which are defined only in `reledpar`, because this feature has no sens
 without `reledpar`.

```

4386 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
4387   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4388     \csuse{setnotespositionliketwocolumns@\columns@position}%
4389   }{}%
4390 }%
4391
4392 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%

```

```

4393 \iftoggle{noteswidthliketwocolumnsX@#1}{%
4394 \csuse{setnotespositionliketwocolumns@\columns@position}%
4395 }{}%
4396 }%
4397
4398 %

```

XVI Footnotes' order

`\fnpos` The `\fnpos` and `\mpfnpos` simply place their arguments in `\@fnpos` and `\@mpfnpos`, which will be used later in the output routine.

```

\mpfnpos
\@fnpos
\@mpfnpos
4399 \def\@fnpos{familiar-critical}
4400 \def\@mpfnpos{critical-familiar}
4401 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
4402 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
4403 %

```

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 4404 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoterule 4405 \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
4406 \nointerlineskip%
4407 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
4408 \nointerlineskip%
4409 \vskip\csuse{Xafterrule@#1}%
4410 }%
4411
4412 \newcommand{\print@footnoterule}[1]{%
4413 \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
4414 \nointerlineskip%
4415 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
4416 \nointerlineskip%
4417 \vskip\csuse{afterruleX@#1}%
4418 }%
4419
4420 %

```


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.

```

4421 \newtoggle{Xprenotes@}%
4422 \toggletrue{Xprenotes@}%
4423 \newcommand{\Xprenotes@}{0pt}%
4424 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
4425 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
4426 %

```

The same, but for familiar footnotes.

```

\XprenotesS27 \newtoggle{prenotesX@}
\Xprenotes@28 \toggletrue{prenotesX@}
4429 \newcommand{\prenotesX@}{Opt}
4430 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
4431 %

```

XVIII.0.3 Internal commands

```

firstXseries@32 \gdef\firstXseries@{}
prepare@Xprenotes33 \newcommand{\prepare@Xprenotes}[1]{%
4434 \ifdimequal{Opt}{\Xprenotes@}%
4435 {}%
4436 {%
4437 \IfStrEq{\firstXseries@}{}{%
4438 \global\skip\csuse{#1footins}=\Xprenotes@%
4439 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4440 \gdef\firstXseries@{#1}%
4441 }%
4442 {%
4443 \ifseriesbefore{#1}{\firstXseries@}%
4444 {%
4445 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@firstXseries@}%
4446 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4447 \gdef\firstXseries@{#1}%
4448 }%
4449 {%
4450 }%
4451 }%
4452 }
4453 %

```

The same thing is required for familiar notes and \prenotesX.

```

firstseriesX@54 \gdef\firstseriesX@{}
prepare@prenotesX55 \newcommand{\prepare@prenotesX}[1]{%
4456 \ifdimequal{Opt}{\prenotesX@}%
4457 {}%
4458 {%
4459 \IfStrEq{\firstseriesX@}{}{%
4460 \global\skip\csuse{footins#1}=\prenotesX@%
4461 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4462 \gdef\firstseriesX@{#1}%
4463 }%
4464 {%
4465 \ifseriesbefore{#1}{\firstseriesX@}%

```

```

4466     {%
4467     \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
4468     \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4469     \gdef\firstXseries@{#1}%
4470     }%
4471     {}%
4472   }%
4473 }%
4474 }
4475 %

```

XIX Endnotes

First, check the noend option.

```

4476 \ifbool{noend@}{}{%Used instead of \ifnoend@ to prevent expansion problem
4477 %

```

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.

```

4478 \newcommand{\l@dend@open}[1]{%
4479   \global\booltrue{l@dend@#1}%
4480   \expandafter\immediate%
4481   \expandafter\openout%
4482   \csname l@d@#1end\endcsname%
4483   =\l@auxdir\jobname.#1end\relax%
4484 }%
4485 \newcommand{\l@dend@close}[1]{%
4486   \global\boolfalse{l@dend@#1}%
4487   \expandafter\immediate%
4488   \expandafter\closeout\csname l@d@#1end\endcsname%
4489 }%
4490 %
4491 %

```

`\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.

```

4492 \newcommand{\l@dend@stuff}{%
4493   \def\do##1{%

```

```

4494 \ifbool{l@dend@##1}{}%
4495 {\l@dend@open{##1}}%
4496 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
4497 }%
4498 \dolistloop{@series}%
4499 }%
4500
4501 %

```

`\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.

```

4502 \global\newbool{parapparatus@}{\long}\def\endprint#1#2#3#4#5#6#7{%
4503 \csuse{Xendbhooknote@#4}%
4504 \csuse{Xendnotefontsize@#4}%
4505 \hangindent=\csuse{Xendhangindent@#4}%
4506 \ifXendinsertsep@%
4507 \hskip\csuse{Xendafternote@#4}\relax%
4508 \csuse{Xendsep@#4}%
4509 \else%
4510 \iftoggle{Xendparagraph@#4}%
4511 {\global\Xendinsertsep@true}%
4512 {}%
4513 \fi%
4514 \xdef\@currentseries{#4}%
4515 \def\do##1{%
4516 \setkeys[mac]{truefootnoteoption}{##1}%
4517 }%
4518 \notblank{#5}{\docsvlist{#5}}{}%
4519 \IfStrEq{#6}{R}{\ledRcol@true}{}%
4520 \def\@this@crossref@start{#7:start}%
4521 \def\@this@crossref@end{#7:end}%

```

```

4522 \printlineendnote{#1}{#4}%
4523 \IfStrEq{#6}{R}{\ledRcol@false}{}%
4524 \undef\@this@crossref@start%
4525 \undef\@this@crossref@end%
4526 \nottoggle{Xendlemmadisablefontselection@#4}%
4527   {\select@lemmafont#1|}%
4528   {}%
4529 \bgroup%
4530   \csuse{Xendlemmafont@#4}%
4531   \csuse{Xendwraplemma@#4}{#2}%
4532 \egroup%
4533 \ifboolexpr{%
4534   togl {nosep@}%
4535   or test{\ifcsemtty{Xendlemmaseparator@#4}}%
4536   }%
4537   {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
4538   {\nobreak%
4539     \hskip\csuse{Xendbeforelemmaseparator@#4}%
4540     \csuse{Xendlemmaseparator@#4}%
4541     \hskip\csuse{Xendafterlemmaseparator@#4}%
4542     \relax%
4543   }%
4544   \csuse{Xendwrapcontent@#4}{#3}%
4545   \nottoggle{Xendparagraph@#4}{\par}{}%
4546   \def\do##1{%
4547     \setkeys[mac]{falsefootnoteoption}{##1}%
4548   }%
4549   \notblank{#5}{\docsvlist{#5}}{}%
4550 }}%
4551
4552 \let\l@d@section=\@gobble
4553
4554 %

```

\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.

```

4555 \newcommand{\printlineendnote}[2]{%
4556   \l@dp@rsefootspec#1|%
4557   \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
4558     \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
4559       \l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
4560       \l@dparsedendsub}%
4561     \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
4562       \l@dparsedstartsub}%
4563     \ifboolexpr{%

```

```

4564     tog1 {nonum@}%
4565     or tog1 {Xendnonumber@#2}%
4566     }%
4567     {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4568     {%
4569     \iftoggle{Xendnumberonlyfirstinline@#2}%
4570     {\ifcdef{prevendline#2}%
4571     {\ifcsequal{prevendline#2}{lineinfo@}%
4572     {%
4573     \csuse{Xendbhookinplaceofnumber@#2}%
4574     \ifcempty{Xendsymmlinenumber@#2}%
4575     {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4576     {\printsymlineendnotearea{#2}}}%
4577     \csuse{Xendahookinplaceofnumber@#2}%
4578     }%
4579     {\printlineendnotearea{#1}{#2}}}%
4580     {\printlineendnotearea{#1}{#2}}}%
4581     }%
4582     {\printlineendnotearea{#1}{#2}}%We keep every time line
4583     \csxdef{prevendline#2}{\lineinfo@}%
4584     }%
4585 }%
4586 %

```

```

\printsymlineendnotearea \newcommand{\printsymlineendnotearea}[1]{%
4588     \hspace{\csuse{Xendbeforesymmlinenumber@#1}}%
4589     \csuse{Xendnotenumfont@#1}%
4590     \ifdimequal{\csuse{Xendboxsymmlinenumber@#1}}{\z@}%
4591     {\csuse{Xendsymmlinenumber@#1}}%
4592     {\hbox to \csuse{Xendboxsymmlinenumber@#1}%
4593     {\csuse{Xendsymmlinenumber@#1}\hfill}%
4594     }%
4595     \hspace{\csuse{Xendaftersymmlinenumber@#1}}%
4596 }%
4597 %

```

\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.)

```

4598 \newcommand{\printlineendnotearea}[2]{%
4599     \csuse{Xendbhooklinenumber@#2}%
4600     \hspace{\csuse{Xendbeforenumber@#2}}%
4601     \bgroup%
4602     \csuse{Xendnotenumfont@#2}%
4603     \ifdimequal{\csuse{Xendboxlinenumber@#2}}{0pt}%
4604     {\printendlines#1||\ifledRcol@{\Rlineflag\fi}%
4605     {\leavevmode%

```

```

4606 \hbox to \csuse{Xendboxlinenum@#2}%
4607 {%
4608 \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4609 \printendlines#1||\ifledRcol@\@Rlineflag\fi%
4610 \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4611 }%
4612 \egroup%
4613 \hspace{\csuse{Xendafternumber@#2}}%
4614 \csuse{Xendahooklinenumber@#2}%
4615 }%
4616 %

```

XIX.2 User level commands

XIX.2.1 Inserting contents to endnotes

The `\Xendnotes` commands are defined above, when defining apparatus commands by series. Here, we define only `\toendnotes` command not specific to a series, in order to insert arbitrary code. The regular version writes an unexpanded argument, while the regular version writes a once-expanded argument.

```

\toendnotes*17 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes*18 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4619 \def\do##1{%
4620 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4621 {\unexpanded{#2}\@percentchar}%
4622 }%
4623 \ifstrempy{#1}%
4624 {\dolistloop{\@series}}%
4625 {\docsvlist{#1}}%
4626 }\led@err@toendnotes@outsidenumbering}%
4627 }%
4628 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
4629 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4630 \def\do##1{%
4631 \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4632 {#2\@percentchar}%
4633 }%
4634 \ifstrempy{#1}%
4635 {\dolistloop{\@series}}%
4636 {\docsvlist{#1}}%
4637 }\led@err@toendnotes@outsidenumbering}%
4638 }%
4639 %

```

XIX.2.2 Printing endnotes

`\doendnotes` `\doendnotes` is the command you use to print one series of endnotes; it takes one argument: the series letter of the note series you want to print. `\Xendinsertsep@`

set to true at the first note of the series, and to false at the last one.

```

4640 \newif\ifXendinsertsep%
4641 \newcommand*{\doendnotes}[1]{%
4642   \l@dend@close{#1}%
4643   \begingroup
4644     \csxdef{prevpagenum@#1}{}%
4645     \csxdef{prevpagerange@#1}{}%
4646     \makeatletter
4647     \expandafter\let\csname #1end\endcsname=\endprint
4648     \input\l@auxdir\jobname.#1end%
4649     \global\Xendinsertsep@false%
4650   \endgroup}
4651 %

```

\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.

```

4652 \newcommand*{\doendnotesbysection}[1]{%
4653   \l@dend@close{#1}%
4654   \csxdef{prevpagenum@#1}{}%
4655   \csxdef{prevpagerange@#1}{}%
4656   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
4657   \begingroup%
4658     \makeatletter%
4659     \def\l@d@section##1{%
4660       \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
4661       {\cslet{#1end}{\endprint}}%
4662       {\cslet{#1end}{\@gobbleseven}}%
4663     }%
4664     \input\l@auxdir\jobname.#1end%
4665     \global\Xendinsertsep@false%
4666   \endgroup%
4667 }%
4668 %

```

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.

```

4669 }%
4670 %

```

\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.

```

4671 \newcommand*{\setprintendlines}[6]{%
4672   \l@dpnumfalse \l@ddashfalse
4673   \ifnum#4=#1 \else
4674     \l@dpnumtrue
4675     \l@ddashtrue
4676   \fi
4677   %

```

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.

```

4678   \ifl@dpnum \l@d@elintrue \else \l@d@elinfalse \fi
4679   \ifnum#2=#5 \else
4680     \l@d@elintrue
4681     \l@ddashtrue
4682   \fi
4683   %

```

We print the starting sub-line if it is nonzero.

```

4684   \l@d@ssubfalse
4685   \ifnum#3=0 \else
4686     \l@d@ssubtrue
4687   \fi
4688   %

```

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.

```

4689   \l@d@eslfalse
4690   \ifnum#6=0 \else
4691     \ifnum#6=#3
4692       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4693     \else
4694       \l@d@esltrue
4695       \l@ddashtrue
4696     \fi
4697   \fi%
4698   %

```

```

4699   \ifl@d@dash%
4700   \ifboolexpr{togl{fulllines@} or test{\ifcempty{Xendtwolines@}\
4701     \currentseries}}}%
4702   {%
4703     \setistwofollowinglines{#1}{#2}{#4}{#5}%

```

```

4704 \ifboolexpr{%
4705   (%
4706     togl {Xendtwolinesbutnotmore@\@currentseries}%
4707     and not%
4708       (%
4709         bool {istwofollowinglines@}%
4710       )%
4711   )%
4712   or%
4713   (%
4714     (not test{\ifnumequal{#1}{#4}})%
4715     and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
4716   )%
4717   }%
4718   {}%
4719   {%
4720     \l@d@dashfalse%
4721     \l@d@Xtwolinestrue%
4722     \l@d@elinfalse%
4723     \l@d@eslfalse%
4724     \ifcempty{Xendmorethantwolines@\@currentseries}%
4725       {%
4726         {\ifistwofollowinglines@\else%
4727           \l@d@Xmorethantwolinestrue%
4728         }%
4729       }%
4730     }%
4731   }%
4732 \fi%
4733 %

End of \setprintendlines.

4734 }%
4735 %

```

`\printendlines` Now we are ready to print it all.

```

4736 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4737   \begingroup
4738   \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
4739 %

```

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.

```

4740 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
4741   {\bgroup}%

```

```

4742 {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}%
4743 %

```

Then, print the starting page number-

```

4744 \ifboolexpr{%
4745   (%
4746     test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
4747     and not%
4748     (togl{Xendpagenumberonlyfirsttifsingl@\@currentseries} and bool{
l@d@pnum}})%
4749   )%
4750   or%
4751   (%
4752     test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
4753     )%
4754   }%
4755   {%
4756     \ifcempty{Xendsympagenum@\@currentseries}%
4757     {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
4758     {\csuse{Xendsympagenum@\@currentseries}}%
4759   }%
4760   {%
4761     \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
4762   }%
4763 %

```

Then, determine what must be printed before the start line.

```

4764 \ifl@d@dash%
4765 \ifl@d@pnum%
4766   \csuse{Xendlineprefixsingle@\@currentseries}%
4767 \else%
4768   \ifcempty{Xendlineprefixmore@\@currentseries}%
4769   {\csuse{Xendlineprefixsingle@\@currentseries}}%
4770   {\csuse{Xendlineprefixmore@\@currentseries}}%
4771 \fi%
4772 \else%
4773   \csuse{Xendlineprefixsingle@\@currentseries}%
4774 \fi%
4775 %

```

The print the starting line, followed, if needed, by the side flag and the starting sub line number.

```

4776 \wrap@edcrossref{\@this@crossref@start}{%
4777   \ifledRcol%
4778     \linenumrepR{#2}%
4779   \else%
4780     \linenumrep{#2}%
4781   \fi%

```

```

4782 }%
4783 \iftoggle{Xendlineflag@{\@currentseries}{\ifl@d@Rcol@{\@Rlineflag\fi}{}}%
4784 \ifl@d@ssub%
4785   \csuse{Xendsublinesep@{\@currentseries}}%
4786   \wrap@edcrossref{\@this@crossref@start}{}%
4787   \ifl@d@Rcol@%
4788     \sublinenumrepR{#3}%
4789   \else%
4790     \sublinenumrep{#3}%
4791   \fi%
4792 }%
4793 \fi%
4794 %

```

Close the box.

```

4795 \egroup%
4796 %

```

Open the box for the ending line number.

```

4797 \ifdimequal{\csuse{Xendboxendlinenum@{\@currentseries}}{0pt}}%
4798 {\bgroup}%
4799 {\hbox to \csuse{Xendboxendlinenum@{\@currentseries}}\bgroup}%
4800 %

```

Print the dash + the ending line number, or the line number range symbol.

```

4801 \ifl@d@Xtwolines%
4802   \ifl@d@Xmorethantwolines%
4803     \csuse{Xendmorethantwolines@{\@currentseries}}%
4804   \else%
4805     \csuse{Xendtwolines@{\@currentseries}}%
4806   \fi%
4807 \else%
4808   \ifl@d@dash%
4809     \ifdefined\linangesep@%
4810       \linangesep@%
4811     \else%
4812       \csuse{Xendlinangeseparator@{\@currentseries}}%
4813     \fi%
4814   \fi%
4815 %

```

Print the ending page number.

```

4816 \ifl@d@pnum%
4817   \ifcsstring{prevpagerange@{\@currentseries}}{#1-#4}%
4818   {%
4819     \ifcempty{Xendsympagenum@{\@currentseries}}%
4820       {\hspace{\csuse{Xendinplaceofpagenumber@{\@currentseries}}}}%
4821       {\csuse{Xendsympagenum@{\@currentseries}}}%
4822     }%

```

```

4823     {%
4824     \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
4825     }%
4826     \fi%
4827 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag and the subtitle number.

```

4828     \ifl@d@elin%
4829     \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
4830     \wrap@edcrossref{\@this@crossref@end}{%
4831     \ifledRcol%
4832     \linenumrepR{#5}%
4833     \else%
4834     \linenumrep{#5}%
4835     \fi%
4836     }%
4837     \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
4838     \fi%
4839     \ifl@d@esl%
4840     \ifl@d@elin%
4841     \csuse{Xendsublinesep@\@currentseries}%
4842     \fi%
4843     \wrap@edcrossref{\@this@crossref@end}{%
4844     \ifledRcol%
4845     \sublinenumrepR{#6}%
4846     \else%
4847     \sublinenumrep{#6}%
4848     \fi%
4849     }%
4850     \fi%
4851     \fi%
4852 %

```

Close the ending line box.

```

4853     \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
4854     {}%
4855     {\hfill}%Prevent underfull hbox
4856     \egroup%
4857 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

4858     \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
4859     {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%
4860     {\csxdef{prevpagerange@\@currentseries}{#1-#4}}%
4861     {\csxdef{prevpagenum@\@currentseries}{#4}}%
4862     }%
4863     {}%
4864 %

```

Now, the end of `\printendlines` macro.

```
4865 \endgroup%
4866 }%
4867
4868 %
```

`\printnpnum` A macro to print a page number in an endnote. Should not be override anymore

```
4869 \newcommand*\printnpnum[1]{\csuse{Xendbeforepagenumber@\@currentseries
}#1\csuse{Xendafterpagenumber@\@currentseries}}
4870
4871 %
```

XX Generate series of notes

In this section, X means the name of the series (A, B etc.)

`\series` `\series` creates one more new series. It is a public command, which just loops on the private command `\newseries@`.

```
4872 \newcommand{\newseries}[1]{%
4873   \def\do##1{\newseries@{##1}}%
4874   \docsvlist{#1}
4875 }
4876 %
```

`\@series` The `\series@` macro is an etoolbox list, which contains the name of all series.

```
4877 \newcommand{\@series}{}
4878 %
```

The command `\newseries@series` creates a new series of the footnote.

```
\newseries@79 \newcommand{\newseries@}[1]{
4880 %
```

XX.1 Test if series is still existing

```
4881 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
4882 {%
4883 %
```

XX.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```

4884 \ifdefined\newseries@par%
4885 \newseries@par{#1}%
4886 \fi%
4887 %

```

XX.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```

4888 \unless\ifnocritical@
4889 %

```

XX.3.1 Options

```

4890 \newtoggle{Xlineflag@#1}
4891 \newtoggle{Xparindent@#1}
4892 \newtoggle{Xlemmadisablefontselection@#1}
4893 \csgdef{Xwrapcontent@#1}{}%
4894 \csgdef{Xbeforeinserting@#1}{}%
4895 \csgdef{Xhangindent@#1}{Opt}%
4896 \csgdef{Xragged@#1}{}%
4897 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
4898 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
4899 \csgdef{Xcolalign@#1}{\raggedright}%
4900 \csgdef{Xnotenumfont@#1}{\normalfont}%
4901 \csgdef{Xnotefontsize@#1}{\footnotesize}%
4902 \csgdef{Xbhooknote@#1}{}%
4903 \csgdef{Xbhookgroup@#1}{}%
4904
4905 \csgdef{Xboxlinenum@#1}{Opt}%
4906 \csgdef{Xboxlinenumalign@#1}{L}%
4907
4908 \csgdef{Xboxstartlinenum@#1}{Opt}%
4909 \csgdef{Xboxendlinenum@#1}{Opt}%
4910
4911 \csgdef{Xboxsymlinenum@#1}{Opt}%
4912 \newtoggle{Xnumberonlyfirstinline@#1}%
4913 \newtoggle{Xgroupbyline@#1}%
4914 \newtoggle{Xgroupbylineseparetwolines@#1}%
4915 \newtoggle{Xnumberonlyfirstintwolines@#1}%
4916 \csgdef{Xtwolines@#1}{}%
4917 \csgdef{Xmorethantwolines@#1}{}%
4918 \csgdef{Xsublinesep@#1}{\fullstop}%
4919 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
4920 \newtoggle{Xtwolinesbutnotmore@#1}%
4921 \newtoggle{Xtwolinesonlyinsamepage@#1}%

```

```

4922 \newtoggle{Xonlypstart@#1}%
4923 \newtoggle{Xpstarteverytime@#1}%
4924 \newtoggle{Xpstart@#1}%
4925 \newtoggle{Xstanza@#1}%
4926 \csgdef{Xstanzaseparator@#1}{}%
4927 \csgdef{Xsymlinenum@#1}{}%
4928 \newtoggle{Xnonnumber@#1}%
4929 \csgdef{Xbeforenumber@#1}{0pt}%
4930 \csgdef{Xtxtbeforenumber@#1}{}%
4931 \csgdef{Xafternumber@#1}{0.5em}%
4932 \newtoggle{Xnonbreakableafternumber@#1}%
4933 \csgdef{Xbeforenymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
4934 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
4935 \csgdef{Xinplaceofnumber@#1}{1em}%
4936 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
4937 \csgdef{Xbeforelemmaseparator@#1}{0em}%
4938 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
4939 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
4940 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
4941 \csgdef{Xafterrule@#1}{0pt}%
4942
4943 \csgdef{Xtxtbeforenotes@#1}{%
4944 \newtoggle{Xtxtbeforenotes@#1@typeset}%Not directly used by user,
but internal
4945
4946 \csgdef{Xmaxhnotes@#1}{0.8\vsizer%
4947 \newtoggle{Xnoteswidthliketwocolumns@#1}%
4948 \csgdef{Xparafootsep@#1}{}%
4949 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%
4950 \csgdef{Xlinrangeseparator@#1}{\endashchar}%
4951
4952 \csgdef{Xlemmafont@#1}{}%
4953 \csgdef{Xwraplemma@#1}{%
4954 \csgdef{Xwidth@#1}{\hsizer%
4955 %

```

XX.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

4956 \expandafter\newinsert\csname #1footins\endcsname%
4957 \unless\ifnoledgroup@%
4958 \expandafter\newinsert\csname mp#1footins\endcsname%
4959 \fi%
4960 %

```

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.


```

4961 \global\newcommand{\parapparatus@}{\expandafter\newcommand\expandafter
*}{\expandafter\newcommand\csname #1footnote\endcsname[2] [] {%
4962 \if@edtext@secondarg%
4963 \ifledRcol%
4964 \ifcsstring{Xonlyside@#1}{L}{\
led@error@note@called@onrightside{#1footnote}}}%
4965 \else%
4966 \ifcsstring{Xonlyside@#1}{R}{\
led@error@note@called@onleftside{#1footnote}}}%
4967 \fi%
4968 \begingroup%
4969 \newcommand{\content}{##2}%
4970 \ifnumberedpar%
4971 \ifledRcol%
4972 \ifluatex%
4973 \footnotelang@lua[R]%
4974 \fi%
4975 \@ifundefined{xpg@main@language}%if polyglossia
4976 {}%
4977 {\footnotelang@poly[R]}%
4978 \footnoteoptions@{R}{##1}{true}%
4979 \xright@appenditem{%
4980 \ifbool{indtl@innote}%
4981 {\unexpanded{\let\index\nindex}}%
4982 {}%
4983 \ifbool{indtl@notenumber}%
4984 {\unexpanded{\let\index\nindex}}%There is no note
...number so
4985 {}%
4986 \noexpand\Xnote@true%
4987 \noexpand\prepare@Xprenotes{#1}%
4988 \noexpand\prepare@edindex@fornote{\l@d@nums}%
4989 \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.
4990 \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
4991 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}}%
4992 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
4993 \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
4994 \noexpand\csuse{v#1footnote}{#1}%
4995 {\l@d@nums}\expandonce\@tag{\expandonce\content}}
%
4996 \noexpand\Xnote@false%
4997 \unexpanded{\advance\@edindex@fornote@m@ne}%
4998 \unexpanded{\undef\@this@crossref@start}%
4999 \unexpanded{\undef\@this@crossref@end}%

```

```

5000         \ifbool{indtl@innote}%
5001             {\unexpanded{\let\index\orig@@index}}%
5002             {}%
5003         \ifbool{indtl@notenumber}%
5004             {\unexpanded{\let\index\orig@@index}}%
5005             {}%
5006         }\to\inserts@listR
5007         \footnoteoptions@{R}{#1}{false}%
5008         \global\advance\insert@countR \@ne%
5009     \else%
5010         \ifluatex%
5011             \footnotelang@lua%
5012         \fi%
5013         \ifundefined{xpg@main@language}%if polyglossia
5014             {}%
5015             {\footnotelang@poly}%
5016         \footnoteoptions@{L}{#1}{true}%
5017         \xright@appenditem{%
5018             \ifbool{indtl@innote}%
5019                 {\unexpanded{\let\index\nindex}}%
5020                 {}%
5021             \ifbool{indtl@notenumber}%
5022                 {\unexpanded{\let\index\nindex}}%There is no note
5023             ...number so
5024             {}%
5025             \noexpand\Xnote@true%
5026             \noexpand\prepare@Xprenotes{#1}%
5027             \noexpand\prepare@edindex@fornote{\l@d@nums}%
5028             \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.
5029             \ifl@dpairing%
5030                 \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
5031             \fi%
5032             \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5033             \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5034             \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
5035             \noexpand\csuse{v#1footnote}%
5036             {#1}%
5037             {\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
5038             \unexpanded{\undef\@this@crossref@start}%
5039             \unexpanded{\undef\@this@crossref@end}%
5040             \noexpand\Xnote@false%
5041             \unexpanded{\advance\@edindex@fornote\@m@ne}%
5042             \ifbool{indtl@innote}%

```

Create counter used to determine on which page the previous note was called.

We need to be able to modify `reledmac`'s footnote macros and restore their

XX.3.4 Set standard display

End of for critical footnotes.

XX.4 For familiar footnotes

```
5070 \unless\ifnofamiliar@
5071 %
```

XX.4.1 Options

```

5072 \newtoggle{parindentX@#1}
5073 \csgdef{wrapcontentX@#1}{}%
5074 \csgdef{hangindentX@#1}{Opt}%
5075 \csgdef{beforeinsertingX@#1}{}%
5076 \csgdef{raggedX@#1}{}%
5077 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
5078 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
5079 \csgdef{colalignX@#1}{\raggedright}%
5080 \csgdef{notenumfontX@#1}{\normalfont}%
5081 \csgdef{notefontsizeX@#1}{\footnotesize}%
5082 \csgdef{bhooknoteX@#1}{}%
5083 \csgdef{bhookgroupX@#1}{}%
5084 \csgdef{afterruleX@#1}{Opt}
5085 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
5086 \csgdef{maxhnotesX@#1}{0.8\vsize}%
5087 \newtoggle{noteswidthliketwocolumnsX@#1}%
5088 \csgdef{parafootsepX@#1}{}%
5089 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
5090 \csgdef{widthX@#1}{\hsize}%
5091 % End of for familiar footnotes.
5092 % \subsubsection{Create inserts, needed to add notes in foot}
5093 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
5094 % \begin{macrocode}
5095 \expandafter\newinsert\csname footins#1\endcsname%
5096 \unless\ifnoledgroup%
5097 \expandafter\newinsert\csname mpfootins#1\endcsname%
5098 \fi%
5099 %

```

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.

```

5100 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
5101 \begingroup%
5102 \prepare@prenotesX{#1}%
5103 \newcommand{\content}{##1}%
5104 %
5105 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read `reledpar`'s handbook about that (V.1.2 p. 50).

```

5106 \global\expandafter\advance\csname footnote#1@reading\endcsname by \@ne%
5107 \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or bool{!@dprintingcolumns}}{%
5108 \ifcsdef{footnote#1@reading\the\csname footnote#1@reading\endcsname=typeset}%

```

```

5109         {\setcounter{footnote#1}{\csuse{footnote#1reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
5110         {\setcounter{footnote#1}{\the\csname footnote#1@reading
\endcsname}}}%
5111     }{%
5112         \stepcounter{footnote#1}%
5113     }%
5114 %

```

We also have to check consistency with `\onlysideX` setting.

```

5115         \ifledRcol%
5116         \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}{%
5117         \else%
5118         \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}{%
5119         \fi%
5120 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

5121         \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5122         \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
5123         {\csuse{@footnotemark#1}}%
5124         {}%
5125     \ifluatex%
5126         \xdef\footnote@luatextextdir{\the\textdir}%
5127         \xdef\footnote@luatexpardir{\the\pardir}%
5128     \fi%
5129     \if@ledgroup%
5130         \led@set@index@fornote{#1}%
5131     \fi%
5132     \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
5133     \ifbool{indtl@innote}%
5134         {\let\index\orig@@index}%
5135         {}%
5136     \ifbool{indtl@notenumber}%
5137         {\let\index\orig@@index}%
5138         {}%
5139     \endgroup%
5140 }
5141 %

```

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).

```

5142     \newcounter{footnote#1}
5143     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}

```

```

5144 \expandafter\newcount\csname footnote#1@reading\endcsname%
5145 %

```

Create counter used to determine on which page the previous note was called.

```

5146 \expandafter\newcount\csname prevpage#1@num\endcsname%
5147 \expandafter\newcount\csname prevpage#1@numR\endcsname%
5148 %

```

Add `\let\footnoteX\@gobble` to `\no@expands`.

```

5149 \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
5150 %

```

Do not forget to initialize the series

```

5151 \arrangementX@normal{#1}%
5152 \fi
5153 %

```

XX.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

5154 \unless\ifnoend@
5155 %

```

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`

```

5156 \expandafter\newwrite\csname l@d@#1end\endcsname%
5157 \expandafter\newif\csname ifl@dend@#1\endcsname%
5158 %

```

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.

```

5159 \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,
5160 usedefault]{%
5161 \bgroup%
5162 \newlinechar='40%

```

```

5163     \global\@noneed@Footnotetrue%
5164     \newcommand{\content}{##2}%
5165     \stepcounter{labidx}%
5166     \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%
5167     \unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\
expandafter{\sw@inthisedtext}}\@percentchar\space%Explicit space, to add a
linebreak in the output file
5168     \expandafter\string\csname #1end\endcsname%
5169     {\ifnumberedpar@l@d@nums\fi}%
5170     {\ifnumberedpar@\expandonce\@tag\fi}%
5171     {\expandonce\content}%
5172     {#1}%
5173     {\unexpanded{##1}}%
5174     {\ifledRcol R\else L\fi}%
5175     {\theedtext}%
5176     \@percentchar%
5177     }%
5178     \egroup%
5179     \ignorespaces%
5180 }%
5181 %

```

XX.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

5182     \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
5183     \toendnotes[#1]{##1}%
5184     }%
5185
5186     \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
endcsname*[1]{%
5187     \toendnotes*[#1]{##1}%
5188     }%
5189
5190
5191 %

```

XX.5.4 Internal commands

`\Xendnote` commands called `\Xend` commands on to the endnote file; these are analogous to the various `footfmt` commands above, and they take the same arguments. When we process this file, we want to pick out the notes of one series and ignore all the rest. To do that, we equate the end command for the series we want to `\endprint`, and leave the rest equated to `\@gobbleseven`, which just skips over its seven arguments.

```

5192

```

```

5193 \global\cslet{#1end}{\@gobbleseven}
5194 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

5195 \global\expandafter\newcount\csname #1end@bysection\endcsname%
5196 %

```

XX.5.5 The options

```

5197 \csgdef{Xendwraplemma@#1}{%
5198 \csgdef{Xendwrapcontent@#1}{}%
5199 \csgdef{Xendtwolines@#1}{}%
5200 \csgdef{Xendmoreethantwolines@#1}{}%
5201 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
5202 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
5203 \newtoggle{Xendlemmadisablefontselection@#1}{}%
5204 \csgdef{Xendnotenumfont@#1}{\normalfont}%
5205 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
5206 \csgdef{Xendbhooknote@#1}{}%
5207
5208 \csgdef{Xendsublinesep@#1}{\fullstop}%
5209
5210 \csgdef{Xendbeforenumber@#1}{0pt}
5211 \csgdef{Xendafternumber@#1}{0.5em}
5212
5213 \csgdef{Xendboxlinenum@#1}{0pt}%
5214 \csgdef{Xendboxlinenumalign@#1}{L}%
5215
5216 \csgdef{Xendboxstartlinenum@#1}{0pt}%
5217 \csgdef{Xendboxendlinenum@#1}{0pt}%
5218
5219 \csgdef{Xendlemmaseparator@#1}{}%
5220 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
5221 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
5222 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
5223
5224 \newtoggle{Xendparagraph@#1}%
5225 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
5226 \csgdef{Xendsep@#1}{}%
5227
5228 \csgdef{Xendinplaceofnumber@#1}{0pt}%
5229 \newtoggle{Xendnonumber@#1}%
5230
5231 \csgdef{Xendhangindent@#1}{0pt}%
5232 \newtoggle{Xendnumberonlyfirstinline@#1}%
5233 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
5234
5235 \csgdef{Xendbeforesynlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
5236 \csgdef{Xendaftersynlinenum@#1}{\csuse{Xendafternumber@#1}}%

```



```

5237 \csgdef{Xendsymmlinenumber@#1}{}%
5238 \csgdef{Xendboxsymmlinenumber@#1}{Opt}%
5239
5240 \csgdef{Xendbhooklinenumber@#1}{}%
5241 \csgdef{Xendehooklinenumber@#1}{}%
5242 \csgdef{Xendbhookinplaceofnumber@#1}{}%
5243 \csgdef{Xendehookinplaceofnumber@#1}{}%
5244
5245 \csgdef{Xendlinerangeseparator@#1}{\endashchar}%
5246
5247 \csgdef{Xendbeforepagenumber@#1}{p.}%
5248 \csgdef{Xendafterpagenumber@#1}{ }%
5249 \csgdef{Xendlineprefixsingle@#1}{}%
5250 \csgdef{Xendlineprefixmore@#1}{}%
5251
5252 \newtoggle{Xendlineflag@#1}
5253
5254 \csgdef{Xendlemmafont@#1}{}%
5255
5256 \newtoggle{Xendpagenumberonlyfirst@#1}%
5257 \newtoggle{Xendpagenumberonlyfirsttifsingl@#1}%
5258 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
5259 \csgdef{Xendsympagenum@#1}{}%
5260 \csgdef{Xendinplaceofpagenumber@#1}{Opt}%
5261
5262 %

```

End of endnotes declaration

```

5263 \fi%
5264 %

```

Dump series in \@series

```

5265 \listxadd{\@series}{#1}
5266 }
5267 }% End of \newseries
5268 %

```

XX.6 Init standards series (A,B,C,D,E)

```

5269 \expandafter\newseries\expandafter{\default@series}
5270 %

```

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.

```

5271 \newcommand{\seriesatbegin}[1]{%
5272   \StrDel{\@series}{#1}[\@series]%
5273   \edef\@new{%
5274     \listadd{\@new}{#1}%
5275     \listadd{\@new}{\@series}%
5276   \xdef\@series{\@new}%
5277 }
5278 %

```

\seriesatend And `\seriesatend` moves the series to the end of the list.

```

5279 \newcommand{\seriesatend}[1]{%
5280   \StrDel{\@series}{#1}[\@series]%
5281   \edef\@new{%
5282     \listadd{\@new}{\@series}%
5283     \listadd{\@new}{#1}%
5284   \xdef\@series{\@new}%
5285 }
5286 %

```

XXI.2 Test series order

\ifseriesbefore `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands `<true>` if `<seriesA>` is printed before `<seriesB>`, expands `<false>` otherwise.

```

5287 \newcommand{\ifseriesbefore}[4]{%
5288   \StrPosition{\@series}{#1}[\@first]%
5289   \StrPosition{\@series}{#2}[\@second]%
5290   \ifnumgreater{\@second}{\@first}{#3}{#4}%
5291 }
5292 %

```

XXI.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries93 \newcommand{\@getfirstseries}{%
5294   \ifdefempty{\@series}%
5295   {\xdef\@firstseries{}}%
5296   {\StrChar{\@series}{1}[\@firstseries]}%
5297 }%
5298 %

```

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`.

```

5299 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
5300   \def\do##1{%
5301     \global\settoggle{#2@##1}{#3}%
5302     \ifstrequal{#4}{critical}{
5303       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
5304     }{}
5305     \ifstrequal{#4}{familiar}{
5306       \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
5307     }{}
5308   }%
5309   \ifstrepty{#1}{%
5310     \dolistloop{\@series}%
5311     \ifstrepty{#5}{}%
5312       \docsvlist{#5}%
5313   }
5314 }%
```

```

5315         {%
5316         \docsvlist{#1}%
5317         }%
5318     }
5319     %

```

`\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.

```

5320 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
5321     \def\do##1{
5322         \csgdef{#2@##1}{#3}
5323         \ifstrequal{#4}{critical}{%
5324             \csuse{Xarrangement@\csuse{series@display##1}}{##1}%
5325         }{}
5326         \ifstrequal{#4}{familiar}{%
5327             \csuse{arrangementX@\csuse{series@displayX##1}}{##1}%
5328         }{}%
5329     }%
5330     \ifstreempty{#1}{%
5331         \dolistloop{\@series}%
5332         \ifstreempty{#5}{}{%
5333             \docsvlist{#5}
5334         }
5335     }%
5336     {%
5337         \docsvlist{#1}%
5338     }%
5339 }%
5340 %

```

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`.

```

5341 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
5342   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1]{%
5343     \setcommand@series{##1}{#1}{##2}[1][#2]%
5344   }%
5345   \ifstrempy{#2}{#1}{%
5346     \def\do##1{%
5347       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
5348       [1]{%
5349         \csuse{#1}{##1}{####1}%
5350       }%
5351     }%
5352   \docsvlist{#2}%
5353 }%
5354 %

```

`\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`.

```

5355 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
5356   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
5357     true},usedefault]{%
5358     \settoggle@series{##1}{#1}{##2}[1][#2]%
5359   }%
5360   \ifstrempy{#2}{#1}{%
5361     \def\do##1{%
5362       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
5363         \csuse{#1}{##1}%
5364       }%
5365     }%
5366   \docsvlist{#2}%
5367 }%
5368 %

```

`\newhookcommand@toggle@reload` `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement, depending of type or notes

```

5369 \newcommand{\newhooktoggle@series@reload}[2]{%
5370   \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
5371     true},usedefault]{%
5372     \settoggle@series{##1}{#1}{##2}[#2]%
5373   }%
5374 %

```

`\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.

```

5375 \newcommand{\newhookcommand@series@reload}[2]{%
5376   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5377     \setcommand@series{##1}{#1}{##2}[#2]%
5378   }%
5379 }
5380 %

```

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.

```

5381 \unless\ifnocritical@
5382   \newhookcommand@series{Xwrapcontent}%
5383   \newhookcommand@series{Xbeforeinserting}%
5384   \newhookcommand@series{Xlemmafont}%
5385   \newhookcommand@series{Xwraplemma}%
5386   \newhooktoggle@series{Xparindent}
5387   \newhookcommand@series{Xhangindent}
5388   \newhookcommand@series{Xragged}
5389   \newhookcommand@series{Xhsizetwocol}
5390   \newhookcommand@series{Xhsizethreecol}
5391   \newhookcommand@series{Xcolalign}%
5392   \newhookcommand@series{Xnotenumfont}
5393   \newhookcommand@series{Xbhooknote}
5394   \newhookcommand@series@reload{Xbhookgroup}{critical}
5395   \newhookcommand@series{Xboxsymlinenum}%
5396   \newhookcommand@series{Xsymlinenum}
5397   \newhookcommand@series{Xbeforenumber}
5398   \newhookcommand@series{Xtxtbeforenumber}
5399   \newhookcommand@series{Xafternumber}
5400   \newhookcommand@series{Xbeforesymlinenum}
5401   \newhookcommand@series{Xaftersymlinenum}
5402   \newhookcommand@series{Xinplaceofnumber}
5403   \newhookcommand@series{Xlemmaseparator}
5404   \newhookcommand@series{Xbeforelemmaseparator}
5405   \newhookcommand@series{Xafterlemmaseparator}
5406   \newhookcommand@series{Xinplaceoflemmaseparator}
5407   \newhookcommand@series{Xtxtbeforenotes}
5408   \newhookcommand@series@reload{Xafterrule}{critical}
5409   \newhooktoggle@series{Xnumberonlyfirstinline}
5410   \newhooktoggle@series{Xnumberonlyfirstintwolines}
5411   \newhooktoggle@series{Xgroupbyline}%
5412   \newhooktoggle@series{Xgroupbylineseparetwolines}%
5413   \newhooktoggle@series{Xnonumber}
5414   \newhooktoggle@series{Xpstart}

```

```

5415 \newhooktoggle@series{Xstarteverytime}%
5416
5417 \newhooktoggle@series{Xstanza}%
5418 \newhookcommand@series{Xstanzaseparator}%
5419
5420 \newhooktoggle@series{Xonlypstart}
5421 \newhooktoggle@series{Xnonbreakableafternumber}
5422 \newhooktoggle@series{Xlemmadisablefontselection}
5423 \newhookcommand@series@reload{Xmaxhnotes}{critical}
5424 \newhookcommand@series@reload{Xbeforenotes}{critical}
5425 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
5426 \newhookcommand@series@reload{Xnotefontsize}{critical}
5427
5428 \newhookcommand@series{Xboxlinenum}%
5429 \newhookcommand@series{Xboxlinenumalign}%
5430
5431 \newhookcommand@series{Xboxstartlinenum}%
5432 \newhookcommand@series{Xboxendlinenum}%
5433
5434 \newhookcommand@series{Xafternote}%
5435 \newhookcommand@series{Xparafootsep}
5436
5437 \newhookcommand@series@reload{Xwidth}{critical}%
5438
5439 \ifundef{\Xhsize}%
5440 {%
5441   \newcommandx{\Xhsize}[2][1,usedefault]{%
5442     \led@warning@Xhsize@deprecated%
5443     \Xwidth[#1]{#2}%
5444   }%
5445 }%
5446 {}%
5447 \fi
5448 \newhooktoggle@series{Xlineflag}[appref,SEref]
5449 \newhookcommand@series{Xtwolines}[appref,SEref]
5450 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
5451 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
5452 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
5453 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
5454 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
5455 \newhookcommand@series{Xlinerrangeseparator}[appref,SEref]
5456 %

```

XXI.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

5457 \unless\ifnofamiliar@
5458   \newhookcommand@series{wrapcontentX}%

```

```

5459 \newhookcommand@series{beforeinsertingX}%
5460 \newhooktoggle@series{parindentX}
5461 \newhookcommand@series{hangindentX}
5462 \newhookcommand@series{raggedX}
5463 \newhookcommand@series{hsizetwocolX}
5464 \newhookcommand@series{hsizethreecolX}
5465 \newhookcommand@series{colalignX}%
5466 \newhookcommand@series{notenumfontX}
5467 \newhookcommand@series{bhooknoteX}
5468 \newhookcommand@series@reload{bhookgroupX}{familiar}
5469 \newhookcommand@series@reload{beforenotesX}{familiar}
5470 \newhookcommand@series@reload{maxhnotesX}{familiar}
5471 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
5472 \newhookcommand@series@reload{aftererruleX}{familiar}
5473 \newhookcommand@series@reload{notefontsizeX}{familiar}
5474 \newhookcommand@series{afternoteX}
5475 \newhookcommand@series{parafootsepX}
5476 \newhookcommand@series@reload{widthX}{familiar}%
5477 \ifundef{\hsizeX}%
5478   {%
5479     \newcommandx{\hsizeX}[2][1,usedefault]{%
5480       \led@warning@hsizeX@deprecated%
5481       \widthX[#1]{#2}%
5482     }%
5483   }%
5484   {}%
5485 \fi
5486 %

```

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.

```

5487 \unless\ifnoend@
5488   \newhookcommand@series{Xendwraplemma}
5489   \newhookcommand@series{Xendwrapcontent}
5490   \newhookcommand@series{Xendnotenumfont}
5491   \newhookcommand@series{Xendlemmafонт}%
5492   \newhookcommand@series{Xendbhooknote}
5493
5494   \newhookcommand@series{Xendboxlinenum}%
5495   \newhookcommand@series{Xendboxlinenumalign}%
5496
5497   \newhookcommand@series{Xendboxstartlinenum}%
5498   \newhookcommand@series{Xendboxendlinenum}%
5499
5500   \newhookcommand@series{Xendnotefontsize}
5501   \newhooktoggle@series{Xendlemmadisablefontselection}
5502   \newhookcommand@series{Xendlemmaseparator}

```



```

5503 \newhookcommand@series{Xendbeforelemmaseparator}
5504 \newhookcommand@series{Xendafterlemmaseparator}
5505 \newhookcommand@series{Xendinplaceoflemmaseparator}
5506
5507 \newhookcommand@series{Xendbeforenumber}%
5508 \newhookcommand@series{Xendafternumber}%
5509
5510 \newhooktoggle@series{Xendparagraph}
5511 \newhookcommand@series{Xendafternote}
5512 \newhookcommand@series{Xendsep}
5513
5514 \newhookcommand@series{Xendinplaceofnumber}%
5515 \newhooktoggle@series{Xendnonumber}%
5516
5517 \newhooktoggle@series{Xendnumberonlyfirstinline}%
5518 \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
5519
5520 \newhookcommand@series{Xendsymlinenum}%
5521 \newhookcommand@series{Xendbeforesymlinenum}%
5522 \newhookcommand@series{Xendaftersymlinenum}%
5523 \newhookcommand@series{Xendboxsymlinenum}%
5524
5525 \newhookcommand@series{Xendbhooklinenumber}%
5526 \newhookcommand@series{Xendahooklinenumber}%
5527 \newhookcommand@series{Xendbhookinplaceofnumber}%
5528 \newhookcommand@series{Xendahookinplaceofnumber}%
5529
5530 \newhookcommand@series{Xendhangindent}%
5531
5532 \newhooktoggle@series{Xendpagenumberonlyfirst}%
5533 \newhooktoggle@series{Xendpagenumberonlyfirsttifsingl}%
5534 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
5535 \newhookcommand@series{Xendsympagenum}%
5536 \newhookcommand@series{Xendinplaceofpagenumber}%
5537
5538 \fi
5539 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
5540 \newhookcommand@series{Xendtwolines}[apprefwithpage,SErefwithpage]
5541 \newhookcommand@series{Xendmorethantwolines}[apprefwithpage,SErefwithpage]
5542 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage,SErefwithpage]
5543 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage,
SErefwithpage]
5544 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage]
]
5545 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
SErefonlypage]
5546 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
5547 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
5548 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
5549 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]

```

```
5550
5551 %
```

XXI.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```
5552 \newcommand{\newhooktoggle@specific}[1]{%
5553   \newtoggle{#1}%
5554   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
5555   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false}
5556   }
5557 %
```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```
5558 \newcommand{\newhookarg@specific}[1]{%
5559   \define@key[mac]{truefootnoteoption}{#1}{\global\def\linrangesep@{##1}}%
When enabling footnote option
5560   \define@key[mac]{falsefootnoteoption}{#1}{\global\undef\linrangesep@}%
When
5561   }
5562 %
```

And now, we define some hooks specific to a note.

```
5563 \newhooktoggle@specific{fulllines}%
5564 \newhooktoggle@specific{nonum}
5565 \newhooktoggle@specific{nosep}
5566 \newhookarg@specific{linrangesep}
5567 %
```

`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.

```
5568 %
```

`\nomk@` `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnotexmk`. Read `reledpar` handbook.

```
5569 \newtoggle{nomk@}%
5570 %
```

XXI.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```
5571 \newcommand*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]{}}
5572 %
```

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.

```
5573 \newcommand*{\l@ddoxtrafeet}{%
5574   \IfStrEq{familiar-critical}{\@fnpos}
5575     {\do@feetX\do@Xfeet}%
5576   {%
5577     \IfStrEq{critical-familiar}{\@fnpos}%
5578     {\do@Xfeet\do@feetX}%
5579     {%
5580       \setbox\@outputbox \vbox{%
5581         \unvbox\@outputbox%
5582         \do@feet@custom@order}{\@fnpos}%
5583       }%
5584     }%
5585   }%
5586 }%
5587 %
5588 %
```

`\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.

```
5589 \newcommand{\do@feet@custom@order}[2]{%
5590   \def\do##1{%
5591     \edef\@notesseries{\@firstoftwo##1}%
5592     \edef\@notetype{\@secondoftwo##1}%
5593     \ifdefstring{\@notetype}{critical}%
```

```

5594     {\csuse{#1append@Xnotes}{\@@notesseries}}}%
5595     {\ifdefstring{\@@notetype}{familiar}%
5596       {\csuse{#1append@notesX}{\@@notesseries}}}%
5597       {}%
5598     }%
5599   }%
5600   \expandafter\docsvlist\expandafter{#2}%
5601 }%
5602 %

```

`\do@Xfeet` `\do@Xfeet` is the code extending `\@makecol` to cater to the extra critical feet.

```

5603 \newcommand*{\do@Xfeet}{%
5604   \setbox\@outputbox \vbox{%
5605     \unvbox\@outputbox
5606     \@opXfeet}}
5607 %

```

`\@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`

```

5608 \newcommand{\append@Xnotes}[1]{%
5609   \ifvoid\csuse{#1footins}\else%
5610     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
5611     \global\advance\skip\csuse{#1footins} by\csuse{Xafterterrule@#1}%
5612     \print@Xnotes{#1}%
5613   \fi%
5614 }%
5615 %

```

The normal way to add one series, `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```

5616 \newcommand\print@Xnotes[1]{%
5617   \xdef\@currentseries{#1}%
5618   \csuse{#1footstart}{#1}%
5619   \csuse{#1footgroup}{#1}%%
5620 }%
5621 %

```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```

5622 \newcommand*{\@opXfeet}{%
5623   \unless\ifnocritical@%
5624     \gdef\firstXseries@{}%
5625     \def\do##1{%
5626       \append@Xnotes{##1}%
5627     }%
5628     \dolistloop{\@series}%
5629   \fi%

```

```
5630 }%
5631 %
```

\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.

```
5632 \newcommand*{\l@ddodoreintrafeet}{%
5633   \IfStrEq{familiar-critical}{\@fnpos}
5634     {\@doreinfeetX\X@doreinfeet}%
5635   {%
5636     \IfStrEq{critical-familiar}{\@fnpos}%
5637       {\X@doreinfeet\@doreinfeetX}%
5638       {\@doreinfeetX\X@doreinfeet}%
5639   }%
5640 }
5641
5642 %
```

\X@doreinfeet \X@doreinfeet is the code for catering for the extra critical footnotes within \@reinserts.

```
5643 \newcommand*{\X@doreinfeet}{%
5644   \unless\ifnocritical@%
5645     \def\do##1{%
5646       \ifvoid\csuse{##1footins}\else%
5647         \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
5648       \fi}%
5649     \dolistloop{\@series}
5650   \fi%
5651 }
5652
5653 %
```

\print@notesX We have to add all the new kinds of familiar footnotes to the output routine. A macro
\append@notesX which appends the familiar footnotes of one series onto the output routine, also adding
\do@feetX vertical skip before notes.

\@doreinfeetX

```
5654 \newcommand{\append@notesX}[1]{%
5655   \ifvoid\csuse{footins#1}\else%
5656     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
5657     \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
5658     \print@notesX{#1}%
5659   \fi%
5660 }%
5661 %
```

The normal way to print one series of notes. \print@Xnotes is replaced by reledpar when using \Pages.

```
5662 \newcommand\print@notesX[1]{%
5663   \xdef\@currentseries{#1}%
```

```

5664 \csuse{footstart#1}{#1}%
5665 \csuse{footgroup#1}{#1}%
5666 }%
5667 %

```

We print all the series of notes by looping on them. We check before printing them that they are not voided.

```

5668 \newcommand*{\do@feetX}{%
5669   \unless\ifnofamiliar@%
5670   \gdef\firstseriesX@{}%
5671   \setbox\@outputbox \vbox{%
5672     \unvbox\@outputbox%
5673     \def\do##1{%
5674       \append@notesX{##1}%
5675     }%
5676     \dolistloop{\@series}}%
5677   \fi%
5678 }%
5679
5680 \newcommand{\@doreinfeetX}{%
5681   \unless\ifnofamiliar@%
5682   \def\do##1{%
5683     \ifvoid\csuse{footins##1}\else
5684       \insert%
5685         \csuse{footins##1}
5686         {\unvbox\csuse{footins##1}}%
5687     \fi%
5688   }%
5689   \dolistloop{\@series}%
5690   \fi%
5691 }%
5692 %
5693 %

```

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.)

```

5694 \@ifclassloaded{memoir}{%
5695 %

```

memoir is loaded so we use memoir's built in hooks.

```

5696   \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
5697   \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinxtrafeet}%
5698 }{%
5699 %

```

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`.

```

5700 \ifboolexpr{%
5701   test{\@ifpackageloaded{fancyhdr}}%
5702   and test {\ifdef{\latex@makecol}}%
5703 }{%
5704   \patchcmd%
5705     {\latex@makecol}%
5706     {\xdef\@freelist{\@freelist\@midlist}}%
5707     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5708     {}%
5709     {\led@error@fail@patch@\@makecol}%
5710 }{%
5711   \patchcmd%
5712     {\@makecol}%
5713     {\xdef\@freelist{\@freelist\@midlist}}%
5714     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5715     {}%
5716     {\led@error@fail@patch@\@makecol}%
5717 }%
5718
5719 \patchcmd%
5720   {\@reinserts}%
5721   {\ifvbox}%
5722   {\l@ddodoreinxtrafeet\ifvbox}%
5723   {}%
5724   {\led@error@fail@patch@\@reinserts}%
5725 }
5726
5727 %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

5728 \newif\if@led@nofoot
5729
5730 %

```

```

5731 \@ifclassloaded{memoir}{%
5732 %

```

If the `memoir` class is loaded we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet533 \g@addto@macro{\@mem@extranofeet}{%%

```

```

5734 \def\do#1{%
5735   \unless\ifnocritical@%
5736     \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
5737   \fi%
5738   \unless\ifnofamiliar@%
5739     \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
5740   \fi%
5741 }
5742 \dolistloop{\@series}%
5743 }%
5744 }{%
5745 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot46 \newcommand*{\@led@testifnofoot}{%
\@doclearpage47   \@led@nofoottrue%
5748   \ifvoid\footins\else%
5749     \@led@nofootfalse%
5750   \fi%
5751   \def\do##1{%
5752     \unless\ifnocritical@%
5753       \ifvoid\csuse{##1footins}\else%
5754         \@led@nofootfalse%
5755       \fi%
5756     \fi%
5757     \unless\ifnofamiliar@%
5758       \ifvoid\csuse{footins##1}\else%
5759         \@led@nofootfalse%
5760       \fi%
5761     \fi%
5762   }%
5763   \dolistloop{\@series}%
5764 }%
5765
5766 \pretocmd%
5767 {\@doclearpage}%
5768 {\@led@testifnofoot}%
5769 {}%
5770 {\led@error@fail@patch@@doclearpage}%
5771
5772 \patchcmd%
5773 {\@doclearpage}%
5774 {\ifvoid\footins}%
5775 {\if@led@nofoot}%
5776 {}%
5777 {\led@error@fail@patch@@doclearpage}%
5778
5779 }

```



```
5780
5781 %
```

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.

```
5782 \list@create{\labelref@list}
5783 %
```

`\zz@@@` A convenience macro to zero three labeling counters in one go.

```
5784 \newcommand*{\zz@@@}{000|000|000}% Set three counters to zero in one go
5785
5786 %
```

`\edlabel` The `\edlabel` command first writes a `\@lab` macro to the `\linenum@out` file. It then checks to see that the `\labelref@list` actually has something in it (if not, it creates a dummy entry), and pops the next value for the current label, storing it in `\label@refs`. Finally it defines the label to be `\empty` so that any future check will turn up the fact that it has been used.³³

This version of the original edmac `\label` uses `\@bsphack` and `\@esphack` to eliminate extra space problems and also use the \TeX write methods for the `.aux` file.

Jesse Billett³⁴ found that the original code could be off by several pages. This version, hopefully cures that, and also allows for non-arabic page numbering.

```
5787 \newcommand*{\edlabel}[1]{%
5788   \ifl@dpairing\ifautopar%
5789     \strut%
5790     \fi\fi%
5791     \@bsphack%
5792     \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
5793       \ifXnote@%
```

³³The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

³⁴(jdb43@cam.ac.uk) via the ctt thread ‘ledmac cross referencing’, 25 August 2003.

```

5794 \protected@write\@auxout{}%
5795 {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\the\c@pstartR|{#1}}}%
5796 \ifdef{\hypertarget}%
5797 {\Hy@raisedlink{\hypertarget{#1}-{}}}%
5798 {}%
5799 \else%
5800 \write\linenum@outR{\string\@lab}%
5801 \ifx\labelref@listR\empty%
5802 \xdef\label@refs{\zz@@@}%
5803 \else%
5804 \gl@p\labelref@listR\to\label@refs%
5805 \fi%
5806 \ifvmode%
5807 \advancelabel@refs%
5808 \fi%
5809 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

5810 \protected@write\@auxout{}%
5811 {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{#1}}}%
5812 \ifdef{\hypertarget}%
5813 {\Hy@raisedlink{\hypertarget{#1}-{}}}%
5814 {}%
5815 \fi%
5816 }{%
5817 \ifXnote@%
5818 \ifl@dpairing%pstart or pstartL?
5819 \protected@write\@auxout{}%
5820 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\the\c@pstartL|{#1}}}%
5821 \ifdef{\hypertarget}%
5822 {\Hy@raisedlink{\hypertarget{#1}-{}}}%
5823 {}%
5824 \else%
5825 \protected@write\@auxout{}%
5826 {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\the\c@pstart|{#1}}}%
5827 \ifdef{\hypertarget}%
5828 {\Hy@raisedlink{\hypertarget{#1}-{}}}%
5829 {}%
5830 \fi%
5831 \else%
5832 \write\linenum@out{\string\@lab}%
5833 \ifx\labelref@list\empty%
5834 \xdef\label@refs{\zz@@@}%
5835 \else%
5836 \gl@p\labelref@list\to\label@refs%

```

```

5837 \fi%
5838 \ifvmode%
5839 \advance\label@refs%
5840 \fi%
5841 \ifl@dpairing%Pstart or PstartL?
5842 \protected@write\@auxout{}%
5843 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}%
5844 \ifdef\hypertarget{%
5845 {\Hy@raisedlink{\hypertarget{#1}}}%
5846 {}%
5847 \else%
5848 \protected@write\@auxout{}%
5849 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}%
5850 \ifdef\hypertarget{%
5851 {\Hy@raisedlink{\hypertarget{#1}}}%
5852 {}%
5853 \fi%
5854 \fi%
5855 }%
5856 \@esphack}%
5857
5858 %

```

`\advance\label@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 `\advance\label@refs` command.

```

5859 \newcounter{line}%
5860 \newcounter{subline}%
5861 \newcounter{absline}%
5862 \newcommand{\advance\label@refs}{%
5863 \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
5864 \stepcounter{line}%
5865 \setcounter{absline}{\expandafter\labelrefsparseabsline\label@refs}%
5866 \stepcounter{absline}%
5867 \ifsublines@%
5868 \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
5869 %
5870 \stepcounter{subline}{1}%
5871 \def\label@refs{\theline|\thesubline|\theabsline}%
5872 \else%
5873 \def\label@refs{\theline|0|\theabsline}%
5874 \fi%
5875 }
5876 \def\labelrefsparseline#1|#2|#3{#1}%
5877 \def\labelrefsparsesubline#1|#2|#3{#2}%

```

```

5877 \def\labelrefsparseabsline#1|#2|#3{#3}%
5878 %

```

\l@dmake@labels The `\l@dmake@labels` macro gets executed when the labels file is read. For each label it defines a macro, whose name is made up partly from the label you supplied, that contains the page, line and sub-line numbers. But first it checks to see whether the label has already been used (and complains if it has).

The initial use of `\newcommand` is to catch if `\l@dmake@labels` has been previously defined (by a class or package).

#1 page number, #2 line number, #3 sub-line number, #4 absolute line number, #5 pstart number, #6 label.

```

5879 \newcommand*\l@dmake@labels{}
5880 \def\l@dmake@labels#1|#2|#3|#4|#5|#6{%
5881   \expandafter\ifx\csname the@label\csuse{XR@prefix}#6\endcsname%
5882     \relax%
5883   \else%
5884     \led@warn@DuplicateLabel{\csuse{XR@prefix}#6}%
5885   \fi
5886   \global\providetoggle{label@#6@ledRcol}%False is the default value of
5887   this toggle, which tell us if a label is linked to a right or a left side
5888   \expandafter\gdef\csname the@label\csuse{XR@prefix}#6\endcsname
5889   {#1|#2|#3|#4|#5|\relax}%
5890   \ignorespaces}
5891 %

```

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.

```

5891 \AtBeginDocument{%
5892   \def\l@dmake@labels#1|#2|#3|#4|#5|#6{}%
5893 }
5894 %
5895 %

```

\@lab The `\@lab` command, which appears in the `\linenum@out` file, appends the current value of page, line, sub-line, and absolute line to the `\labelref@list`. These values are defined by the earlier `\@page`, `\@nl`, and the `\sub@on` and `\sub@off` commands appearing in the `\linenum@out` file.

TeX uses the page counter for page numbers. However, it appears that this is not the right place to grab the page number. That task is now done in the `\edlabel` macro. This version of `\@lab` appends just the current line and sub-line numbers to `\labelref@list`.

```

5896 \newcommand*\@lab{%
5897   \ifledRcol
5898     \xright@appenditem{\linenumr@p{\line@numR}}|%
5899

```



```

5933     {\string\l@make@labels\space\l@parsedstartpage|\
l@parsedstartline|\l@parsedstartsub||\the\c@pstartL|{#1:start}}}%
5934     \ifdef{\hypertarget}%
5935     {\Hy@raisedlink{\hypertarget{#1:start}}{}}}%
5936     {}%
5937     \protected@write\@auxout{}%
5938     {\string\l@make@labels\space\l@parsedendpage|\
l@parsedendline|\l@parsedendsub||\the\c@pstartL|{#1:end}}}%
5939     \else%
5940     \protected@write\@auxout{}%
5941     {\string\l@make@labels\space\l@parsedstartpage|\
l@parsedstartline|\l@parsedstartsub||\the\c@pstartL|{#1:start}}}%
5942     \ifdef{\hypertarget}%
5943     {\Hy@raisedlink{\hypertarget{#1:start}}{}}}%
5944     {}%
5945     \protected@write\@auxout{}%
5946     {\string\l@make@labels\space\l@parsedendpage|\
l@parsedendline|\l@parsedendsub||\the\c@pstartL|{#1:end}}}%
5947     \fi%
5948     \fi%
5949     %

```

Use the \TeX standard hack for label.

```

5950     \@esphack%
5951     %

```

Warning if `\applabel` is called outside of `\edtext`.

```

5952     \else%
5953     \led@warn@AppLabelOutSecondArgEdtext{#1}%
5954     \fi%
5955     %

```

End of `\applabel`

```

5956 }%
5957 %

```

`\edlabels` `\edlabelS` and `\edlabelE` are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
5958 \newcommand{\edlabelS}[1]{%
5959   \edlabel{#1:start}%
5960 }
5961 \newcommand{\edlabelE}[1]{%
5962   \edlabel{#1:end}%
5963 }
5964 \newcommand{\edlabelSE}[1]{%
5965   \edlabelS{#1}%
5966   \edlabelE{#1}%
5967 }
5968 %

```

\wrap@edcrossref \wrap@edcrossref is called around all reledmac crossref commands, except those which start with x. It adds the hyperlink.

```

5969 \newrobustcmd{\wrap@edcrossref}[2]{%
5970   \ifdef{\hyperlink}%
5971     {\hyperlink{#1}{#2}}%
5972   {#2}%
5973 }
5974 %

```

\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.

```

5975 \newcommand*\edpageref[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
5976 \newcommand*\xpageref[1]{\l@dgetref@num{1}{#1}}
5977
5978 %

```

\edlineref If the specified label exists, \lineref gives its line number.

```

5979 \newcommand*\edlineref[1]{%
5980   \l@dref@undefined{#1}%
5981   \wrap@edcrossref{#1}{%
5982     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
5983     \iftoggle{label@#1@ledRcol}%
5984       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
5985       {\linenumrep{\l@dgetref@num{2}{#1}}}%
5986     \xflagref{#1}%
5987   }%
5988 }%
5989 \newcommand*\xlineref[1]{\l@dgetref@num{2}{#1}}%
5990
5991 %

```

\sublineref If the specified label exists, \sublineref gives its sub-line number.

```

5992 \newcommand*\sublineref[1]{%
5993   \l@dref@undefined{#1}%
5994   \wrap@edcrossref{#1}{%
5995     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
label has not yet been parsed on the .aux file
5996     \iftoggle{label@#1@ledRcol}%

```

```

5997     {\sublinenumrepR{\l@getref@num{3}{#1}}}%
5998     {\sublinenumrep{\l@getref@num{3}{#1}}}%
5999   }%
6000 }%
6001 \newcommand*{\xsublineref}[1]{\l@getref@num{3}{#1}}
6002
6003 %

```

\xabslineref If the specified label exists, `\xabslineref` gives its absolute line number. That is used usually only by some `reledmac` internal macros.

```

6004 \newcommand*{\xabslineref}[1]{\l@getref@num{4}{#1}}%
6005 %

```

\pstartref If the specified label exists, `\pstartref` gives its `pstart` number.

```

\pstartref
6006 \newcommand*{\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@getref@num{5}{#1}}}%
6007 \newcommand*{\xpstartref}[1]{\l@getref@num{5}{#1}}%
6008
6009 %

```

\xflagref `\xflagref` finds the side flag of any ref defined with `\edlabel`.

```

6010 \newcommand*{\xflagref}[1]{\l@getref@num{6}{#1}}%
6011 %

```

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.

```

6012 \newcommand*{\l@dref@undefined}[1]{%
6013   \expandafter\ifx\csname the@label#1\endcsname\relax
6014     \led@warn@RefUndefined{#1}%
6015   \fi}
6016
6017 %

```

\l@getref@num Next, `\l@getref@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@dlabell@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`.


```

6018 \newcommand*\l@dgetref@num}[2]{%
6019   \expandafter
6020   \ifx\csname the@label#2\endcsname \relax
6021     000%
6022   \else
6023     \expandafter\expandafter\expandafter
6024     \l@dlabel@parse\csname the@label#2\endcsname|#1%
6025   \fi}
6026
6027 %

```

\l@dlabel@parse Notice that we slipped another | delimiter into the penultimate line of \l@dgetref@num, to keep the ‘switch-number’ separate from the reference numbers. This | is used as another parameter delimiter by \l@dlabel@parse, which extracts the appropriate number from its first arguments. The |-delimited arguments consist of the expanded label-macro (three reference numbers), followed by the switch-number (1, 2, 3, 4 or 5) which defines which of the earlier six numbers to pick out. (It was earlier given as the first argument of \l@dgetref@num.)

```

6028 \newcommand*\l@dlabel@parse#1{}
6029 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7{%
6030   \ifcase #7%
6031     \or #1%
6032     \or #2%
6033     \or #3%
6034     \or #4%
6035     \or #5%
6036     \or #6%
6037   \fi}
6038 %

```

\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.

```

6039 \newcommand*\xxref}[2]{%
6040   {%
6041     \expandafter\ifx\csname the@label#1\endcsname \relax%
6042       \expandafter\let\csname the@@label#1\endcsname\zz@@@%
6043     \else%
6044       \expandafter\def\csname the@@label#1\endcsname{\l@dgetref@num
6045         {1}{#1}|\l@dgetref@num{2}{#1}|\l@dgetref@num{3}{#1}}%
6046       \fi%
6047     \expandafter\ifx\csname the@label#2\endcsname \relax%

```

```

6047 \expandafter\let\csname the@@label#2\endcsname\zz@@@%
6048 \else%
6049 \expandafter\def\csname the@@label#2\endcsname{\l@dgetref@num
6050 {1}{#2}|\l@dgetref@num{2}{#2}|\l@dgetref@num{3}{#2}}%
6051 \fi%
6052 \letcs{\@tempa}{the@@label#1}%
6053 \letcs{\@tempb}{the@@label#2}%
6054 \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@start{#1}}%
6055 \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@end{#2}}%
6056 \linenum{\@tempa|}%
6057 \@tempb}}}%
6058 %

```

`\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. `\SErefwithpage` So, first declare the default value of the hooks for the pseudo-series. Also declare the internal toggle which are switch by `reledmac`.

```

6059 \def\Xtwolines@appref{}%
6060 \def\Xtwolines@SEref{}%
6061
6062 \def\Xmorethantwolines@appref{}%
6063 \def\Xmorethantwolines@SEref{}%
6064
6065 \def\Xlinerangeseparator@appref{\endashchar}%
6066 \def\Xlinerangeseparator@SEref{\endashchar}%
6067
6068 \def\Xsublinesep@appref{\fullstop}%
6069 \def\Xsublinesep@SEref{\fullstop}%
6070
6071 \def\Xpagelinesep@appref{\fullstop}%
6072 \def\Xpagelinesep@SEref{\fullstop}%
6073
6074
6075 \newtoggle{Xtwolinesbutnotmore@appref}%
6076 \newtoggle{Xtwolinesbutnotmore@SEref}%
6077
6078 \newtoggle{Xtwolinesonlyinsamepage@appref}%
6079 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
6080
6081 \newtoggle{Xlineflag@appref}%
6082 \toggletrue{Xlineflag@appref}%Here exception
6083 \newtoggle{Xlineflag@SEref}%
6084 \toggletrue{Xlineflag@SEref}%%Here exception
6085

```

```

6086 \def\Xendtwolines@apprefwithpage{}%
6087 \def\Xendtwolines@SErefwithpage{}%
6088
6089
6090 \def\Xendmoreethantwolines@apprefwithpage{}%
6091 \def\Xendmoreethantwolines@SErefwithpage{}%
6092
6093 \def\Xendlinrangeseparator@apprefwithpage{\endashchar}
6094 \def\Xendlinrangeseparator@SErefwithpage{\endashchar}
6095 \def\Xendlinrangeseparator@SErefonlypage{\endashchar}
6096
6097 \def\Xendbeforepagenumber@apprefwithpage{p.}%
6098 \def\Xendbeforepagenumber@SErefwithpage{p.}%
6099 \def\Xendbeforepagenumber@SEonlypage{p.}%
6100
6101 \def\Xendafterpagenumber@apprefwithpage{ }%
6102 \def\Xendafterpagenumber@SErefwithpage{ }%
6103
6104
6105 \def\Xendlineprefixsingle@apprefwithpage{}%
6106 \def\Xendlineprefixsingle@SErefwithpage{}%
6107
6108 \def\Xendlineprefixmore@apprefwithpage{}%
6109 \def\Xendlineprefixmore@SErefwithpage{}%
6110
6111 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
6112 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
6113
6114 \def\Xendsublinesep@apprefwithpage{\fullstop}%
6115 \def\Xendsublinesep@SErefwithpage{\fullstop}%
6116
6117 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
6118 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
6119
6120 \newtoggle{Xendlineflag@apprefwithpage}
6121 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
6122 \newtoggle{Xendlineflag@SErefwithpage}
6123 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
6124
6125 %

```

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.

```

6126
6127 \gdef\Xboxstartlinenum@appref{0pt}
6128 \gdef\Xboxstartlinenum@SEref{0pt}
6129
6130 \gdef\Xboxendlinenum@appref{0pt}
6131 \gdef\Xboxendlinenum@SEref{0pt}

```

```

6132 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}
6133 \gdef\Xendboxstartlinenum@SErefwithpage{Opt}
6134
6135 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
6136 \gdef\Xendboxendlinenum@SErefwithpage{Opt}
6137
6138 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
6139 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
6140
6141 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
6142 \newtoggle{Xendpagenumberonlyfirstifsingle@SErefwithpage}
6143
6144 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
6145 \newtoggle{Xendpagenumberonlyfirstintwo@SErefwithpage}
6146
6147 \gdef\Xendsympagenum@apprefwithpage{}
6148 \gdef\Xendsympagenum@SErefwithpage{}
6149
6150 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
6151 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
6152
6153 %
6154 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

6155 \newcommand\@apprefprefixsingle{}%
6156 \newcommand\@SErefprefixsingle{}%
6157
6158 \newcommand\@apprefprefixmore{}%
6159 \newcommand\@SErefprefixmore{}%
6160
6161 \newcommand{\setapprefprefixsingle}[1]{%
6162   \gdef\@apprefprefixsingle{#1}%
6163 }
6164 \newcommand{\setSErefprefixsingle}[1]{%
6165   \gdef\@SErefprefixsingle{#1}%
6166 }
6167
6168 \newcommand{\setapprefprefixmore}[1]{%
6169   \gdef\@apprefprefixmore{#1}%
6170 }
6171 \newcommand{\setSErefprefixmore}[1]{%
6172   \gdef\@SErefprefixmore{#1}%
6173 }
6174
6175 %

```

And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.

```

6176 \newcommand{\setSErefonlypageprefixsingle}[1]{%

```

```

6177 \gdef\Serefonlypage@prefixsingle{#1}%
6178 }%
6179 \newcommand{\setSerefonlypageprefixmore}[1]{%
6180 \gdef\Serefonlypage@prefixmore{#1}%
6181 }%
6182 %

```

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.

```

6183
6184 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
6185 \newcommandx{\Seref}[2][1,usedefault]{\reformatted@{#1}{#2}{Seref}}
6186
6187 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
6188 {#1}{#2}{appref}}
6189 \newcommandx{\Serefwithpage}[2][1,usedefault]{\reformattedwithpage@
6190 {#1}{#2}{Seref}}
6191 \newcommandx{\Serefonlypage}[2][1,usedefault]{\reformattedonlypage@
6192 {#1}{#2}{Seref}}
6193
6194 \newcommand{\reformatted@}[3]{%
6195 \def\do##1{%
6196 \setkeys[mac]{truefootnoteoption}{##1}%
6197 }%
6198 \notblank{#1}{\docsvlist{#1}}}%
6199 \xdef\@currentseries{#3}%
6200 \ifcempty{#3prefixmore}%
6201 {\@apprefprefixsingle}%
6202 {\%
6203 \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}}%
6204 {\csuse{#3prefixsingle}}}%
6205 {\csuse{#3prefixmore}}}%
6206 }%
6207 \ifboolexpr{%
6208 test{\ifcsundef{the@label#2:start}}}%
6209 or test{\ifcsundef{the@label#2:end}}}%
6210 }%
6211 {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6212 {\%
6213 \def\@this@crossref@start{#2:start}%
6214 \def\@this@crossref@end{#2:end}%
6215 \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
6216 start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}}|}%
6217 \undef\@this@crossref@end%
6218 \undef\@this@crossref@start%
6219 }%

```

```

6217 \def\do##1{%
6218   \setkeys[mac]{falsefootnoteoption}{##1}%
6219 }%
6220 \notblank{#1}{\docsvlist{#1}}{}%
6221 }%
6222
6223 \newcommand{\reformattedwithpage@}[3]{%
6224   \def\do##1{%
6225     \setkeys[mac]{truefootnoteoption}{##1}%
6226   }%
6227   \notblank{#1}{\docsvlist{#1}}{}%
6228   \xdef\@currentseries{#3withpage}%
6229   \ifboolexpr{%
6230     test{\ifcsundef{the@label#2:start}}%
6231     or test{\ifcsundef{the@label#2:end}}%
6232   }%
6233   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6234   {%
6235     \def\@this@crossref@start{#2:start}%
6236     \def\@this@crossref@end{#2:end}%
6237     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|}%
6238     \undef\@this@crossref@end%
6239     \undef\@this@crossref@start%
6240   }%
6241   \def\do##1{%
6242     \setkeys[mac]{falsefootnoteoption}{##1}%
6243   }%
6244   \notblank{#1}{\docsvlist{#1}}{}%
6245 }%
6246
6247 \newcommand{\reformattedonlypage@}[3]{%
6248   \def\do##1{%
6249     \setkeys[mac]{truefootnoteoption}{##1}%
6250   }%
6251   \notblank{#1}{\docsvlist{#1}}{}%
6252   \xdef\@currentseries{#3onlypage}%
6253   \ifboolexpr{%
6254     test{\ifcsundef{the@label#2:start}}%
6255     or test{\ifcsundef{the@label#2:end}}%
6256   }%
6257   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6258   {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
6259     {%
6260       \ifcsvoid{#3onlypage@prefixsingle}%
6261       {}%
6262       {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
}}%
6263     }%
6264     \printnpnum%

```

```

6264 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
6265 }%
6266 }%
6267 {%
6268 \ifcsvoid{#3onlypage@prefixmore}%
6269 {}%
6270 {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
6271 \ifdefined\linrangesep%
6272 \printnpnum{%
6273 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
6274 \linrangesep@%
6275 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
6276 }%
6277 \else%
6278 \printnpnum{%
6279 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}}%
6280 \csuse{Xendlinrangeseparator@}\@currentseries}%
6281 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}}%
6282 }%
6283 \fi%
6284 }%
6285 }%
6286 \def\do##1{%
6287 \setkeys[mac]{falsefootnoteoption}{##1}%
6288 }%
6289 \notblank{#1}{\docsvlist{#1}}}%
6290 }%
6291 %

```

\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`.

```

6292 \newcommand*{\edmakelabel}[2]{\expandafter\edef\csname the@label#1\
6293 endcsname{#2}}
6294 %

```

(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. 130 and V.9 p. 97), 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`.

```
6295 \pretocmd{\XR@test}%
6296   {\XR@test@mac+++#1#2#3#4+++}%
6297   {}%
6298   {}%
6299   %
```

`\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`.

```
6300 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
6301 %
```

`\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`.

```
6302 \long\def\xR@test@mac@test#1#2...{%The triple dots (NOT \ldots) are because
of the line 22 of xr.sty v5.02 1994/05/28
6303   \ifx#1\l@dmake@labels%
6304     \l@dmake@labels#2%
6305   \else
6306     \ifx#1\l@dmake@labelsR%
6307       \l@dmake@labelsR #2%
6308     \fi%
6309   \fi%
6310 }%
6311 %
```

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.

```
6312 \pretocmd{\@xympar}%
6313   {\ifnumberedpar@
6314     \led@warn@NoMarginpars
6315     \@esphack
6316   \else}%
6317   {}%
6318   {}%
6319
```



```

6320 \apptocmd{\@xympar}%
6321   {\fi}%
6322   {}
6323   {}
6324
6325 %

```

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

```

6326 \newcount\sidenote@margin
6327 \newcommand*{\sidenotemargin}[1]{\{
6328   \l@dgetsidenote@margin{#1}%
6329   \ifnum\@l@tempcntb>\m@ne
6330     \ifledRcol
6331       \global\sidenote@marginR=\@l@tempcntb
6332     \else
6333       \global\sidenote@margin=\@l@tempcntb
6334     \fi
6335   \fi}}
6336 \newcommand*{\l@dgetsidenote@margin}[1]{\{
6337   \def\@tempa{#1}\def\@tempb{left}%
6338   \ifx\@tempa\@tempb
6339     \@l@tempcntb \z@
6340   \else
6341     \def\@tempb{right}%
6342     \ifx\@tempa\@tempb
6343       \@l@tempcntb \@ne
6344     \else
6345       \def\@tempb{outer}%
6346       \ifx\@tempa\@tempb
6347         \@l@tempcntb \tw@
6348       \else
6349         \def\@tempb{inner}%
6350         \ifx\@tempa\@tempb
6351           \@l@tempcntb \thr@@
6352         \else
6353           \led@warn@BadSidenotemargin

```

```

6354 \l@dttempcntb \m@ne
6355 \fi
6356 \fi
6357 \fi
6358 \fi}
6359 \sidenotemargin{right}
6360
6361 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
6362 \newbox\l@dlp@rbox
6363 \newbox\l@drp@rbox
6364
6365 %

```

`\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
6366 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
6367 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
6368 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup
6369 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
6370 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
6371 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
6372
6373 %

```

`\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
6374 \newcommand*\ledleftnote[1]{\edtext{}\l@dlsnote{#1}}
6375 \newcommand*\ledrightnote[1]{\edtext{}\l@drsnote{#1}}
6376 \newcommand*\ledsidenote[1]{\edtext{}\l@dcnote{#1}}%
6377 \newcommand*\ledinnernote[1]{\edtext{}\l@disnote{#1}}%
6378 \newcommand*\ledouternote[1]{\edtext{}\l@dosnote{#1}}%
6379 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent
`\l@drsnote` of the critical footnotes code.

```

\l@dcnote
6380 \newif\ifrighnoteup
\l@desnote
6381 \righnoteuptrue
\l@disnote
6382
6383 \newcommand*\l@dlsnote[1]{%
6384 \begingroup%
6385 \newcommand{\content}{#1}%
6386 \ifnumberedpar@

```

```

6387 \ifledRcol%
6388 \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}}%
6389 \to\inserts@listR
6390 \global\advance\insert@countR \@ne%
6391 \else%
6392 \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}}%
6393 \to\inserts@list
6394 \global\advance\insert@count \@ne%
6395 \fi
6396 \fi%
6397 \ignorespaces%
6398 \endgroup%
6399 }%
6400
6401 \newcommand*{\l@drsnote}[1]{%
6402 \begingroup%
6403 \newcommand{\content}{#1}%
6404 \ifnumberedpar@
6405 \ifledRcol%
6406 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
6407 \to\inserts@listR
6408 \global\advance\insert@countR \@ne%
6409 \else%
6410 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
6411 \to\inserts@list
6412 \global\advance\insert@count \@ne%
6413 \fi
6414 \fi\ignorespaces%
6415 \endgroup%
6416 }%
6417
6418 \newcommand*{\l@dcsnote}[1]{%
6419 \begingroup%
6420 \newcommand{\content}{#1}%
6421 \ifnumberedpar@
6422 \ifledRcol%
6423 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
6424 \to\inserts@listR
6425 \global\advance\insert@countR \@ne%
6426 \else%
6427 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
6428 \to\inserts@list
6429 \global\advance\insert@count \@ne%
6430 \fi
6431 \fi\ignorespaces%
6432 \endgroup%
6433 }%
6434
6435 \newcommand*{\l@disnote}[1]{%
6436 \begingroup%

```

```

6437 \newcommand{\content}{#1}%
6438 \ifnumberedpar{%
6439   \ifledRcol%
6440     \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}%
6441       \to\inserts@listR%
6442     \global\advance\insert@countR \@ne%
6443   \else%
6444     \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}%
6445       \to\inserts@list%
6446     \global\advance\insert@count \@ne%
6447   \fi%
6448   \fi\ignorespaces%
6449 \endgroup%
6450 }%
6451
6452 \newcommand*{\l@dosnote}[1]{%
6453   \begingroup%
6454   \newcommand{\content}{#1}%
6455   \ifnumberedpar{%
6456     \ifledRcol%
6457       \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}%
6458         \to\inserts@listR%
6459     \global\advance\insert@countR \@ne%
6460   \else%
6461     \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}%
6462       \to\inserts@list%
6463     \global\advance\insert@count \@ne%
6464   \fi%
6465   \fi\ignorespaces%
6466 \endgroup%
6467 }%
6468
6469 %

```

\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.

```

6470 \newcommand*{\vl@dlsnote}[1]{%
6471   \ifledRcol{%
6472     \@l@tempcntb=\sidenote@marginR%
6473     \ifnum\@l@tempcntb>\@ne%
6474       \advance\@l@tempcntb by\page@numR%
6475     \fi%
6476   \else%
6477     \@l@tempcntb=\sidenote@margin%
6478     \ifnum\@l@tempcntb>\@ne%
6479       \advance\@l@tempcntb by\page@num%
6480     \fi%
6481   \fi%
6482   \ifodd\@l@tempcntb%
6483     \listgadd{\l@dcsnotetext@l}{#1}%
6484   \else%
6485     \listgadd{\l@dcsnotetext}{#1}%
6486   \fi
6487 }
6488 \newcommand*{\vl@drsnote}[1]{%
6489   \ifledRcol{%
6490     \@l@tempcntb=\sidenote@marginR%
6491     \ifnum\@l@tempcntb>\@ne%
6492       \advance\@l@tempcntb by\page@numR%
6493     \fi%
6494   \else%
6495     \@l@tempcntb=\sidenote@margin%
6496     \ifnum\@l@tempcntb>\@ne%
6497       \advance\@l@tempcntb by\page@num%
6498     \fi%
6499   \fi%
6500   \ifodd\@l@tempcntb%
6501     \listgadd{\l@dcsnotetext}{#1}%
6502   \else%
6503     \listgadd{\l@dcsnotetext@r}{#1}%
6504   \fi%
6505 }
6506 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
6507
6508 \newcommand{\vl@disnote}[1]{%
6509   \ifledRcol{%
6510     \@tempcnta=\page@numR%
6511   \else%
6512     \@tempcnta=\page@num%
6513   \fi%
6514   \ifodd\@tempcnta% ODD => right page => inner side = left side
6515     \vl@dlsnote{#1}%

```

```

6516 \else%
6517 \vl@drsnote{#1}%
6518 \fi%
6519 }%
6520
6521 \newcommand{\vl@dosnote}[1]{%
6522 \ifledRcol%
6523 \@tempcnta=\page@numR%
6524 \else%
6525 \@tempcnta=\page@num%
6526 \fi%
6527 \ifodd\@tempcnta% ODD => right page => outer side = right side
6528 \vl@drsnote{#1}%
6529 \else%
6530 \vl@dlsnote{#1}%
6531 \fi%
6532 }%
6533
6534 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box. And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

6535 \newcommand*{\setl@dlp@rbox}[1]{%
6536 \begingroup%
6537 \parindent\z@\hsize=\ledlsnotewidth%
6538 \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
6539 \global\setbox\l@dlp@rbox%
6540 \ifleftnoteup%
6541 =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
6542 \else%
6543 =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}}%
6544 \fi%
6545 \endgroup%
6546 }
6547
6548 \newcommand*{\setl@drp@rbox}[1]{%
6549 \begingroup%
6550 \parindent\z@\hsize=\ledrsnotewidth%
6551 \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
6552 \global\setbox\l@drp@rbox%
6553 \ifrightrightnoteup%
6554 =\vbox to\z@{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.

```

```

6555 \else%
6556   =\vbox to0.7\baselineskip{{\ledrsnotefontsetup\strut#1\vss}}}%
6557   \fi%
6558 \endgroup%
6559 }%
6560 \newif\ifleftnoteup
6561 \leftnoteuptrue
6562 %

```

\@sidenotesep This macro is used to separate sidenotes of the same line.

```

6563 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
6564 \newcommand{\@sidenotesep}{, }
6565 %

```

\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.

```

6566 \newcommand*{\affixside@note}{%
6567   \prepare@edindex@for@note{\the\page@num|\the\line@num|\the\subline@num|\
the\page@num|\the\line@num|\the\subline@num}%
6568   \def\sidenotecontent@{}%
6569   \numgdef{\itemcount@}{0}%
6570   \def\do##1{%
6571     \ifnumequal{\itemcount@}{0}%
6572     {%
6573       \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
6574     {\appto\sidenotecontent@{\@sidenotesep ##1}}%
6575     }%
6576     \numgdef{\itemcount@}{\itemcount@+\@ne}%
6577   }%
6578   \dolistloop{\l@dcnotesetext}%
6579   \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
6580 %

```

And we do the same for left and right notes (not movable).

```

6581 \gdef\@templ@d{%
6582 \gdef\@templ@n{\l@dcnotesetext\l@dcnotesetextl\l@dcnotesetext@r}%
6583 \ifx\@templ@d\@templ@n \else%
6584   \if@twocolumn%
6585     \if@firstcolumn%
6586       \setl@dlp@rbox{##1}{\sidenotecontent@}%
6587     \else%
6588       \setl@drp@rbox{\sidenotecontent@}%
6589     \fi%

```

```

6590 \else%
6591 \l@dttempcntb=\sidenote@margin%
6592 \ifnum\l@dttempcntb>\@ne%
6593 \advance\l@dttempcntb by\page@num%
6594 \fi%
6595 \ifodd\l@dttempcntb%
6596 \setl@drp@rbox{\sidenotecontent@}%
6597 \gdef\sidenotecontent@{%
6598 \numgdef{\itemcount@}{0}%
6599 \dolistloop{\l@dcsnotetext@l}%
6600 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
6601 \setl@dlp@rbox{\sidenotecontent@}%
6602 \else%
6603 \setl@dlp@rbox{\sidenotecontent@}%
6604 \gdef\sidenotecontent@{%
6605 \numgdef{\itemcount@}{0}%
6606 \dolistloop{\l@dcsnotetext@r}%
6607 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
6608 \setl@drp@rbox{\sidenotecontent@}%
6609 \fi%
6610 \fi%
6611 \fi%
6612 \advance\@edindex@fornote@\m@ne%
6613 }
6614 %

```

XXV Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage`.
`\l@dfetendmini` They can be extended to handle other things if necessary.

```

6615 \ifnoledgroup@else%
6616 \newcommand*{\l@dfetbeginmini}{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
6617 \newcommand*{\l@dfetendmini}{%
6618 \IfStrEq{critical-familiar}{\@mpfnpos}%
6619 {\l@dedendmini\l@dfamendmini}%
6620 {%
6621 \IfStrEq{familiar-critical}{\@mpfnpos}%
6622 {\l@dfamendmini\l@dedendmini}%
6623 {\do@feet@custom@order{mp@}{\@mpfnpos}}%
6624 }%
6625 }%
6626 %

```


`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

`\l@dedendmini`

`\mp@append@Xnotes`

```

6627 \newcommand*{\l@dedbeginmini}{%
6628   \unless\ifnocritical%
6629   \def\do##1{%
6630     \csletcs{v##1footnote}{mpv##1footnote}%
6631   }%
6632   \dolistloop{\@series}%
6633   \fi%
6634 }
6635 \newcommand*{\l@dedendmini}{%
6636   \unless\ifnocritical%
6637   \ifl@dpairing%
6638     \ifledRcol%
6639       \flush@notesR%
6640     \else%
6641       \flush@notes%
6642     \fi%
6643   \fi
6644   \def\do##1{%
6645     \mp@append@Xnotes{##1}%
6646   }%
6647   \dolistloop{\@series}%
6648   \fi%
6649 }%
6650 \newcommand{\mp@append@Xnotes}[1]{%
6651   \ifvoid\csuse{mp#1footins}\else%
6652     \ifl@dpairing%
6653       \ifparledgroup%
6654         \ifledRcol%
6655           \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
6656         \else%
6657           \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\skip
\@nameuse{mp#1footins}}%
6658         \fi%
6659       \fi%
6660     \fi%
6661     \ifcsstring{series@display#1}{paragraph}{%}{%
6662       \setbox\@nameuse{mp#1footins}=\vbox{%
6663         \csuse{Xnotefontsize@#1}%
6664         \ifcsdef{Xhsize}\csuse{series@display#1}@#1{%
6665           \hsize\csuse{Xhsize}\csuse{series@display#1}@#1}%
6666         }{%
6667           \noindent\csuse{Xtxtbeforenotes@#1}%
6668           \unvbox\@nameuse{mp#1footins}%
6669           \@parboxrestore%
6670         }%
6671       }%
6672       \csuse{mp#1footgroup}{#1}%

```

```

6673 \fi%
6674 }%
6675 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
6676 \newcommand*\l@dfambeginmini{%
6677 \unless\ifnofamiliar%
6678 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
6679 \dolistloop{\@series}%
6680 \fi%
6681 }%
6682
6683 \newcommand*\l@dfamendmini{%
6684 \unless\ifnofamiliar%
6685 \def\do##1{%
6686 \mp@append@notesX{##1}%
6687 }%
6688 \dolistloop{\@series}%
6689 \fi%
6690 }%
6691 \newcommand{\mp@append@notesX}[1]{%
6692 \ifvoid\csuse{mpfootins#1}\else%
6693 \csuse{mpfootgroup#1}{#1}%
6694 \fi%
6695 }%
6696 %

```

`\@iiminipage` This is our extended form of the kernel `\@iiminipage` defined in `ltboxes.dtx`.

```

6697 \patchcmd%
6698 {\@iiminipage}%
6699 {\let\@footnotetext\@mpfootnotetext}%
6700 {\let\@footnotetext\@mpfootnotetext\l@dfeetbeginmini}%
6701 {}%
6702 {\led@error@fail@patch@\@iiminipage}%
6703 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

6704 \patchcmd%
6705 {\endminipage}%
6706 {\footnoterule}%
6707 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
6708 {}%
6709 {\led@error@fail@patch@endminipage}%
6710
6711 \patchcmd%
6712 {\endminipage}%
6713 {\@minipagefalse}%

```

```

6714 {\l@dfteetendmini\@minipagefalse}%
6715 {}%
6716 {\led@error@fail@patch@endminipage}
6717
6718 %

```

`\l@dunboxmpfoot` `\@ldunboxmpfoot` insert normal footnotes for `ledgroup`.

`edgroup@beforenormalnotes`

```

6719 \newcommand*{\l@dunboxmpfoot}{%
6720   \vskip\skip\@mpfootins
6721   \normalcolor
6722   \footnoterule
6723   \l@advance@parledgroup@beforenormalnotes
6724   \unvbox\@mpfootins%
6725 }
6726 %

```

When using parallel `ledgroup`, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

6727 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
6728   \ifparledgroup
6729     \ifl@pairing
6730       \ifledRcol
6731         \dimdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
6732       \else
6733         \dimdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
6734       \fi
6735     \fi
6736   \fi
6737 }
6738 %

```

`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`.

```

6739 \newenvironment{ledgroup}{%
6740   \resetprevpage@num%
6741   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
6742   \let\@footnotetext\@mpfootnotetext
6743   \l@dfeetbeginmini%
6744 }{%
6745   \par
6746   \unskip
6747   \ifvoid\@mpfootins\else
6748     \l@dunboxmpfoot
6749

```

```

6750 \fi
6751 \l@dfeetendmini%
6752 \@ledgroupfalse%
6753 }
6754
6755
6756 %

```

```

\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.

```

6757 \newenvironment{ledgroupsize}[2][1]{%
6758 %

```

Set the various text measures.

```

6759 \hsize #2\relax
6760 %

```

Initialize fills for centering.

```

6761 \let\ledllfill\hfil
6762 \let\ledrlfill\hfil
6763 \def\@tempa{#1}\def\@tempb{1}%
6764 %

```

Left adjusted numbered lines

```

6765 \ifx\@tempa\@tempb
6766 \let\ledllfill\relax
6767 \else
6768 \def\@tempb{r}%
6769 \ifx\@tempa\@tempb
6770 %

```

Right adjusted numbered lines

```

6771 \let\ledrlfill\relax
6772 \fi
6773 \fi
6774 %

```

Set up the footnoting.

```

6775 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
6776 \let\@footnotetext\@mpfootnotetext
6777 \l@dfeetbeginmini%
6778 }{
6779 \par
6780 \unskip
6781 \ifvoid\@mpfootins\else
6782 \l@dunboxmpfoot
6783 \fi

```

```

6784 \l@dfetendmini%
6785 }
6786
6787 %

```

Close the \ifnoledgroup@\else.

```

6788 \fi%
6789 %

```

`\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`.

```

6790 \newif\ifledgroupnotesL@
6791 \newif\ifledgroupnotesR@
6792 %

```

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`.

```

6793 \AtBeginDocument{%
6794   \unless\ifl@imakeidx%
6795     \@ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
6796   %
6797   \fi%
6798   \unless\ifl@indextools%
6799     \@ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}
6800   %
6801   \fi%
6802   \unless\ifl@footmisc%
6803     \@ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
6804   %
6805   \fi%
6806 }
6807 %

```

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
6805 \newcommand{\pagelinesep}{-}
6806 \newcommand{\edindexlab}{${\&}}
6807 \newcounter{labidx}

```

```

6808 \setcounter{labidx}{0}
6809
6810 %

```

\doedindexlabel This macro sets an \edlabel.

```

6811 \newcommand{\doedindexlabel}{%
6812   \stepcounter{labidx}%
6813   \edlabel{\edindexlab\thelabidx}%
6814 }
6815
6816 %

```

\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.

```

6817 \newcounter{pageline}%
6818 \renewcommand{\thepageline}{%
6819   \thepage%
6820   \pagelinesep%
6821   \xlineref{\edindexlab\thelabidx}%
6822 }
6823 %

```

\thestartpageline These macros make up the page/line start/end number when the \edindex command is called in critical notes.
\theendpageline

```

6824 \newcommand{\thestartpageline}{%
6825   \l@dparsedstartpage%
6826   \pagelinesep%
6827   \l@dparsedstartline%
6828 }
6829 \newcommand{\theendpageline}{%
6830   \l@dparsedendpage%
6831   \pagelinesep%
6832   \l@dparsedendline%
6833 }
6834 %

```

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.

```

6835 \newcount\@edindex@fornote@
6836 %

```

`\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 `|`.

```

6837 \newcommand{\prepare@edindex@fornote}[1]{%
6838   \l@dp@rsefootspec#1|}%
6839   \advance\@edindex@fornote@\@ne%
6840 }
6841 %

```

`\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.

```

6842 \newcommand{\get@edindex@ledinnote@command}{%
6843   \ifxindy%
6844     \gdef\@ledinnote@command{%
6845       ledinnote\thelabidx%
6846     }%
6847     \ifxindyhyperref%
6848       \immediate\write\eledmac@xindy@out{%
6849         (define-attributes ("ledinnote\thelabidx"))^^J
6850         \space\space(markup-locref^^J
6851           \eledmacmarkuplocdepth^^J
6852           :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
6853         }{"^^J
6854           :close "}"^^J
6855           :attr "ledinnote\thelabidx"^^J
6856         )
6857       }%
6858     \else%
6859       \immediate\write\eledmac@xindy@out{%
6860         (define-attributes ("ledinnote\thelabidx"))^^J
6861         \space\space(markup-locref^^J
6862           \eledmacmarkuplocdepth^^J
6863           :open "\string\ledinnote{\@index@command}{"}^^J
6864           :close "}"^^J
6865           :attr "ledinnote\thelabidx"^^J
6866         )
6867       }%
6868     \fi%
6869 %

```

If we do not use `xindy` option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

6869 \else%
6870   \gdef\@ledinnote@command{%
6871     ledinnote[\edindexlab\thelabidx]{\@index@command}%

```

```

6872 }%
6873 \fi%
6874 }
6875 %

```

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 |.

```

6876 \def\get@index@command#1|#2+{%
6877   \gdef\@index@txt{#1}%
6878   \gdef\@index@command{#2}%
6879   \xdef\@index@parenthesis{%
6880     \IfBeginWith{\@index@command}{(}{%
6881       \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
6882       \global\let\@index@command\@index@command@%
6883       \xdef\@index@parenthesis{(%
6884     )}%
6885     \IfBeginWith{\@index@command}{)}{%
6886       \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
6887       \global\let\@index@command\@index@command@%
6888       \xdef\@index@parenthesis{)%
6889     }%
6890   }
6891   %

```

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`

```

6892 \newcommandx{\ledinnote}[3][1,usedefault]{%
6893   \ifboolexpr{%
6894     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
6895     or%
6896     bool {xindyhyperref@}%
6897   }%
6898   {%
6899     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
6900   }%
6901   {%
6902     \csuse{#2}{\ledinnotemark{#3}}%
6903   }%
6904 }%
6905 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage{#2}}}}%
6906 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
6907 %

```


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
6908 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
6909 \ifl@imakeidx%
6910 \ifnum\@edindex@fornote@>\z@%
6911 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6912 \get@edindex@ledinnote@command%
6913 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
6914 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%
6915 \else%
6916 \get@edindex@hyperref{#2}%
6917 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
6918 \fi%
6919 \else%
6920 \ifnum\@edindex@fornote@>\z@%
6921 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
6922 \get@edindex@ledinnote@command%
6923 \expandafter\protected@write\@indexfile{}%
6924 {\string\indexentry{\@index@txt|(\@ledinnote@command}{\thestartpageline}
}{%
6925 }%
6926 \expandafter\protected@write\@indexfile{}%
6927 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
}{%
6928 }%
6929 \else%
6930 \protected@write\@indexfile{}%
6931 {\string\indexentry{#2}{\thepageline}
}{%
6932 }%
6933 \fi%
6934 \fi%
6935 \endgroup
6936 \@esphack%
6937 }
6938 %

```

Need to add the definition of \edindex to \makeindex, and initialise \edindex to do nothing.

```

6939 \pretocmd{\makeindex}{%
6940   \def\edindex{%
6941     \ifboolexpr{bool{numbering} or bool{numberingR} or bool{
6942       l@dprintingpages} or bool{l@dprintingcolumns}}{%
6943       \@bsphack%
6944       \doedindexlabel%
6945       \begingroup%
6946       \@sanitize%
6947       \wredindex%
6948     }%
6949     {\%
6950       \led@warn@edinde@outsidenumbering%
6951       \index%
6952     }%
6953   }%
6954   {}%
6955   {\led@error@fail@patch@makeindex}%
6956   \newcommand{\edindex}[1]{\@bsphack\@esphack}
6957   %

```

XXVI.7 Hyperlink

\hyperlinkformat \hyperlinkformat command is to be used to have both a internal hyperlink and a format, when indexing.

```

6958 \newcommand{\hyperlinkformat}[3]{%
6959   \ifstrempy{#1}%
6960   {\hyperlink{#2}{#3}}%
6961   {\csuse{#1}{\hyperlink{#2}{#3}}}%
6962   }%
6963   %

```

\hyperlinkR \hyperlinkR command is to be used to create a internal hyperlink and \ledRflag, when indexing.

```

6964 \newcommand{\hyperlinkR}[2]{%
6965   \hyperlink{#1}{#2\@Rlineflag}%
6966   }%
6967   %
6968   %

```

\hyperlinkformatR \hyperlinkformatR command is to be used to create a internal hyperlink, a format and a \@Rlineflag, when indexing.

```

6969 \newcommand{\hyperlinkformatR}[3]{%
6970   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
6971   }%
6972   %
6973   %

```

`\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`).

```
6974 \newcommand{\get@edindex@hyperref}[1]{%
6975 %
```

We have to disable temporary spaces to work through a xstring bug (or feature?)

```
6976 \edef\temp@{%
6977 \catcode\ =9 %space need for catcode
6978 \detokenize{#1}%For active character in unicode
6979 \catcode\ =10 % space need for catcode
6980 }%
6981 %
```

Now, we define `\@edindex@hyperref` if the hyperindex of `hyperref` is enabled.

```
6982 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
6983 \IfSubStr{\temp@}{|}%
6984 {\get@index@command#1+%
6985 \ifledRcol%
6986 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6987 hyperlinkformatR{\@index@command}%
6988 {\edindexlab\thelabidx}}%
6989 \else%
6990 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
6991 hyperlinkformat{\@index@command}%
6992 {\edindexlab\thelabidx}}%
6993 \fi%
6994 }%
6995 {\get@index@command#1|+%
6996 \ifledRcol%
6997 \gdef\@edindex@hyperref{|\hyperlinkR{\edindexlab\thelabidx}}%
6998 \else%
6999 \gdef\@edindex@hyperref{|\hyperlink{\edindexlab\thelabidx}}%
7000 \fi%
7001 }%
7002 }%
7003 %
```

```
7004 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
7005 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
7006 % \begin{macrocode}
7007 {\ifxindyhyperref%
7008 \IfSubStr{\temp@}{|}%
7009 {\get@index@command#1+%
7010 {\get@index@command#1|+%
7011 \gdef\@edindex@hyperref{|\eledmac\thelabidx}%
7012 %
```

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`.

```

7013 \IfStrEq{\@index@parenthesis}{(}{%
7014   {%
7015   \csxdef{xindy@parenthesis@\@index@txt}{\thelabidx}%
7016   \gdef\@edindex@hyperref{|\@edmac\thelabidx}%
7017   }%
7018   {}%
7019 %

```

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`.

```

7020 \IfStrEq{\@index@parenthesis}{)}{%
7021   {%
7022   \xdef\@edindex@hyperref{|\@edmac\cuse{xindy@parenthesis@\@index@txt}}%
7023   \global\csundef{xindy@parenthesis@\@index@txt}%
7024   }%
7025 %

```

Write in the `.xdy` file the attributes of the location.

```

7026   {%
7027   \immediate\write\eledmac@xindy@out{%
7028     (define-attributes ("eledmac\thelabidx"))^^J
7029     \space\space(markup-locref^^J
7030     \eledmacmarkuplocdepth^^J
7031     :open "\string\hyperlink%
7032       \ifledRcol R\fi%
7033       {\edindexlab\thelabidx}%
7034       {\ifdefempty{\@index@command}%
7035       }%
7036       {\@backslashchar\@index@command}%
7037       {"^^J
7038     :close "}")^^J
7039     :attr "eledmac\thelabidx"^^J
7040   )
7041   }%
7042   }%
7043 %

```

And now, in any other case.

```

7044 \else%
7045   \gdef\@index@txt{#1}%
7046   \gdef\@edindex@hyperref{}%
7047   \fi%
7048   }%
7049 }
7050 %

```

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.

```

7051 \newcommand{\led@set@index@fornote}[1]{%
7052   \ifbool{indtl@innote}%
7053     {\let\index\nindex}%
7054     {}%
7055   \ifbool{indtl@notenumber}%
7056     {%
7057     \renewcommand{\index}[2][\indtl@jobname]{%
7058       \orig@@index[##1]{%
7059         ##2|innotenumber{\this@footnoteX@reading}%
7060       }%
7061     }%
7062     }%
7063     {}%
7064   }%
7065   %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

7066 \newcommand{\led@reinit@index@fornote}{%
7067   \ifbool{indtl@innote}%
7068     {\let\index\orig@@index}%
7069     {}%
7070   \ifbool{indtl@notenumber}%
7071     {\let\index\orig@@index}%
7072     {}%
7073   }%
7074   %

```

XXVII Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

7075 \AtBeginDocument{%
7076   \ifpackageloaded{glossaries}{%
7077     %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

7078   \renewcommand{\do}[1]{%
7079     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
7080       \doedindexlabel%
7081       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%

```

```

7082 }%
7083 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,3,usedefault]{%
7084 \doedindexlabel%
7085 \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
7086 }%
7087 }%
7088 \docsvlist{%
7089 gls,%
7090 GLs,%
7091 GLS,%
7092 glspl,%
7093 Glspl,%
7094 GLSpl,%
7095 glstext,%
7096 Glstext,%
7097 GLStext,%
7098 Glsfirst,%
7099 GLSfirst,%
7100 glsplural,%
7101 Glsplural,%
7102 GLSplural,%
7103 glsfirstplural,%
7104 Glsfirstplural,%
7105 GLSfirstplural,%
7106 glsname,%
7107 Glsname,%
7108 GLSname,%
7109 glssymbol,%
7110 Glssymbol,%
7111 GLSsymbol,%
7112 glsdesc,%
7113 Glsdesc,%
7114 GLSdesc,%
7115 glsuseri,%
7116 Glsuseri,%
7117 GLSuseri,%
7118 glsuserii,%
7119 Glsuserii,%
7120 GLSuserii,%
7121 glsuseriii,%
7122 Glsuseriii,%
7123 GLSuseriii,%
7124 glsuseriv,%
7125 Glsuseriv,%
7126 GLSuseriv,%
7127 glsuserv,%
7128 Glsuserv,%
7129 GLSuserv,%
7130 glsuservi,%

```

```

7131     Glsuservi,%
7132     GLSuservi%
7133     }%
7134 %

```

Then those which arguments are $[\langle options \rangle] \{ \langle label \rangle \} \{ \langle link text \rangle \}$.

```

7135     \renewcommand{\do}[1]{%
7136         \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,
usedefault]{%
7137             \doedindexlabel%
7138             \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
7139         }%
7140         \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,usedefault]{%
7141             \doedindexlabel%
7142             \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
7143         }%
7144     }%
7145     \docsvlist{glsdisp,glslink}%
7146 %

```

Then those which arguments are $[\langle options \rangle] \{ \langle label \rangle \}$.

```

7147     \renewcommand{\do}[1]{%
7148         \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[2][1,
usedefault]{%
7149             \doedindexlabel%
7150             \csname#1\endcsname[counter=pageline,##1]{##2}%
7151         }%
7152         \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[2][1,usedefault]{%
7153             \doedindexlabel%
7154             \csname#1\endcsname*[counter=pageline,##1]{##2}%
7155         }%
7156     }%
7157     \docsvlist{glsadd}%
7158     {}%
7159 }%
7160 %

```

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.

```
7161 \def\@hangingsymbol{}
7162 \newcommand*{\sethangingsymbol}[1]{%
7163   \gdef\@hangingsymbol{#1}%
7164 }%
7165 \newif\ifinstanza
7166 %
```

`\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.

```
7167 \newif\ifinserthangingsymbol
7168 \newcommand{\inserthangingsymbol}{%
7169   \ifinserthangingsymbol%
7170     \ifinstanza%
7171       \@hangingsymbol%
7172     \fi%
7173 \fi%
7174 }
7175 %
```

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.

```
7176 \newcommand*{\ampersand}{\char`\&}
7177
7178 %
```

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.

```
7179 \chardef\body=\catcode`\@
7180 \catcode`\@=11
7181 \chardef\next=\catcode`\&
```



```

7182 \catcode`\&=\active
7183
7184 %

```

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.

```

7185 \newcount\stanza@count
7186 \newlength{\stanzaindentbase}
7187 \setlength{\stanzaindentbase}{20pt}
7188
7189 %

```

`\strip@szacnt`
`\setstanzavalues`

The indentations of stanza lines are non-negative integer multiples of the unit called `\stanzaindentbase`. To make it easier for the user to specify these numbers, some list macros are defined. These take numerical values in a list separated by commas and assign the values to special control sequences using `\mathchardef`. Though this does limit the range from 0 to 32767, it should suffice for most applications, including *penalties*, which will be discussed below.

```

7190 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
7191 \newcommand*\setstanzavalues}[2]{\def\@tempa{#2,,|}%
7192   \stanza@count\z@
7193   \def\next{\expandafter\strip@szacnt\@tempa
7194     \ifx\@tempb\empty\let\next\relax\else
7195       \expandafter\mathchardef\csname #1@\number\stanza@count
7196       \@endcsname\@tempb\relax
7197       \advance\stanza@count\@ne\fi\next}%
7198   \next}
7199
7200 %

```

`\setstanzaindents`
`\setstanzapenalties`

In the original edmac, `\setstanzavalues{sza}{...}` had to be called to set the indents, and similarly `\setstanzavalues{szp}{...}` to set the penalties. `\setstanzaindents` and `\setstanzapenalties` macros are a convenience to give the user one less thing to worry about (misspelling the first argument).

```

7201 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
7202 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
7203 %
7204 %

```

`\managestanza@modulo`

Since version 0.13, the `stanzaindentrepetition` counter can be used when the indentation is repeated every *n* verses. The `\managestanza@modulo` is a command which modifies the counter `stanza@modulo`. The command adds 1 to `stanza@modulo`, but if

stanza@modulo is equal to the stanzaindentrepetition counter, the command restarts it.

```

7205 \newcounter{stanzaindentrepetition}
7206 \newcount\stanza@modulo
7207
7208 \newcommand*{\managestanza@modulo}[0]{%
7209     \advance\stanza@modulo\@ne%
7210     \ifnum\stanza@modulo>\value{stanzaindentrepetition}%
7211         \stanza@modulo\@ne%
7212     \fi%
7213 }
7214 %

```

\stanzaindent The macro \stanzaindent, when called at the beginning of a verse, changes the indentation normally defined for this verse by \setstanzaindent. The starred version skips the current verse for the repetition of stanza indent.

```

7215 \newcommand{\stanzaindent}[1]{%
7216     \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
7217     \ignorespaces%
7218 }%
7219 \WithSuffix\newcommand\stanzaindent*[1]{%
7220     \stanzaindent{#1}%
7221     \global\advance\stanza@modulo-\@ne%
7222     \ifnum\stanza@modulo=0%
7223         \global\stanza@modulo=\value{stanzaindentrepetition}%
7224     \fi%
7225     \ignorespaces%
7226 }%
7227 %

```

XXVIII.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza28 \newcounter{stanza}
7229 \renewcommand{\thestanza}{%
7230     \textbf{\arabic{stanza}}%
7231 }
7232 %

```

\ifnumberstanza Then, macro to activate automatically numbering of stanza.

```

7233 \newif\ifnumberstanza%
7234 %

```

\@insertstanzanumber Now, macro called at the first line of verse of a stanza.

```

7235 \newcommand{\@insertstanzanumber}[0]{%
7236   \ifnumberstanza%
7237   \ifl@dpairing%
7238     \ifledRcol%
7239       \stanzanumwrapper{\thestanzaR}%
7240     \else%
7241       \stanzanumwrapper{\thestanzaL}%
7242     \fi%
7243   \else%
7244     \stanzanumwrapper{\thestanza}%
7245   \fi%
7246   \setline{1}%
7247 \fi%
7248 }%
7249 %

```

\@advancestanzanumber Also a command to advance the counter of stanza.

```

7250 \newcommand{\@advancestanzanumber}[0]{%
7251   \ifnumberstanza%
7252   \ifl@dpairing%
7253     \ifledRcol%
7254       \addtocounter{stanzaR}{1}%
7255     \else%
7256       \addtocounter{stanzaL}{1}%
7257     \fi%
7258   \else%
7259     \addtocounter{stanza}{1}%
7260   \fi%
7261 \fi%
7262 }%
7263 %

```

\stanzanumwrapper And finally, the wrapper for stanza number

```

7264 \newcommand{\stanzanumwrapper}[1]{%
7265   \flagstanza{#1}%
7266 }%
7267 %

```

XXVIII.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

7268 \newcommand{\printstanza}[0]{%
7269   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
7270     l@dprintingcolumns}}{%
7271     \ifledRcol%

```

```

7272     \else%
7273         \thestanzaL%
7274     \fi%
7275 }{%
7276     \thestanza%
7277 }%
7278 }
7279 %

```

XXVIII.7 Main work

`\stanza@line` Now we arrive at the main works. `\stanza@line` sets the indentation for the line and starts a numbered paragraph—each line is treated as a paragraph. `\stanza@hang` sets the hanging indentation to be used if the stanza line requires more than one print line.

If it is known that each stanza line will fit on one print line, it is advisable to set the hanging indentation to zero. `\sza@penalty` places the specified penalty following each stanza line. By default, this facility is turned off so that no penalty is included. However, the user may initiate these penalties to indicate good and bad places in the stanza for page breaking.

```

7280 \newcommandx{\stanza@line}[1][1]{
7281     \ifnum\value{stanzaindentrepetition}=0
7282         \ifcsdef{sza@\number\stanza@count @}%
7283             {%
7284                 \parindent=\csname sza@\number\stanza@count @\endcsname\
7285                 stanzaindentbase%
7286             }{%
7287                 \led@err@StanzaIndentNotDefined%
7288             }%
7289         \else
7290             \ifcsdef{sza@\number\stanza@modulo @}{%
7291                 \parindent=\csname sza@\number\stanza@modulo @\endcsname\
7292                 stanzaindentbase%
7293                 \managestanza@modulo%
7294             }{%
7295                 \led@err@StanzaIndentNotDefined%
7296             }%
7297         \fi
7298     \pstart[#1]\stanza@hang\ignorespaces}
7299 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
7300     \hangindent\expandafter
7301     \noexpand\csname sza@0@\endcsname\stanzaindentbase
7302     \hangafter\@ne}
7303 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
7304     \ifnum\count@>@M\advance\count@-\@M\penalty-\else
7305     \penalty\fi\count@}
7306 %

```

`\@startstanza` Now we have the components of the `\stanza` macro, which appears at the start of a
`\stanza` group of lines. This macro initializes the count and checks to see if hanging indentation
`\@stopstanza` and penalties are to be included. Hanging indentation suspends the line count, so that
`\AtEveryStopStanza` the enumeration is by verse line rather than by print line. If the print line count is
`\AtEveryStanza` desired, invoke `\let\startlock\relax` and do the same for `\endlock`. Here and
`\newverse` above we have used `\xdef` to make the stored macros take up a bit less space, but it also
 makes them more obscure to the reader. Lines of the stanza are delimited by ampersands
 &. The last line of the stanza must end with `\&`.

```

7306 \xdef\@startstanza[#1]{%
7307   \noexpand\instanzatrue\expandafter
7308   \begingroup%
7309   \catcode`\noexpand\&\active%
7310   \global\stanza@count\@ne\stanza@modulo\@ne
7311   \noexpand\ifnum\expandafter\noexpand
7312   \csname sz@00\endcsname=z@\let\noexpand\stanza@hang\relax
7313   \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
7314   \@M\righskipz@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
7315   \expandafter\noexpand\csname szp@00\endcsname=z@
7316   \let\noexpand\sza@penalty\relax\noexpand\fi%
7317   \def\noexpand&{%
7318     \noexpand\newverse[] []}%
7319   \def\noexpand\&\noexpand\@stopstanza}%
7320   \noexpand\@advancestanza@number%
7321   \noexpand\stanza@line[#1]%
7322   \noexpand\@insertstanza@number%
7323   \let\par\relax\ignorespaces%No paragraph in verses
7324 }
7325
7326 \newcommandx{\stanza}[1][1,usedefault]{%
7327   \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}} and test{\ifstrempy
7328   {#1}}}%
7329   {\@startstanza[\at@every@stanza]}%
7330   {\@startstanza[#1]}%
7331 }%
7332
7332 \newcommandx{\@stopstanza}[1][1,usedefault]{%
7333   \unskip%
7334   \endlock%
7335   \ifboolexpr{not test{\ifdefvoid{\at@every@stop@stanza}} and test{\
7336   ifstrempy{#1}}}%
7337   {\pend[\at@every@stop@stanza]}%
7338   {\pend[#1]}%
7339   \endgroup%
7340   \instanzafalse%
7341 }
7342
7342 \newcommand{\AtEveryStopStanza}[1]{%
7343   \ifstrempy{#1}%
7344   {\xdef\at@every@stop@stanza{}}%

```

```

7345 {\gdef\at@every@stop@stanza{#1}}%
7346 }%
7347 \def\at@every@stop@stanza{}%
7348
7349 \newcommand{\AtEveryStanza}[1]{%
7350   \ifstrempy{#1}%
7351     {\xdef\at@every@stanza{}}%
7352     {\gdef\at@every@stanza{#1}}%
7353 }%
7354 \def\at@every@stanza{}%
7355
7356
7357 \newcommand*{\newverse}[2][1,2,usedefault]{%
7358   \unskip%
7359   \endlock\pend[#1]\sza@penalty\global%
7360   \advance\stanza@count\@ne\stanza@line[#2]%
7361 }
7362
7363 %

```

\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`.

```

7364 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
7365   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
7366
7367 %

```

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.

```

7368 \catcode`\&=\next
7369 \catcode`\@=\body
7370 \setstanzavalues{szp}{0}
7371
7372 %

```

XXIX Apparatus of Manuscripts

XXIX.1 User level macro

\msdata The user level `\msdata` command only writes the manuscripts data in numbered auxiliary file. There is two associated etoolbox counter.

\msdata@c

\msdata@cR

```

7373 \def\msdata@c{}%
7374 \def\msdata@cR{}%
7375 \newcommand{\msdata}[1]{%
7376   \leavevmode%
7377   \unless\ifstopmsdata@inserted@%
7378     \stopmsdata%
7379     \led@warning@msdatawithoutstop%
7380   \fi%
7381   \global\stopmsdata@inserted@false%
7382   \unless\ifledRcol%
7383     \numgdef{\msdata@c}{\msdata@c+1}%
7384     \ifdef{\hypertarget}{%
7385       \edlabel{\msdata@c:start:msdata}%
7386     }{}%
7387     \protected@write\linenum@out{}{%
7388       \string\@msd{#1}%
7389     }%
7390   \else%
7391     \numgdef{\msdata@cR}{\msdata@cR+1}%
7392     \ifdef{\hypertarget}{%
7393       \edlabel{\msdata@cR:start:msdata}%
7394     }{}%
7395     \protected@write\linenum@outR{}{%
7396       \string\@msd{#1}%
7397     }%
7398   \fi%
7399 }%
7400 %

```

\stopmsdata The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

7401 \newcommand{\stopmsdata}[0]{%
7402   \leavevmode%
7403   \unless\ifledRcol%
7404     \protected@write\linenum@out{}{%
7405       \string\@stopmsd%
7406     }%
7407     \ifdef{\hypertarget}{%
7408       \edlabel{\msdata@c:end:msdata}%
7409     }{}%
7410   \else%
7411     \protected@write\linenum@outR{}{%
7412       \string\@stopmsd%
7413     }%
7414     \ifdef{\hypertarget}{%
7415       \edlabel{\msdata@cR:end:msdata}%
7416     }{}%
7417   \fi%
7418   \global\stopmsdata@inserted@true%

```

```
7419 }%
7420 %
```

\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`

```
7421 \newif\ifstopmsdata@inserted@%
7422 %
```

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?

```
7423 \newcommand{\setmsdataseries}[1]{%
7424   \gdef\@msdata@series{#1}%
7425 }%
7426 \def\@msdata@series{A}%
7427 %
```

\setmsdataposition The label for the manuscripts data.

```
7428 \def\ms@data@position{msdata-regular}%
7429 \newcommand{\setmsdataposition}[1]{%
7430   \gdef\ms@data@position{#1}%
7431 }%
7432 %
```

\setmsdatalabel The label for the manuscripts data.

```
7433 \def\ms@data@label{Ms.}%
7434 \newcommand{\setmsdatalabel}[1]{%
7435   \gdef\ms@data@label{#1}%
7436 }%
7437 %
```

XXIX.3 Counters and lists

\@msd@c `\@msd@c` is a counter incremented at each `\@msd` read in auxiliary file.

```
7438 \numdef{\@msd@c}{0}
7439 \numdef{\@msd@cR}{0}
7440 %
```

\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.


```

7441 \numdef{\add@msd@c}{0}%
7442 \numdef{\add@msd@cR}{0}%
7443 %

```

\@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.

```

7444 \def\@msdata@list{}%
7445 %

```

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.

```

7446 \newcommand{\@msd}[1]{%
7447   \unless\ifledRcol%
7448     \global\numdef{\@msd@c}{\@msd@c+\@ne}%
7449     \csgdef{\@msdata@\@msd@c @data}{#1}%
7450     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
7451     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
7452     \xright@appenditem{\the\absline@num}\to\actionlines@list%
7453     \xright@appenditem{-1010}\to\actions@list%
7454   \else%
7455     \global\numdef{\@msd@cR}{\@msd@cR+\@ne}%
7456     \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
7457     \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
7458     \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
7459     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
7460     \xright@appenditem{-1010}\to\actions@listR%
7461   \fi%
7462 }%
7463 %

```

\@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.

```

7464 \newcommand{\@stopmsd}[0]{%
7465   \unless\ifledRcol%
7466     \ifcsundef{\@msdata@\@msd@c @stoplinenumber}{%
7467       \csxdef{\@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
7468       \csxdef{\@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
7469     }%
7470   \else%
7471     \ifcsundef{\@msdata@\@msd@cR @stoplinenumberR}{%
7472       \csxdef{\@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%

```

```

7473 \csxdef{@msdata@ \@msd@cR @stoplinenumberR}{\the\line@numR}%
7474 }%
7475 {}%
7476 \fi%
7477 }%
7478 %

```

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.

```

7479 \newcommand{\add@msdata}{%
7480 \bgroup%
7481 \normalfont%
7482 \unless\ifledRcol@%
7483 \numgdef{\add@msd@c}{\add@msd@c+\@ne}%
7484 \ifcsdef{@msdata@ \@add@msd@c @data}{%
7485 \letcs{\@data}{@msdata@ \@add@msd@c @data}%
7486 \edef\l@d@nums{%
7487 000| % Start page = we don't print it
7488 \csuse{@msdata@ \@add@msd@c @linenumber}| % Start line number
7489 000| % Start subline number, for now, not used
7490 000| % End page number, we don't print it
7491 \ifnumless{\csuse{@msdata@ \@add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@ \the\page@num}}}%
7492 {\csuse{@msdata@ \@add@msd@c @stoplinenumber}}}% End line number if
in the same page
7493 {\csuse{@lastline@forpage@ \the\page@num}}}% Otherwiser, last
number of the page
7494 |%
7495 000| % End sub line number, for now, not used
7496 \edfont@info % Font
7497 }%
7498 \@msd@options@fullpagefalse%
7499 \if@firstlineofpage % Try if the data are for the full page. If yes
, will add options to the list.
7500 \unless\if@msdata@insertedfrompreviouspage%
7501 \ifnumless{\csuse{@lastabsline@forpage@ \the\page@num}}{\csuse{
@msdata@ \@add@msd@c @stopabslinenumber}+\@ne}%
7502 {%
7503 \numdef{\@tmp}{\add@msd@c+\@ne}%
7504 \ifcsdef{@msdata@ \@tmp @abslinenumber}%
7505 {\ifnumequal{\csuse{@msdata@ \@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@ \the\page@num}}}%
7506 }%
7507 {\@msd@options@fullpagetrue}%

```

```

7508         }%
7509         {\@msd@options@fullpagetrue}%
7510     }%
7511     {}%
7512     \fi%
7513 \fi%
7514 \listxadd{\@msdata@list}{%
7515     \@msd@options@iffullpage%
7516     \ifluatex%
7517         \csxdef{footnote@luatextextdir}{\the\textdir}%
7518         \csxdef{footnote@luatexpardir}{\the\pardir}%
7519     \fi%
7520     \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
7521     \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
7522     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
7523     \reset@msd@options@iffullpage%
7524 }%
7525 }%
7526 {}%
7527 \else%
7528     \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
7529     \ifcsdef{@msdata@\add@msd@cR @dataR}{%
7530         \letcs{@data}{@msdata@\add@msd@cR @dataR}%
7531         \edef\l@d@nums{%
7532             000| % Start page = we don't print it
7533             \csuse{@msdata@\add@msd@cR @linenumberR}| % Start line number
7534             000| % Start subline number, for now, not used
7535             000| % End page number, we don't print it
7536             \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\@
csuse{@lastline@forpageR@the\page@numR}}}%
7537             {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
7538             {\csuse{@lastline@forpageR@the\page@numR}}}%Otherwiser, last
number of the page
7539             |%
7540             000| % End sub line number, for now, not used
7541             \edefont@info%Font
7542         }%
7543         \@msd@options@fullpagefalse%
7544         \if@firstlineofpageR%
7545             \unless\if@msdata@insertedfrompreviouspage%
7546                 \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\@
csuse{@msdata@\add@msd@c @stopabslinenumberR}+\@ne}%
7547                 {%
7548                     \numdef{\@tmp}{\add@msd@cR+\@ne}%
7549                     \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7550                         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
7551                         {}%

```

```

7552         {\@msd@options@fullpagetrue}%
7553     }%
7554     {\@msd@options@fullpagetrue}%
7555 }%
7556 {}%
7557 \fi%
7558 \fi%
7559 \listxadd{\@msdata@list}{%
7560     \@msd@options@iffullpage%
7561     \ifluatex%
7562         \csxdef{footnote@luatextextdir}{\the\textdir}%
7563         \csxdef{footnote@luatexpardir}{\the\pardir}%
7564     \fi%
7565     \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
7566     \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
7567     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
7568     \reset@msd@options@iffullpage%
7569 }%
7570 }%
7571 {}%
7572 \fi%
7573 \egroup%
7574 }%
7575 %

```

`\if@msdata@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.

```

7576 \newif\if@msdata@insertedfrompreviouspage%
7577 %

```

`\add@msdata@firstlineofpage` `\add@msdata@firstlineofpage` is called at the first line of every page. It inserts manuscript data which start on one of the previous pages and continue on this page.

```

7578 \newcommand{\add@msdata@firstlineofpage}{%
7579     \bgroup%
7580     \normalfont%
7581     \unless\ifledRcol@%
7582         \ifcsdef{@msdata@\add@msd@c @data}{%
7583             \ifnumless{\the\absline@num-\@ne}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}%
7584         }%
7585         \global\@msdata@insertedfrompreviouspagetrue%
7586         \letcs{@data}{@msdata@\add@msd@c @data}%
7587         \edef\l@d@nums{%
7588             000|% Start page = we don't print it
7589             \numexpr\the\line@num+\@ne\relax|% Start line number = first line
of the page. As \add@msdata@firstlineofpage is called before line number
has been incremented, we increment it for printing

```

```

7590         000|% Start subline number, for now, not used
7591         000|% End page number, we don't print it
7592         \ifnumless{\csuse{@msdata@add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}%
7593         {\csuse{@msdata@add@msd@c @stoplinenumber}}%End line number if
in the same page
7594         {\csuse{@lastline@forpage@the\page@num}}%Otherwise, last
number of the page
7595         |%
7596         000|% End sub line number, for now, not used
7597         \edfont@info%Font
7598         }%
7599         \@msd@options@fullpagefalse%
7600         \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
7601         {%
7602         \numdef{\@tmp}{\add@msd@c+\@ne}%
7603         \ifcsdef{@msdata@\@tmp @abslinenumber}%
7604         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}}%
7605         }%
7606         {\@msd@options@fullpagetrue}%
7607         }%
7608         {\@msd@options@fullpagetrue}%
7609         }%
7610         {}%
7611         \listxadd{\@msdata@list}{%
7612         \@msd@options@iffullpage%
7613         \ifluatex%
7614         \csxdef{footnote@luatextextdir}{\the\textdir}%
7615         \csxdef{footnote@luatexpardir}{\the\pardir}%
7616         \fi%
7617         \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
7618         \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
7619         \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
7620         \reset@msd@options@iffullpage%
7621         }%
7622         }%
7623         {\global\@msdata@insertedfrompreviouspagefalse}%
7624         }{}%
7625         \else%
7626         \ifcsdef{@msdata@add@msd@cR @dataR}{%
7627         \ifnumless{\the\absline@numR-\@ne}{\csuse{@msdata@add@msd@cR
@stopabslinenumberR}}%
7628         {%
7629         \global\@msdata@insertedfrompreviouspagetrue%
7630         \letcs{\@data}{\@msdata@add@msd@cR @dataR}%
7631         \edef\l@d@nums{%

```

```

7632      000| % Start page = we don't print it
7633      \numexpr\the\line@numR+\@one\relax| % Start line number = first
line of the page. As \add@msdata@firstlineofpage is called before line
number has been incremented, we increment it for printing
7634      000| % Start subline number, for now, not used
7635      000| % End page number, we don't print it
7636      \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@\the\page@numR}}}%
7637      {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}}%End line number
if in the same page
7638      {\csuse{@lastline@forpageR@\the\page@numR}}}%Otherwise, last
number of the page
7639      |%
7640      000| % End sub line number, for now, not used
7641      \edfont@info%Font
7642      }%
7643      \@msd@options@fullpagefalse%
7644      \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@one}%
7645      {%
7646      \numdef{\@tmp}{\add@msd@cR+\@one}%
7647      \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7648      {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}}%
7649      }%
7650      {\@msd@options@fullpagetrue}%
7651      }%
7652      {\@msd@options@fullpagetrue}%
7653      }%
7654      {}%
7655      \listxadd{\@msdata@list}{%
7656      \@msd@options@iffullpage%
7657      \ifluatex%
7658      \csxdef{footnote@luatextextdir}{\the\textdir}%
7659      \csxdef{footnote@luatexpardir}{\the\pardir}%
7660      \fi%
7661      \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
7662      \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
7663      \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\msdata@label}{\expandonce\data}}%
7664      \reset@msd@options@iffullpage%
7665      }%
7666      }%
7667      {\global\@msdata@insertedfrompreviouspagefalse}%
7668      }{}%
7669      \fi%
7670      \egroup%
7671      }%
7672      %

```

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 \newcommand{\insert@msdata}{%
7674 \def\do##1{##1}%
7675 \dolistloop{\@msdata@list}%
7676 \global\let\@msdata@list\relax%
7677 }%
7678 %
```

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.

```
7679 \newif\if@msd@options@fullpage%
7680 \newcommand{\@msd@options@iffullpage}[0]{%
7681 \if@msd@options@fullpage%
7682 \noexpand\toggletrue{nonum@}%
7683 \ifdefvoid{\ms@data@label}%
7684 {\noexpand\toggletrue{nosep@}}%
7685 }%
7686 \fi%
7687 }%
7688 \newcommand{\reset@msd@options@iffullpage}[0]{%
7689 \noexpand\togglefalse{nonum@}%
7690 \noexpand\togglefalse{nosep@}%
7691 }%
7692 %
```

XXX Arrays and tables

XXX.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread ‘`eq and amstex`’, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsgen` package.

```
7693 \newtoks\@emptytoks
7694
7695 %
```

The rest is from amsmath.

`\l@denbody` A token register to contain the body.

```
7696 \newtoks\l@denbody
7697
7698 %
```

`\addtol@denbody` `\addtol@denbody{arg}` adds arg to the token register `\l@denbody`.

```
7699 \newcommand{\addtol@denbody}[1]{%
7700   \global\l@denbody\expandafter\the\l@denbody#1}
7701
7702 %
```

`\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 `env#1{...}` as a macro that processes #1, then the environment form, `\begin{env}` would call `\l@dcollect@body\env`.

```
7703 \newcommand{\l@dcollect@body}[1]{%
7704   \l@denbody{\expandafter#1\expandafter\the\l@denbody}}%
7705   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}%
7706   \l@denbody\emptytoks \def\l@dbegin@stack{b}%
7707   \begingroup
7708     \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
7709     \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
7710     \processl@denbody%
7711   }%
7712
7713 %
```

`\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.

```
7714 \def\l@dpush@begins#1\begin#2{%
7715   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
7716
7717 %
```

`\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.


```

7718 \def\l@dcollect@@body#1\end#2{%
7719   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
7720     \expandafter\@gobble\l@dbegin@stack}%
7721   \ifx\@empty\l@dbegin@stack
7722     \endgroup
7723     \@checkend{#2}%
7724     \addtol@denvbody{#1}%
7725   \else
7726     \addtol@denvbody{#1\end{#2}}%
7727   \fi
7728   \processl@denvbody % A little tricky! Note the grouping
7729 }
7730
7731 %

```

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.

```

7732 \newcommand*{\l@dtabnoexpands}{%
7733   \let\rtab=0%
7734   \let\ctab=0%
7735   \let\ltab=0%
7736   \let\rtabtext=0%
7737   \let\ltabtext=0%
7738   \let\ctabtext=0%
7739   \let\edbeforetab=0%
7740   \let\edaftertab=0%
7741   \let\edatleft=0%
7742   \let\edatright=0%
7743   \let\edvertline=0%
7744   \let\edvertdots=0%
7745   \let\edrowfill=0%
7746 }
7747
7748 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in `edtabularx` and `edarrayx` environments.

```

7749 \newcommand{\disable@familiarnotes}{%
7750   \unless\ifnofamiliar%
7751     \def\do##1{%
7752       \csletcs{footnote@##1}{footnote##1}%

```

```

7753     \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
7754         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
7755         \csuse{@footnotemark##1}%
7756     }%
7757 }%
7758 \dolistloop{\@series}%
7759 \fi%
7760 }%
7761 \newcommand{\restore@familiarnotes}{%
7762     \unless\ifnofamiliar%
7763     \def\do##1{%
7764         \csletcs{footnote##1}{footnote@@##1}%
7765     }%
7766     \dolistloop{\@series}%
7767 \fi%
7768 }%
7769 %
7770 %

```

`\disable@sidenotes` The same, for side notes.

```

\restore@sidenotes
7771 \newcommand{\disable@sidenotes}{%
7772     \let\@@ledrightnote\ledrightnote%
7773     \let\@@ledleftnote\ledleftnote%
7774     \let\@@ledsidenote\ledsidenote%
7775     \let\ledrightnote@gobble%
7776     \let\ledleftnote@gobble%
7777     \let\ledsidenote@gobble%
7778 }%
7779 \newcommand{\restore@sidenotes}{%
7780     \let\ledrightnote\@@ledrightnote%
7781     \let\ledleftnote\@@ledleftnote%
7782     \let\ledsidenote\@@ledsidenote%
7783 }%
7784 %

```

`\disable@notes` Disable/restore side and familiar notes.

```

\restore@notes
7785 \newcommand{\disable@notes}{%
7786     \disable@sidenotes%
7787     \disable@familiarnotes%
7788 }%
7789 \newcommand{\restore@notes}{%
7790     \restore@sidenotes%
7791     \restore@familiarnotes%
7792 }%
7793 %

```

`\EDTEXT` We need to be able to modify the `\edtext` macros and also restore their original definitions.

`\xedtext`

```

7794 \let\EDTEXT=\edtext
7795 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
7796 %

```

\EDLABEL We need to be able to modify and restore the `\edlabel` macro.

```

\xedlabel
7797 \let\EDLABEL=\edlabel
7798 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}
7799 %

```

\xedindex Macros supporting modification and restoration of `\edindex`.

```

\nulledindex
7800 \AtBeginDocument{\let\xedindex\edindex}%
7801 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
7802
7803 %

```

\@line@num Macro supporting restoration of `\linenum`.

```

7804 \let\@line@num=\linenum
7805 %

```

\l@dgobblearg `\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```

7806 \newcommand*{\l@dgobbleoptarg}[2][\relax]%
7807
7808 %

```

\Relax `\let\Relax=\relax`

\NEXT `\let\NEXT=\next`

```

7811
7812 %

```

\l@dmodforedtext Modify and restore various macros for when `\edtext` is used.

\l@drestoreforedtext

```

7813 \newcommand{\l@dmodforedtext}{%
7814   \let\edtext\relax
7815   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
7816   \dolistloop{\@series}%
7817   \let\edindex\nulledindex
7818   \let\linenum@gobble}
7819 \newcommand{\l@drestoreforedtext}{%
7820   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
7821   \dolistloop{\@series}%
7822   \let\edindex\xedindex}
7823 %

```

`\l@dnnullfills` Nullify and restore some column fillers, etc.

```
\l@drestorefills
7824 \newcommand{\l@dnnullfills}{%
7825   \def\edlabel##1{%
7826     \def\edrowfill##1##2##3{%
7827   }
7828 \newcommand{\l@drestorefills}{%
7829   \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
7830 }
7831
7832 %
```

`\letsforverteilen` Gathers some lets and other code that is common to the **verteilen** macros.

```
7833 \newcommand{\letsforverteilen}{%
7834   \let\edtext\xedtext
7835   \let\edindex\xedindex
7836   \def\do##1{\global\csletcs{##1footnote}{##1@@footnote}}
7837   \dolistloop{\@series}%
7838   \let\linenum\@line@num
7839   \hilfsskip=\l@dcolwidth%
7840   \advance\hilfsskip by -\wd\hilfsbox
7841   \def\edlabel##1{\xedlabel{##1}}
7842
7843 %
```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for `\edtext`.

```
7844 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
7845 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
7846 %
```

XXX.2.2 Counters, boxes and lengths

`\l@dampcount` `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter for the columns.

```
7847 \newcount\l@dampcount
7848 \l@dampcount=1\relax
7849 \newcount\l@dcolcount
7850 \l@dcolcount=0\relax
7851
7852 %
```

`\hilfsbox` Some (temporary) helper items.

```
\hilfsskip
7853 \newbox\hilfsbox
\Hilfsbox
7854 \newskip\hilfsskip
\hilfscount
7855 \newbox\Hilfsbox
```

```

7856 \newcount\hilfscount
7857
7858 %

```

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).

```

7859 \newdimen\dcoli
7860 \newdimen\dcolii
7861 \newdimen\dcoliii
7862 \newdimen\dcoliv
7863 \newdimen\dcolv
7864 \newdimen\dcolvi
7865 \newdimen\dcolvii
7866 \newdimen\dcolviii
7867 \newdimen\dcolix
7868 \newdimen\dcolx
7869 \newdimen\dcolxi
7870 \newdimen\dcolxii
7871 \newdimen\dcolxiii
7872 \newdimen\dcolxiv
7873 \newdimen\dcolxv
7874 \newdimen\dcolxvi
7875 \newdimen\dcolxvii
7876 \newdimen\dcolxviii
7877 \newdimen\dcolxix
7878 \newdimen\dcolxx
7879 \newdimen\dcolxxi
7880 \newdimen\dcolxxii
7881 \newdimen\dcolxxiii
7882 \newdimen\dcolxxiv
7883 \newdimen\dcolxxv
7884 \newdimen\dcolxxvi
7885 \newdimen\dcolxxvii
7886 \newdimen\dcolxxviii
7887 \newdimen\dcolxxix
7888 \newdimen\dcolxxx
7889 \newdimen\dcolerr % added for error handling
7890
7891 %

```

`\l@dcolwidth` This is a cunning way of storing the columnwidths indexed by the column number `\l@dcolcount`, like an array. (was `\Dimenzuordnung`)

```

7892 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???
7893 \or \dcoli \or \dcolii \or \dcoliii
7894 \or \dcoliv \or \dcolv \or \dcolvi
7895 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
7896 \or \dcolxi \or \dcolxii \or \dcolxiii
7897 \or \dcolxiv \or \dcolxv \or \dcolxvi

```

```

7898 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
7899 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
7900 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
7901 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
7902 \else \dcolerr \fi}
7903
7904 %

```

\stepl@dc colcount This increments the column counter, and issues an error message if it is too large.

```

7905 \newcommand*{\stepl@dc colcount}{\advance\l@dc colcount\@ne
7906 \ifnum\l@dc colcount>30\relax
7907 \led@err@TooManyColumns
7908 \fi}
7909
7910 %

```

\l@dsetmaxcolwidth Sets the column width to the maximum value seen so far.

```

7911 \newcommand{\l@dsetmaxcolwidth}{%
7912 \ifdim\l@dc colwidth < \wd\hilfsbox
7913 \l@dc colwidth = \wd\hilfsbox
7914 \else \relax \fi}
7915
7916 %

```

\measuremcell Measure (recursively) the width required for a math cell.

```

7917 \def\measuremcell #1{%
7918 \ifx #1\ \ifnum\l@dc colcount=0\let\NEXT\relax%
7919 \else\l@dcheckcols%
7920 \l@dc colcount=0%
7921 \let\NEXT\measuremcell%
7922 \fi%
7923 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7924 \stepl@dc colcount%
7925 \l@dsetmaxcolwidth%
7926 \let\NEXT\measuremcell%
7927 \fi\NEXT}
7928
7929 %

```

\measuretcell Measure (recursively) the width required for a text cell.

```

7930 \def\measuretcell #1{%
7931 \ifx #1\ \ifnum\l@dc colcount=0\let\NEXT\relax%
7932 \else\l@dcheckcols%
7933 \l@dc colcount=0%
7934 \let\NEXT\measuretcell%
7935 \fi%

```



```

7936 \else\setbox\hilfsbox=\hbox{#1}%
7937 \step1@dcolcount%
7938 \l@dsetmaxcolwidth%
7939 \let\NEXT\measuretcell%
7940 \fi\NEXT}
7941
7942 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

7943 \def\measuremrow #1\{%
7944 \ifx #1&\let\NEXT\relax%
7945 \else\measuremcell #1\&\&\&%
7946 \let\NEXT\measuremrow%
7947 \fi\NEXT}
7948 %

```

\measuretrow Measure (recursively) the width required for a text row.

```

7949 \def\measuretrow #1\{%
7950 \ifx #1&\let\NEXT\relax%
7951 \else\measuretcell #1\&\&\&%
7952 \let\NEXT\measuretrow%
7953 \fi\NEXT}
7954
7955 %

```

\edtabcolsep The length `\edtabcolsep` controls the distance between columns.

```

7956 \newskip\edtabcolsep
7957 \global\edtabcolsep=10pt
7958
7959 %

```

\variab₉₀ \newcommand{\variab}{\relax}

```

7961
7962 %

```

\l@dcheckcols Check that the number of columns is consistent.

```

7963 \newcommand*{\l@dcheckcols}{%
7964 \ifnum\l@dcolcount=1\relax
7965 \else
7966 \ifnum\l@dampcount=1\relax
7967 \else
7968 \ifnum\l@dcolcount=\l@dampcount\relax
7969 \else
7970 \l@d@err@UnequalColumns

```

```

7971     \fi
7972     \fi
7973     \l@dampcount=\l@dcolcount
7974 \fi}
7975
7976 %

```

\edfilldimen A length.

```

7977 \newdimen\edfilldimen
7978 \edfilldimen=0pt
7979
7980 %

```

\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

```

7981 \newcounter{addcolcount}
7982 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
7983 %

```

XXX.2.3 Tabular typesetting

\setmcellright Typeset (recursively) cells of display math right justified.

```

7984 \def\setmcellright #1&{\def\edlabel##1{}}%
7985     \let\edindex\nulledindex
7986     \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
7987         \let\Next\relax%
7988     \else\l@dcolcount=0%
7989         \let\Next=\setmcellright%
7990     \fi%
7991 \else%
7992     \disablel@dtabfeet%
7993     \stepl@dcolcount%
7994     \disable@notes%
7995     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
7996     \restore@notes%
7997     \letsforverteilen%
7998     \hskip\hilfsskip$\displaystyle{#1}$%
7999     \hskip\edtabcolsep%
8000     \let\Next=\setmcellright%
8001 \fi\Next}
8002
8003 %

```

\settcellright Typeset (recursively) cells of text right justified.

```

8004 \def\settcclright #1&{\def\edlabel##1{}}%
8005 \let\edindex\nulledindex
8006 \ifx #1\\ \ifnum\l@dcclcount=0%\removelastskip
8007 \let\Next\relax%
8008 \else\l@dcclcount=0%
8009 \let\Next=\settcclright%
8010 \fi%
8011 \else%
8012 \disablel@dtabfeet%
8013 \stepl@dcclcount%
8014 \disable@notes%
8015 \setbox\hilfsbox=\hbox{#1}%
8016 \restore@notes%
8017 \letsforverteilen%
8018 \hskip\hilfsskip#1%
8019 \hskip\edtabcolsep%
8020 \let\Next=\settcclright%
8021 \fi\Next}
8022 %

```

\setmcellleft Typeset (recursively) cells of display math left justified.

```

8023 \def\setmcellleft #1&{\def\edlabel##1{}}%
8024 \let\edindex\nulledindex
8025 \ifx #1\\ \ifnum\l@dcclcount=0 \let\Next\relax%
8026 \else\l@dcclcount=0%
8027 \let\Next=\setmcellleft%
8028 \fi%
8029 \else \disablel@dtabfeet%
8030 \stepl@dcclcount%
8031 \disable@notes%
8032 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8033 \restore@notes%
8034 \letsforverteilen%
8035 $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
8036 \let\Next=\setmcellleft%
8037 \fi\Next}
8038 %
8039 %

```

\settcclleft Typeset (recursively) cells of text left justified.

```

8040 \def\settcclleft #1&{\def\edlabel##1{}}%
8041 \let\edindex\nulledindex
8042 \ifx #1\\ \ifnum\l@dcclcount=0 \let\Next\relax%
8043 \else\l@dcclcount=0%
8044 \let\Next=\settcclleft%
8045 \fi%
8046 \else \disablel@dtabfeet%
8047 \stepl@dcclcount%

```

```

8048         \disable@notes%
8049         \setbox\hilfsbox=\hbox{#1}%
8050         \restore@notes%
8051         \letsforverteilen%
8052         #1\hskip\hilfsskip\hskip\edtabcolsep%
8053         \let\Next=\settcclleft%
8054     \fi\Next}
8055 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

8056 \def\setmcellcenter #1{\def\edlabel##1{%
8057     \let\edindex\nulledindex
8058     \ifx #1\\ \ifnum\l@dcolcount=0\let\Next\relax%
8059         \else\l@dcolcount=0%
8060             \let\Next=\setmcellcenter%
8061         \fi%
8062     \else \disablel@dtabfeet%
8063         \stepl@dcolcount%
8064         \disable@notes%
8065         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8066         \restore@notes%
8067         \letsforverteilen%
8068         \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
8069         \hskip\edtabcolsep%
8070         \let\Next=\setmcellcenter%
8071     \fi\Next}
8072
8073 %

```

\settcclcenter Typeset (recursively) cells of text centered.

```

8074 \def\settcclcenter #1{\def\edlabel##1{%
8075     \let\edindex\nulledindex
8076     \ifx #1\\ \ifnum\l@dcolcount=0 \let\Next\relax%
8077         \else\l@dcolcount=0%
8078             \let\Next=\settcclcenter%
8079         \fi%
8080     \else \disablel@dtabfeet%
8081         \stepl@dcolcount%
8082         \disable@notes%
8083         \setbox\hilfsbox=\hbox{#1}%
8084         \restore@notes%
8085         \letsforverteilen%
8086         \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
8087         \hskip\edtabcolsep%
8088         \let\Next=\settcclcenter%
8089     \fi\Next}
8090
8091 %

```

```
\NEXT92 \let\NEXT=\relax
```

```
8093  
8094 %
```

\setmrowright Typeset (recursively) rows of right justified math.

```
8095 \def\setmrowright #1\{\%  
8096   \ifx #1& \let\NEXT\relax  
8097   \else \centerline{\setmcellright #1&\\&\\&}  
8098         \let\NEXT=\setmrowright  
8099   \fi\NEXT}  
8100 %
```

\settroright Typeset (recursively) rows of right justified text.

```
8101 \def\settroright #1\{\%  
8102   \ifx #1& \let\NEXT\relax  
8103   \else \centerline{\settcellright #1&\\&\\&}  
8104         \let\NEXT=\settroright  
8105   \fi\NEXT}  
8106  
8107 %
```

\setmrowleft Typeset (recursively) rows of left justified math.

```
8108 \def\setmrowleft #1\{\%  
8109   \ifx #1& \let\NEXT\relax  
8110   \else \centerline{\setmcellleft #1&\\&\\&}  
8111         \let\NEXT=\setmrowleft  
8112   \fi\NEXT}  
8113 %
```

\settrorleft Typeset (recursively) rows of left justified text.

```
8114 \def\settrorleft #1\{\%  
8115   \ifx #1& \let\NEXT\relax  
8116   \else \centerline{\settcellleft #1&\\&\\&}  
8117         \let\NEXT=\settrorleft  
8118   \fi\NEXT}  
8119  
8120 %
```

\setmrowcenter Typeset (recursively) rows of centered math.

```
8121 \def\setmrowcenter #1\{\%  
8122   \ifx #1& \let\NEXT\relax%  
8123   \else \centerline{\setmcellcenter #1&\\&\\&}  
8124         \let\NEXT=\setmrowcenter  
8125   \fi\NEXT}  
8126 %
```

\settrrowcenter Typeset (recursively) rows of centered text.

```

8127 \def\settrrowcenter #1\\{%
8128   \ifx #1& \let\NEXT\relax
8129   \else \centerline{\settcellcenter #1&\\&\\&}
8130   \let\NEXT=\settrrowcenter
8131   \fi\NEXT}
8132
8133 %

```

```

\newcommand{\nullsetzen}%
8135   \step1@dcolcount%
8136   \l@dcolwidth=0pt%
8137   \ifnum\l@dcolcount=30\let\NEXT\relax%
8138   \l@dcolcount=0\relax
8139   \else\let\NEXT\nullsetzen%
8140   \fi\NEXT}
8141
8142 %

```

\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.

```

8143 \newcommand{\edatleft}[3][\@empty]{%
8144   \ifx#1\@empty
8145     \vbox to 10pt{\vss\hbox{\$ \left#2\vrule width0pt height #3
8146       depth 0pt \right. \$\hss}\vfil}
8147   \else
8148     \vbox to 4pt{\vss\hbox{\$#1\left#2\vrule width0pt height #3
8149       depth 0pt \right. \$}\vfil}
8150   \fi}
8151 %

```

\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.

```

8152 \newcommand{\edatright}[3][\@empty]{%
8153   \ifx#1\@empty
8154     \vbox to 10pt{\vss\hbox{\$ \left.\vrule width0pt height #3
8155       depth 0pt \right#2 \$\hss}\vfil}
8156   \else
8157     \vbox to 4pt{\vss\hbox{\$ \left.\vrule width0pt height #3
8158       depth 0pt \right#2 #1 \$}\vfil}
8159   \fi}
8160
8161 %

```

\edvertline $\edvertline{\langle len \rangle}$ vertical line $\langle len \rangle$ high.

```

8162 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1\vfil}}
8163
8164 %

```

\edvertdots `\edvertdots{⟨len⟩}` vertical dotted line ⟨len⟩ high.

```

8165 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
8166   {\cleaders\hbox{$\math\hbox{.}\vbox to 0.5em{ }$}\vfil}}}
8167
8168 %

```

\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@`.

```

8169 \newcommand{\l@dtabaddcols}[2]{%
8170   \l@dcheckstartend{#1}{#2}%
8171   \ifl@dstartendok
8172   \setcounter{addcolcount}{#1}%
8173   \@whilenum \value{addcolcount}<#2\relax \do
8174     {\advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
8175     \advance\edfilldimen by \edtabcolsep
8176     \stepcounter{addcolcount}}%
8177   \advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
8178   \fi
8179 }
8180
8181 %

```

\ifl@dstartendok `\l@dcheckstartend{⟨startcol⟩}{⟨endcol⟩}` checks that the values of ⟨startcol⟩ and **\l@dcheckstartend** ⟨endcol⟩ are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```

8182 \newif\ifl@dstartendok
8183 \newcommand{\l@dcheckstartend}[2]{%
8184   \l@dstartendoktrue
8185   \ifnum #1<\@ne
8186     \l@dstartendokfalse
8187     \led@err@LowStartColumn
8188   \fi
8189   \ifnum #2>30\relax
8190     \l@dstartendokfalse
8191     \led@err@HighEndColumn
8192   \fi
8193   \ifnum #1>#2\relax
8194     \l@dstartendokfalse
8195     \led@err@ReverseColumns
8196   \fi
8197 }
8198
8199 %

```

`\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.

```
8200 \newcommand*{\edrowfill}[3]{%
8201   \l@dtabaddcols{#1}{#2}%
8202   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
8203 \let\@edrowfill=\edrowfill
8204 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
8205
8206 %
```

`\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`.

```
8207 \newcommand{\leftltab}[1]{%
8208   \hb@xt@ \z@{\vbox{\edtabindent%
8209     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8210
8211 %
```

`\leftrtab` `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```
8212 \newcommand{\leftrtab}[2]{%
8213   #2\hb@xt@ \z@{\vbox{\edtabindent%
8214     \advance\Hilfsskip by \dcoli%
8215     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8216
8217 %
```

`\leftctab` `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```
8218 \newcommand{\leftctab}[2]{%
8219   \hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8220     \advance\Hilfsskip by 0.5\dcoli%
8221     \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
8222     \disablel@dtabfeet$\displaystyle{#2}$}%
8223     \advance\Hilfsskip by -0.5\wd\hilfsbox%
8224     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
8225   #2}
8226
8227 %
```

`\rightctab` `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.


```

8228 \newcommand{\rightctab}[2]{%
8229     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8230         \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8231         #1\hb@xt@{\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8232         \advance\Hilfsskip by 0.5\l@dcolwidth%
8233         \advance\Hilfsskip by -\wd\hilfsbox%
8234         \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8235             \disablel@dtabfeet$\displaystyle{#1}$}%
8236             \advance\Hilfsskip by -0.5\wd\hilfsbox%
8237             \advance\Hilfsskip by \edtabcolsep%
8238             \moveright\Hilfsskip\hbox{ #2}}\hss}%
8239         }
8240     }
8241     %

```

\rightltab `\rightltab{<math><math>}{<text>}` for `\edaftertab` in `\ltab`.

```

8242 \newcommand{\rightltab}[2]{%
8243     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8244         \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8245         #1\hb@xt@{\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8246         \advance\Hilfsskip by \l@dcolwidth%
8247         \advance\Hilfsskip by -\wd\hilfsbox%
8248         \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8249             \disablel@dtabfeet$\displaystyle{#1}$}%
8250             \advance\Hilfsskip by -\wd\hilfsbox%
8251             \advance\Hilfsskip by \edtabcolsep%
8252             \moveright\Hilfsskip\hbox{ #2}}\hss}%
8253         }
8254     }
8255     %

```

\rightrtab `\rightrtab{<math><math>}{<text>}` for `\edaftertab` in `\rtab`.

```

8256 \newcommand{\rightrtab}[2]{%
8257     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8258         \disablel@dtabfeet#2}%
8259         #1\hb@xt@{\z@{\vbox{\edtabindent%
8260         \advance\Hilfsskip by -\wd\hilfsbox%
8261         \advance\Hilfsskip by \edtabcolsep%
8262         \moveright\Hilfsskip\hbox{ #2}}\hss}%
8263     }
8264 }
8265 %

```

\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.

```

8266 \newcommand{\rtab}[1]{%
8267   \l@dnnullfills
8268   \def\edbeforetab##1##2{\lefttrtab{##1}{##2}}%
8269   \def\edaftertab##1##2{\righttrtab{##1}{##2}}%
8270   \measuretbody{#1}%
8271   \l@drestorefills
8272   \variab
8273   \setmrowright #1\\&\\%
8274   \enablel@dtabfeet}
8275
8276 %

```

\measuretbody `\measuretbody{<body>}` measures the array `<body>`.

```

8277 \newcommand{\measuretbody}[1]{%
8278   \disablel@dtabfeet%
8279   \l@dcolcount=0%
8280   \nullsetzen%
8281   \l@dcolcount=0
8282   \measuremrow #1\\&\\%
8283   \global\l@dampcount=1}
8284
8285 %

```

\rtabtext `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```

8286 \newcommand{\rtabtext}[1]{%
8287   \l@dnnullfills
8288   \measuretbody{#1}%
8289   \l@drestorefills
8290   \variab
8291   \settroright #1\\&\\%
8292   \enablel@dtabfeet}
8293
8294 %

```

\measuretbody `\measuretbody{<body>}` measures the tabular `<body>`.

```

8295 \newcommand{\measuretbody}[1]{%
8296   \disable@notes%
8297   \disablel@dtabfeet%
8298   \l@dcolcount=0%
8299   \nullsetzen%
8300   \l@dcolcount=0
8301   \measuretbody #1\\&\\%
8302   \restore@notes%
8303   \global\l@dampcount=1}
8304
8305 %

```

`\ltab` Array with entries left justified.

```

\edbeforetab
\edaftertab
8306 \newcommand{\ltab}[1]{%
8307 \l@dnnullfills
8308 \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
8309 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
8310 \measurebody{#1}%
8311 \l@drestorefills
8312 \variab
8313 \setmrowleft #1\\&\\%
8314 \enablel@dtabfeet}
8315 %
8316 %

```

`\ltabtext` Tabular with entries left justified.

```

8317 \newcommand{\ltabtext}[1]{%
8318 \l@dnnullfills
8319 \measuretbody{#1}%
8320 \l@drestorefills
8321 \variab
8322 \settrrowleft #1\\&\\%
8323 \enablel@dtabfeet}
8324 %
8325 %

```

`\ctab` Array with centered entries.

```

\edbeforetab
\edaftertab
8326 \newcommand{\ctab}[1]{%
8327 \l@dnnullfills
8328 \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
8329 \def\edaftertab##1##2{\rightctab{##1}{##2}}%
8330 \measurebody{#1}%
8331 \l@drestorefills
8332 \variab
8333 \setmrowcenter #1\\&\\%
8334 \enablel@dtabfeet}
8335 %
8336 %

```

`\ctabtext` Tabular with entries centered.

```

8337 \newcommand{\ctabtext}[1]{%
8338 \l@dnnullfills
8339 \measuretbody{#1}%
8340 \l@drestorefills
8341 \variab
8342 \settrrowcenter #1\\&\\%
8343 \enablel@dtabfeet}
8344 %
8345 %

```

```
\spreadtext \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
8347 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
8348 %
```

```
\spreadmath \newcommand{\spreadmath}[1]{%
8350 \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
8351
8352 %
```

\HILFSskip More helpers.

```
\Hilfsskip
8353 \newskip\HILFSskip
8354 \newskip\Hilfsskip
8355
8356 %
```

```
\EDTABINDENT \newcommand{\EDTABINDENT}{%
8358 \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
8359 \else\step\l@dcolcount%
8360 \advance\Hilfsskip by\l@dcolwidth%
8361 \ifdim\l@dcolwidth=0pt\advance\hilfscout\@ne
8362 \else\advance\Hilfsskip by \the\hilfscout\edtabcolsep%
8363 \hilfscout=1\fi%
8364 \let\NEXT=\EDTABINDENT%
8365 \fi\NEXT}%
8366 %
```

\edtabindent (was \tabindent)

```
8367 \newcommand{\edtabindent}{%
8368 \l@dcolcount=0\relax
8369 \Hilfsskip=0pt%
8370 \hilfscout=1\relax
8371 \EDTABINDENT%
8372 \hilfsskip=\hsize%
8373 \advance\hilfsskip -\Hilfsskip%
8374 \Hilfsskip=0.5\hilfsskip%
8375 }%
8376
8377 %
```

\EDTAB (was \TAB)

```
8378 \def\EDTAB #1|#2|{%
8379 \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
8380 \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
8381 \advance\tabelskip -\wd\tabhilfbox%
```

```

8382 \advance\tabelskip -\wd\tabHilfbox%
8383 \unhbox\tabhilfbox\hskip\tabelskip%
8384 \unhbox\tabHilfbox}%
8385
8386 %

```

`\EDTABtext` (was `\TABtext`)

```

8387 \def\EDTABtext #1|#2|{%
8388   \setbox\tabhilfbox=\hbox{#1}%
8389   \setbox\tabHilfbox=\hbox{#2}%
8390   \advance\tabelskip -\wd\tabhilfbox%
8391   \advance\tabelskip -\wd\tabHilfbox%
8392   \unhbox\tabhilfbox\hskip\tabelskip%
8393   \unhbox\tabHilfbox}%
8394 %

```

`\tabhilfbox` Further helpers.

```

8395 \newbox\tabhilfbox
8396 \newbox\tabHilfbox
8397
8398 %

```

XXX.2.4 Environments

`edarrayl edarrayc edarrayr` The ‘environment’ forms for `\ltab`, `\ctab` and `\rtab`.

```

8399 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{}
8400 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{}
8401 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{}
8402
8403 %

```

`edtabularl edtabularc edtabularr` The ‘environment’ forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

```

8404 \newenvironment{edtabularl}{\l@dcollect@body\ltabtext}{}
8405 \newenvironment{edtabularc}{\l@dcollect@body\ctabtext}{}
8406 \newenvironment{edtabularr}{\l@dcollect@body\rtabtext}{}
8407
8408 %

```

XXXI Quotation’s commands

`\initnumbering@quote` This macro, called at the beginning of any numbered section, locally redefines the quotation and quote environments, in order to allow their use inside of numbered sections.

```

\quotation \initnumbering@quote defines quotation environment.
\endquotation
\quote
\endquote
8409 \newcommand{\initnumbering@quote}{
8410 \ifnoquotation@else
8411 \renewcommand{\quotation}{\par\leavevmode%
8412 \parindent=1.5em%
8413 \skipnumbering%
8414 \ifautopar%
8415 \vskip-\parskip%
8416 \else%
8417 \vskip\topsep%
8418 \fi%
8419 \global\leftskip=\leftmargin%
8420 \global\rightskip=\leftmargin%
8421 }
8422 \renewcommand{\endquotation}{\par%
8423 \global\leftskip=0pt%
8424 \global\rightskip=0pt%
8425 \leavevmode%
8426 \skipnumbering%
8427 \ifautopar%
8428 \vskip-\parskip%
8429 \else%
8430 \vskip\topsep%
8431 \fi%
8432 }
8433 \renewcommand{\quote}{\par\leavevmode%
8434 \parindent=0pt%
8435 \skipnumbering%
8436 \ifautopar%
8437 \vskip-\parskip%
8438 \else%
8439 \vskip\topsep%
8440 \fi%
8441 \global\leftskip=\leftmargin%
8442 \global\rightskip=\leftmargin%
8443 }
8444 \renewcommand{\endquote}{\par%
8445 \global\leftskip=0pt%
8446 \global\rightskip=0pt%
8447 \leavevmode%
8448 \skipnumbering%
8449 \ifautopar%
8450 \vskip-\parskip%
8451 \else%
8452 \vskip\topsep%
8453 \fi%
8454 }
8455 \fi
8456 }

```

8457 %

XXXII Section's title commands

XXXII.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```
8458 \newcommand{\ledsectnotoc}{\let\addcontentsline@gobblethree}
8459 %
```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```
8460 \newcommand{\ledsectnomark}{%
8461   \let\chaptermark@gobble%
8462   \let\sectionmark@gobble%
8463   \let\subsectionmark@gobble%
8464 }
8465 %
```

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).

```
8466 \catcode`\#=12
8467 \notbool{@noeled@sec}{%
8468 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
8469 \ifl@dmemoir
8470 \newcommand\beforeeledchapter{%
8471 \clearforchapter%
8472 }
8473 \else
8474 \newcommand\beforeeledchapter{%
8475 \if@openright%
8476 \cleardoublepage%
8477 \else%
8478 \clearpage%
8479 \fi%
8480 }
8481 \fi
8482 %
```

XXXII.4 Auxiliary commands

`\print@leftmargin@eledsection` `\print@leftmargin@eledsection` and `\print@rightmargin@eledsection` are added by `reledmac` inside the code of sectioning command, in order to affix lines numbers. They include tests for RTL languages.

```
8483 \def\print@rightmargin@eledsection{%
8484 \if@eled@sectioning%
8485 \begingroup%
8486 \if@RTL%
8487 \let\llap\rlap%
8488 \let\leftlinenum\rightlinenum%
8489 \let\leftlinenumR\rightlinenumR%
8490 \let\l@drd@ta\l@dld@ta%
8491 \let\l@drsn@te\l@dlsn@te%
8492 \fi%
8493 \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
8494 \endgroup%
8495 \fi%
```



```

8496 }%
8497
8498 \def\print@leftmargin@eledsection{%
8499   \if@eled@sectioning%
8500     \leavevmode%
8501     \begingroup%
8502     \if@RTL%
8503       \let\rlap\llap%
8504       \let\rightlinenum\leftlinenum%
8505       \let\rightlinenumR\leftlinenumR%
8506       \let\l@dld@ta\l@drd@ta%
8507       \let\l@dlsn@te\l@drsn@te%
8508     \fi%
8509     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
8510     \endgroup%
8511   \fi%
8512 }%
8513
8514 %

```

XXXII.5 Patching standard commands

`\M@sect` We have to patch `ℒℒℒ`, `book` and `memoir` sectioning commands in order to:

- `\@mem@old@ssect` • Disable `\edtext` inside.
- `\@makechapterhead` • Disable page breaking (for `\chapter`).
- `\@makechapterhead` • Add line numbers and sidenotes.
- `\@makeschapterhead`
- `\@sect`
- `\@ssect`

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.

```

8515 \AtBeginDocument{%
8516
8517
8518 \pretocmd{\M@sect}
8519   {\let\old@edtext=\edtext%
8520    \let\edtext=\dummy@edtext@showlemma%
8521   }
8522   {}
8523   {}
8524
8525 \apptocmd{\M@sect}
8526   {\let\edtext=\old@edtext}
8527   {}
8528   {}
8529
8530 \patchcmd{\M@sect}
8531   { #9}

```

```

8532 { #9%
8533 \print@rightmargin@eledsection%
8534 }
8535 {}
8536 {}
8537
8538 \patchcmd{\M@sect}
8539 {\hskip #3\relax}
8540 {\hskip #3\relax%
8541 \print@leftmargin@eledsection%
8542 }
8543 {}
8544 {}
8545
8546 \patchcmd{\@mem@old@ssect}
8547 {#5}
8548 {#5%
8549 \print@leftmargin@eledsection%
8550 }
8551 {}
8552 {}
8553
8554 \patchcmd{\@mem@old@ssect}
8555 {\hskip #1}
8556 {\hskip #1%
8557 \print@rightmargin@eledsection%
8558 }
8559 {}
8560 {}
8561
8562
8563
8564 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8565 \if@eled@sectioning\else%
8566 \ifl@dprintingpages\else%
8567 \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a
\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
8568 \fi%
8569 \fi%
8570 }
8571 {}
8572 {}
8573
8574 \patchcmd{\@makechapterhead}
8575 {#1}
8576 {\print@leftmargin@eledsection%
8577 #1%
8578 \print@rightmargin@eledsection%

```

```

8579 }
8580 {}
8581 {}
8582
8583 \patchcmd{\@makechapterhead}% For BIDI
8584 {\if@RTL\raggedleft\else\raggedright\fi}%
8585 {\if@eled@sectioning\else%
8586   \if@RTL\raggedleft\else\raggedright\fi%
8587   \fi%
8588 }%
8589 {}%
8590 {}%
8591
8592 \patchcmd{\@makeschapterhead}
8593 {#1}
8594 {\print@leftmargin@eledsection%
8595   #1%
8596   \print@rightmargin@eledsection%
8597 }
8598 {}
8599 {}
8600
8601 \pretocmd{\@sect}
8602 {\let\old@edtext=\edtext
8603   \let\edtext=\dummy@edtext@showlemma%
8604 }
8605 {}
8606 {}
8607
8608 \apptocmd{\@sect}
8609 {\let\edtext=\old@edtext}
8610 {}
8611 {}
8612
8613 \pretocmd{\@ssect}
8614 {\let\old@edtext=\edtext%
8615   \let\edtext=\dummy@edtext@showlemma%
8616 }
8617 {}
8618 {}
8619
8620 \apptocmd{\@ssect}
8621 {\let\edtext=\old@edtext}
8622 {}
8623 {}
8624
8625 %

```

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.

```

8626 \@ifpackageloaded{nameref}{
8627
8628     \patchcmd{\NR@sect}
8629         {#8}
8630         {#8%
8631         \print@rightmargin@eledsection%
8632         }
8633         {}
8634         {}
8635
8636     \patchcmd{\NR@sect}
8637         {\hskip #3\relax}
8638         {\hskip #3\relax%
8639         \print@leftmargin@eledsection%
8640         }
8641         {}
8642         {}
8643
8644     \patchcmd{\NR@ssect}
8645         {#5}
8646         {#5%
8647         \print@rightmargin@eledsection%
8648         }
8649         {}
8650         {}
8651
8652     \patchcmd{\NR@ssect}
8653         {\hskip #1}
8654         {\hskip #1%
8655         \print@leftmargin@eledsection%
8656         }
8657         {}
8658         {}
8659     }%
8660     {
8661         \patchcmd{\@sect}
8662             {#8}
8663             {#8%
8664             \print@rightmargin@eledsection%
8665             }
8666             {}
8667             {}
8668
8669         \patchcmd{\@sect}
8670             {\hskip #3\relax}
8671             {\hskip #3\relax%
8672             \print@leftmargin@eledsection%
8673             }
8674             {}
8675             {}

```

```

8676
8677 \patchcmd{\@ssect}
8678   {\#5}
8679   {\#5%
8680   \print@rightmargin@eledsection%
8681   }
8682   {}
8683   {}
8684
8685 \patchcmd{\@ssect}
8686   {\hskip #1}
8687   {\hskip #1%
8688   \print@leftmargin@eledsection%
8689   }
8690   {}
8691   {}
8692   }%
8693 }%
8694 %

```

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 `#`.

```

8695 {}}%
8696 \protect\catcode`\#=6 %Space NEEDS by \catcode
8697 %

```

\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.

```

8698 \AtBeginDocument{%
8699 \patchcmd{\chapter}{\clearforchapter}{%
8700   \if@eled@sectioning\else%
8701   \ifl@dprintingpages\else%
8702     \clearforchapter%
8703   \fi%
8704   \fi%
8705   }%
8706   {}%
8707   {}%
8708
8709 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8710   \if@eled@sectioning\else%
8711   \ifl@dprintingpages%
8712     \endgraf%
8713   \else%
8714     \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
8715   \fi%

```

```

8716 \fi%
8717 }%
8718 {}%
8719 {}%
8720 }%
8721 %

```

\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.

```

8722 \newif\if@eled@sectioning%
8723 %

```

We reopen a new `\notbool{@noeled@sec}` statement, as we will define the `\elesection` commands.

```

8724 \notbool{@noeled@sec}{%
8725 %

```

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.

```

8726 \newwrite\eled@sectioning@out
8727 %

```

\eledchapter **\eledsection** And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsubsection
\eledsubsubsection
\eledchapter*
\eledsection*
\eledsubsection*
\eledsubsubsection*
8728 \newcommand{\eledchapter}[2] [] {%
8729   #2%
8730   \ifledRcol%
8731     \immediate\write\eled@sectioningR@out{%
8732       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}
8733     }%
8734   \else%
8735     \immediate\write\eled@sectioning@out{%
8736       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{-}{-}
8737     }%
8738   \fi%
8739 }
8740
8741 \newcommand{\eledsection}[2] [] {%
8742   #2%
8743   \ifledRcol%
8744     \immediate\write\eled@sectioningR@out{%
8745       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{-}{R}

```

```

8746     }%
8747 \else%
8748     \immediate\write\eled@sectioning@out{%
8749     \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{}}
8750     }%
8751 \fi%
8752 }
8753
8754 \newcommand{\eledsubsection}[2][{}]{%
8755 #2%
8756 \ifledRcol%
8757     \immediate\write\eled@sectioningR@out{%
8758     \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{}}{R}
8759     }%
8760 \else%
8761     \immediate\write\eled@sectioning@out{%
8762     \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{}}
8763     }%
8764 \fi%
8765 }
8766 \newcommand{\eledsubsubsection}[2][{}]{%
8767 #2%
8768 \ifledRcol%
8769     \immediate\write\eled@sectioningR@out{%
8770     \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
8771     }{R}}
8772     }%
8773 \else%
8774     \immediate\write\eled@sectioning@out{%
8775     \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}
8776     }{}{}
8777     }%
8778 \fi%
8779 }
8780 \WithSuffix\newcommand{\eledchapter*}[2][{}]{%
8781 #2%
8782 \ifledRcol%
8783     \immediate\write\eled@sectioningR@out{%
8784     \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8785     }%
8786 \else%
8787     \immediate\write\eled@sectioning@out{%
8788     \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{}}
8789     }%
8790 \fi%
8791 }
8792

```

```

8793 \WithSuffix\newcommand\eledsection*[2][]{%
8794   #2%
8795   \ifledRcol%
8796     \immediate\write\eled@sectioningR@out{%
8797       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8798     }%
8799   \else%
8800     \immediate\write\eled@sectioning@out{%
8801       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{L}
8802     }%
8803   \fi%
8804 }
8805
8806 \WithSuffix\newcommand\eledsubsection*[2][]{%
8807   #2%
8808   \ifledRcol%
8809     \immediate\write\eled@sectioningR@out{%
8810       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8811     }%
8812   \else%
8813     \immediate\write\eled@sectioning@out{%
8814       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}
8815       }{*}{L}
8816     }%
8817   \fi%
8818 }
8819
8820 \WithSuffix\newcommand\eledsubsubsection*[2][]{%
8821   #2%
8822   \ifledRcol%
8823     \immediate\write\eled@sectioningR@out{%
8824       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
8825       }{*}{R}
8826     }%
8827   \else%
8828     \immediate\write\eled@sectioning@out{%
8829       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}
8830       }{*}{L}
8831     }%
8832   \fi%
8833 }
8834 %

```

XXXII.7 Macros written in the auxiliary file

`\eled@chapter` The sectioning macros, called in the auxiliary file. They have five arguments:
`\eled@section`
`\eled@subsection` 1. Optional arguments of \LaTeX sectioning command.
`\eled@subsubsection`

2. Mandatory arguments of \LaTeX sectioning command.
3. Pstart number.
4. Side: R if right, nothing if left.
5. Starred or not.

```

8832 \def\eled@chapter#1#2#3#4#5{%
8833   \ifstrempy{#4}%
8834   {%
8835     \ifstrempy{#1}%
8836     {%
8837       \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}%
8838       \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%
8839       }%Need for \pairs, because of using parbox.
8840       {%
8841         \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
8842         \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}}%Need for \pairs, because of using parbox.
8843       }%
8844     }%
8845     {%
8846       \ifstrempy{#1}%
8847       {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
8848       {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#1}{#2}}}%Bug in LaTeX!
8849     }%
8850     \listcsgadd{eled@sections#5@@}{#3}%
8851   }
8852 \def\eled@section#1#2#3#4#5{%
8853   \ifstrempy{#4}%
8854   {\ifstrempy{#1}%
8855   {%
8856     \csgdef{eled@sectioning@#3#5}{\section{#2}}%
8857     \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#2}}}%Need for \pairs, because of using parbox.
8858   }%
8859   {%
8860     \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
8861     \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{#1}}}%Need for \pairs, because of using parbox.
8862   }%
8863   }%
8864   {\ifstrempy{#1}%
8865   {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%

```

```

8866     {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
8867     }
8868     \listcsgadd{eled@sections#5@@}{#3}%
8869     }
8870 \def\eled@subsection#1#2#3#4#5{%
8871     \ifstrempy{#4}%
8872     {\ifstrempy{#1}%
8873     {%
8874     \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
8875     \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)
8876     }%
8877     {%
8878     \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
8879     \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)
8880     }%
8881     }%
8882     {\ifstrempy{#1}%
8883     {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
8884     {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
8885     }
8886     \listcsgadd{eled@sections#5@@}{#3}%
8887     }
8888 \def\eled@subsubsection#1#2#3#4#5{%
8889     \ifstrempy{#4}%
8890     {\ifstrempy{#1}%
8891     {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
8892     {\csgdef{eled@sectioning@#3#5}{\subsubsection{#1}{#2}}}%
8893     }%
8894     {\ifstrempy{#1}%
8895     {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
8896     {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in
LaTeX!
8897     }
8898     \listcsgadd{eled@sections#5@@}{#3}%
8899     }
8900
8901 %

```

End of the conditional test about noeledsec option.

```

8902 }{}
8903 %

```

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.

```
8904 \def\normal@page@break{}
8905 %
```

\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.

```
8906 \def\l@prev@pb{}
8907 \def\l@prev@nopb{}
8908 %
```

\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.

```
8909 \newcommand{\ledpb}{\write\linenumout{\string\led@pb}}
8910 \newcommand{\ledpbnum}[1]{\write\linenumout{\string\led@pbnum{#1}}}
8911 \newcommand{\lednopb}{\write\linenumout{\string\led@nopb}}
8912 \newcommand{\lednopbnum}[1]{\write\linenumout{\string\led@nopbnum{#1}}}
8913 %
```

\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.

```
8914 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
8915 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
8916 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
8917 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
8918 %
```

\ledpbsetting The \ledpbsetting macro only changes the value of \led@pb@macro, for which the default value is before.

```
8919 \def\led@pb@setting{before}
8920 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
8921 %
```

`\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`.

```

8922 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\
pagebreak[4]}}{}
8923 \newcommand{\led@check@nopb}{%
8924   \IfStrEq{\led@pb@setting}{before}{%
8925     \xifinlist{\the\absline@num}{\l@prev@nopb}%
8926     {\numdef{\abs@prevline}{\the\absline@num-1}%
8927     \xifinlist{\abs@prevline}{\normal@page@break}%
8928     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
8929     {}}%
8930   {}}%
8931   {}%
8932   {}%
8933   \IfStrEq{\led@pb@setting}{after}{%
8934     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
8935     \xifinlist{\the\absline@num}{\normal@page@break}%
8936     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
8937     {}}%
8938   }%
8939   {}%
8940   {}%
8941   {}%
8942 }
8943 %

```

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.

```

8944 \newcommand{\check@pb@in@verse}{%
8945   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
8946   \ifnum\page@num=\last@page@num\else%If we have change page
8947   \IfStrEq{\led@pb@setting}{before}{%

```

```

8948         \numdef{\abs@line@verse}{\the\absline@num-1}%
8949         \ledpbnum{\abs@line@verse}%
8950     }{}%
8951     \IfStrEq{\led@pb@setting}{after}{%
8952         \numdef{\abs@line@verse}{\the\absline@num-1}%
8953         \lednopbnum{\abs@line@verse}%
8954     }{}%
8955     \fi%
8956 \fi\fi\fi%
8957 }
8958 %

```

XXXV Tools for hyperref package

`\Hy@raisedlink@left` The hyperref package provides a `\Hy@raisedlink` command, to be used to add an anchor to the top of a line and not to the bottom of it.³⁵

However, this command disrupts the line breaking mechanism when it is called before any word. This is why `reledmac` defines `\Hy@raisedlink@left` that is called to the left of words, at the beginning of `\edtext` or inside the `\edlabel` commands.³⁶

```

8959 \def\Hy@raisedlink@left#1{%
8960     \ifvmode
8961         #1%
8962     \else
8963         \Hy@SaveSpaceFactor
8964         \llap{\smash{%
8965             \begingroup
8966                 \let\HyperRaiseLinkLength\@tempdima
8967                 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
8968                 \HyperRaiseLinkHook
8969             \expandafter\endgroup
8970             \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
8971                 \Hy@RestoreSpaceFactor
8972                 #1%
8973                 \Hy@SaveSpaceFactor
8974             }}%
8975         \Hy@RestoreSpaceFactor
8976         \penalty\@M\hskip\z@\relax
8977     \fi
8978 }
8979 %
8980

```

³⁵<http://tex.stackexchange.com/a/17138/7712>.

³⁶The code is inspired by an answer given by @unbonpetit. Thanks to him. <http://texnique.fr:80/osqa/questions/781/hyraisedlink-perturbe-la-maniere-dont-se-fait-la-coupe-de-ligne/801>.

XXXVI Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

8981 \ifeledmaccompat@%
8982
8983 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
8984 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
8985 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
8986 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
8987
8988 \unless\ifnocritical@
8989 \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
8990 \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
8991 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
8992 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
8993 \let\hsizetwocol\Xhsizetwocol
8994 \let\hsizethreecol\Xhsizethreecol
8995 \let\bhookXnote\Xbhooknote
8996 \let\boxsymlinenum\Xboxsymlinenum
8997 \let\symlinenum\Xsymlinenum
8998 \let\beforenumberinfootnote\Xbeforenumber
8999 \let\afternumberinfootnote\Xafternumber
9000 \let\beforeXsymlinenum\Xbeforesymlinenum
9001 \let\afterXsymlinenum\Xaftersymlinenum
9002 \let\inplaceofnumber\Xinplaceofnumber
9003 \let\Xlemmaseparator\lemmaseparator
9004 \let\afterlemmaseparator\Xafterlemmaseparator
9005 \let\beforelemmaseparator\Xbeforelemmaseparator
9006 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
9007 \let\txtbeforeXnotes\Xtxtbeforenotes
9008 \let\afterXrule\Xafterrule
9009 \let\numberonlyfirstinline\Xnumberonlyfirstinline
9010 \let\numberonlyfirstintwolines\Xnumberonlyfirstintwolines
9011 \let\nonumberinfootnote\Xnonumberinfootnote
9012 \let\pstartinfootnote\Xpstart
9013 \let\pstartinfootnoteeverytime\Xpstarteverytime
9014 \let\onlyXpstart\Xonlypstart
9015 \let\Xnonumberinfootnote\Xnonumber
9016 \let\nonbreakableafternumber\Xnonbreakableafternumber
9017 \let\maxhXnotes\Xmaxhnotes
9018 \let\beforeXnotes\Xbeforenotes
9019 \let\boxlinenum\Xboxlinenum
9020 \let\boxlinenumalign\Xboxlinenumalign
9021 \let\boxstartlinenum\Xboxstartlinenum
9022 \let\boxendlinenum\Xboxendlinenum
9023 \let\twolines\Xtwolines
9024 \let\morethantwolines\Xmorethantwolines
9025 \let\twolinesbutnotmore\Xtwolinesbutnotmore
9026 \let\twolinesonlyinsamepage\Xtwolinesonlyinsamepage

```

```

9027 \fi
9028
9029 \unless\ifnofamiliar@
9030   \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
9031 \fi
9032 \newcommandx{\parafootsep}[2][1,usedefault]{%
9033   \Xparafootsep[#1]{#2}%
9034   \parafootsepX[#1]{#2}
9035 }%
9036
9037 \newcommandx{\afternote}[2][1,usedefault]{%
9038   \Xafternote[#1]{#2}%
9039   \afternoteX[#1]{#2}%
9040 }%
9041
9042 \unless\ifnoend@
9043   \let\XendXtwolines\Xendtwolines
9044   \let\XendXmorethantwolines\Xendmorethantwolines
9045   \let\XhookXendnote\Xendhooknote
9046   \let\boxXendlinenum\Xendboxlinenum%
9047   \let\boxXendlinenumalign\Xendboxlinenumalign%
9048   \let\boxXendstartlinenum\Xendboxstartlinenum%
9049   \let\boxXendendlinenum\Xendboxendlinenum%
9050   \let\XendXlemmaseparator\Xendlemmaseparator
9051   \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
9052   \let\XendXafterlemmaseparator\Xendafterlemmaseparator
9053   \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
9054 \fi
9055
9056 \AtBeginDocument{%
9057   \ifdef\lineref{}\let\lineref\edlineref}%
9058 }%
9059
9060
9061 \fi%
9062 %

```

</code>

Appendix A Things to do when changing versions

Appendix A.1 Migrating from edmac to ledmac

If you have never used edmac, ignore this section. If you have used edmac and are starting on a completely new document, ignore this section. Only read this section if you are converting an original edmac document to use ledmac.

The package still provides the original `\text` command, but it is (a) deprecated, and (b) its name has been changed³⁷ to `\critext`; use the `\edtext` macro instead. However, if you do use `\critext` (the new name for `\text`), the following is a reminder.

`\critext` Within numbered paragraphs, footnotes and endnotes are generated by forms of the `\critext` macro:

```
\critext{⟨lemma⟩}⟨commands⟩/
```

The `⟨lemma⟩` argument is the lemma in the main text: `\critext` both prints this as part of the text, and makes it available to the `⟨commands⟩` you specify to generate notes. The `/` at the end terminates the command; it is part of the macro's definition so that spaces after the macro will be treated as significant.

For example:

<code>I saw my friend \critext{Smith}</code>	1 I saw my friend
<code>\Afootnote{Jones C, D.}/</code>	2 Smith on Tuesday.
on Tuesday.	<u>2 Smith]</u> Jones C, D.

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

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

<code>\critext{I saw my friend</code>	1 I saw my friend
<code>\critext{Smith}{\Afootnote{Jones</code>	2 Smith on Tuesday.
<code>C, D.}/ on Tuesday.}</code>	<u>2 Smith]</u> Jones C, D.
<code>\Bfootnote{The date was</code>	<u>1–2 I saw my friend</u>
<code>July 16, 1954.}</code>	Smith on Tuesday.] The
<code>/</code>	date was July 16, 1954.

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

The second argument of the `\critext` macro, `⟨commands⟩`, is the same as the second argument to the `\edtext` macro.

It is possible to define aliases for `\critext`, which can be easier to type. You can make a single character substitute for `\critext` by saying this:

```
\catcode`\<=\active
```

³⁷A name like `\text` is likely to be defined by other \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. 121 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.³⁸

If for some reason you do not want to make the modifications to use the new functions of `eledmac`, you can continue using your own `\XXXXXfmt` command, but you must replace:

```
\renewcommand*{XXXXfmt}[3]
```

with

```
\renewcommandx*{XXXXfmt}[4][4=Z]
```

³⁸<https://github.com/maieul/ledmac/issues>

If you do not make that, you will get a spurious [X], where X is series letter.

If you used a `\protect` command inside a `\footnote` command inside a numbered section, you must change the `\protect` to `\noexpand`. Otherwise the command after the `\protect` will be discarded.

Appendix A.3 Migration to eledmac 1.5.1

The version 1.5.1 corrects a bug in `stanzaindentsrepetition` (cf. 9.3 p. 49). 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 `\printnpnum`. This spurious space has been deleted. However, if you want to keep the previous spurious space, you may load the package with the `oldprintnpnumspace` option.

If you have redefined `\endprint`, you must:

- Contact us and ask for the feature that required your hack, in order to avoid such a hack in the future.
- Use the new fifth argument.
- Add `\xdef\@currentseries{#4}` at the beginning of your own command.

Appendix A.7 Migration to eledmac 1.22.0

The `\ledinnote` command now takes a first optional argument, which is the label for the hyperreference. If you have redefined it, change your redefinition, and check whether you can avoid this redefinition by only redefining `\ledinnotemark`.

Appendix A.8 Migration to eledmac 1.23.0

You must delete the numbered auxiliary files before compiling with the new version of eledmac.

Appendix A.9 Migration from eledmac to reledmac

There are many changes in reledmac which require the user to make modifications.

Appendix A.9.1 Risk of ‘no room for a new’

The risk to obtain a ‘no room for a new something’ error is greater in reledmac than it is in eledmac. See 19.1.3 p. 67 in order to know how to limit it.

Appendix A.9.2 Multiple indices with memoir

Eledmac and ledmac used the specific indexing tools of the memoir class designed to produce multiple indices. However, eledmac could also use imakeidx or indextools tools independently of the memoir class. This system forced to maintain redundant code. Since reledmac, we use only the imakeidx or indextools tools.

Consequently: Users of memoir are invited to use indextool or imakeidx to produce multiple indices.

Appendix A.9.3 Deprecated commands and options

The table of deprecated commands and their alternatives follows. Note that the way some commands must be used may have changed. Please read the handbook.

<i>Deprecated command</i>	<i>Replaced with</i>
<code>\addfootins</code>	<code>\newseries</code>
<code>\addfootinsX</code>	<code>\newseries</code>
<code>\critext</code>	<code>\edtext</code>
<code>\falseverse</code>	<code>\newverse</code>
<code>\interparanoteglue</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\ledchapter</code>	<code>\eledchapter</code>
<code>\ledsection</code>	<code>\eledsection</code>
<code>\ledsetnormalparstuff</code>	<code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code>
<code>\ledsubsection</code>	<code>\eledsubsection</code>
<code>\ledsubsubsection</code>	<code>\eledsubsubsection</code>
<code>\noeledsec</code>	Package option <code>noeledsec</code>
<code>\noendnotes</code>	Package option <code>noendnotes</code>
<code>\pageparbreak</code>	<code>\ledpb</code>

The `ledsecnolinenumber` option has been removed, because it was related to deprecated commands.

The `oldprintnpnumspace` option has been removed too, because it was related to a historical bug. The `\usingedtext` and `\usingcritext` commands are also deprecated.

Appendix A.9.4 \renewcommand replaced by command

Many uses of \renewcommand have been replaced with uses of specific commands. Please read handbook about specific commands.

<i>Deprecated \renewcommand</i>	<i>Replaced with</i>
\@led@extranofeet	\newseries
\apprefprefixmore	\setapprefprefixmore
\apprefprefixsingle	\setapprefprefixsingle
\endstanzaextra	Optional argument of \&
\hangingsymbol	\sethangingsymbol
\ledfootinsdim	\Xmaxhnotes and \maxhnotesX
\parafootftmsep	\Xparafootsep and \parafootsepX
\notenumfont	\Xnotenumfont, \Xendnotenumfont and \notenumfontX
\notefontsetup	\Xnotefontsize, \Xendnotefontsize and \notefontsizeX
\sidenotesep	\setsidenotsep
\startstanzahook	Optional argument of \stanza
\symplinenum	\Xsymplinenum

Appendix A.9.5 Commands the names of which have been changed

In order to help the migration from eledmac to reledmac, you may load reledmac with eledmac-compat option. However, it is advised not to, and to change the command names themselves instead. In many cases, you use only a few of them, except the \footparagraph command.

<i>Old command</i>	<i>New command</i>
\footparagraph	\Xarrangement
\footnormal	\Xarrangement
\foottwocol	\Xarrangement
\footthreecol	\Xarrangement
\footparagraphX	\arrangementX
\footnormalX	\arrangementX
\foottwocolX	\arrangementX
\footthreecolX	\arrangementX
\afterlemmaseparator	\Xafterlemmaseparator
\afternote	\Xafternote and \afternoteX
\afternumberinfootnote	\Xafternumber
\afterXrule	\Xafterrule
\afterXsymplinenum	\Xaftersymplinenum
\beforelemmaseparator	\Xbeforelemmaseparator
\beforenumberinfootnote	\Xbeforenumber
\beforeXnotes	\Xbeforenotes
\beforeXsymplinenum	\XbeforeXsymplinenum

<i>Old command</i>	<i>New command</i>
<code>\bhookXnote</code>	<code>\Xbhookendnote</code>
<code>\bhookXnote</code>	<code>\Xbhooknote</code>
<code>\boxendlinenum</code>	<code>\Xboxendlinenum</code>
<code>\boxlinenum</code>	<code>\Xboxlinenum</code>
<code>\boxlinenumalign</code>	<code>\Xboxlinenumalign</code>
<code>\boxstartlinenum</code>	<code>\Xboxstartlinenum</code>
<code>\boxsymlinenum</code>	<code>\Xboxsymlinenum</code>
<code>\boxXendlinenum</code>	<code>\Xendboxlinenum</code>
<code>\boxXendlinenumalign</code>	<code>\Xendboxlinenumalign</code>
<code>\boxXendstartlinenum</code>	<code>\boxXendstartlinenum</code>
<code>\letboxXendendlinenum</code>	<code>\Xendletboxendlinenum</code>
<code>\hsizetwocol</code>	<code>\Xhsizetwocol</code>
<code>\hsizethreecol</code>	<code>\Xhsizethreecol</code>
<code>\inplaceoflemmaseparator</code>	<code>\Xinplaceoflemmaseparator</code>
<code>\inplaceofnumber</code>	<code>\Xinplaceofnumber</code>
<code>\lemmaseparator</code>	<code>\Xlemmaseparator</code>
<code>\maxhXnotes</code>	<code>\Xmaxhnotes</code>
<code>\morethantwolines</code>	<code>\Xmorethantwolines</code>
<code>\nonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\notesXwidthliketwocolumns</code>	<code>\noteswidthliketwocolumnsX</code>
<code>\noXlemmaseparator</code>	<code>\Xnolemmaseparator</code>
<code>\numberonlyfirstinline</code>	<code>\Xnumberonlyfirstinline</code>
<code>\numberonlyfirstintwolines</code>	<code>\Xnumberonlyfirstintwolines</code>
<code>\nonbreakableafternumber</code>	<code>\Xnonbreakableafternumber</code>
<code>\onlyXpstart</code>	<code>\Xonlypstart</code>
<code>\parafootsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\pstartinfootnote</code>	<code>\Xpstart</code>
<code>\pstartinfootnoteeverytime</code>	<code>\Xpstarteverytime</code>
<code>\symlinenum</code>	<code>\Xsymlinenum</code>
<code>\twolines</code>	<code>\Xtwolines</code>
<code>\twolinesbutnotmore</code>	<code>\Xtwolinesbutnotmore</code>
<code>\twolinesonlyinsamepage</code>	<code>\Xtwolinesonlyinsamepage</code>
<code>\txtbeforeXnotes</code>	<code>\Xtxtbeforenotes</code>
<code>\XendXafterlemmaseparator</code>	<code>\Xendafterlemmaseparator</code>
<code>\XendXbeforelemmaseparator</code>	<code>\Xendbeforelemmaseparator</code>
<code>\XendXinplaceoflemmaseparator</code>	<code>\Xendinplaceoflemmaseparator</code>
<code>\XendXlemmaseparator</code>	<code>\Xendlemmaseparator</code>
<code>\XendXmorethantwolines</code>	<code>\Xendmorethantwolines</code>
<code>\XendXtwolines</code>	<code>\Xendtwolines</code>
<code>\Xnonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\lineref</code>	<code>\edlineref</code>

Appendix A.9.6 Endnotes

With *reledmac*, there is now one auxiliary file for every endnotes set (*.Aend*, *.Bend*, *.Cend* etc.). If you have overridden `\doendnotes` (which you would not have done) you must adapt your code.

Appendix A.9.7 Z Series

The ‘Z’ series of notes has been removed. Only five series are provided now by default: A, B, C, D, E.

Appendix A.9.8 Internal commands

Users who have overridden internal commands, which is wrong, must adapt according to the following. Or better, they should not override any of such commands and use *reledmac* options instead.

- If you have modified `\Xfootfmt`, note that the fourth argument is now mandatory.
- `\unvxh` has been replaced with `\Xunvxh` and `\unvxhX` with two mandatory arguments.

Appendix A.10 Migration to *reledmac* 2.1.0

Reledmac 2.1.0 fix some bugs when using `\Xbhooknote` and `\bhooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }.`
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` in the argument of `\Xbhooknote/\bhooknoteX`.

Appendix A.11 Migration to *reledmac* 2.1.3

Reledmac 2.1.3 fix an historical bug, (style in *ledmac* 0.7!) which doubled the space before the rules of paragraphed familiar footnotes. Consequently, if you use paragraphed familiar footnotes, you should maybe adapt it, playing with `\beforenotesX`.

Appendix A.12 Migration to *reledmac* 2.3.0

Before *reledmac* 2.3.0, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use optional argument of `\stanza` or `\endverse`.

Appendix A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{\langle...\rangle}` to the use of `\Xlinerangeseparator` or `/` and `\Xendlinerangeseparator` (7.2.4 p. 37).

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.

Appendix A.19 Migration to reledmac 2.18.0

After updating reledmac, and before any new compilation, you need to clean your `.aux` files, if you use `\edlabel` or related.

References

- [Bre96] Herbert Breger. `tabmac`. October 1996. (Available from CTAN in `macros/plain/contrib/tabmac`)
- [Bur01] John Burt. ‘Typesetting critical editions of poetry’. *TUGboat*, **22**, 4, pp 353–361, December 2001. (Code available from CTAN in `macros/latex/contrib/poemscol`)
- [Eck03] Matthias Eckermann. *The Parallel-Package*. April 2003. (Available from CTAN in `macros/latex/contrib/parallel`)
- [Fai03] Robin Fairbairns. *footmisc—a portmanteau package for customising footnotes in L^AT_EX*. February 2003. (Available from CTAN in `macros/latex/contrib/footmisc`)
- [LW90] John Lavagnino and Dominik Wujastyk. ‘An overview of edmac: a PLAIN TeX format for critical editions’. *TUGboat*, **11**, 4, pp. 623–643, November 1990. (Code available from CTAN in `macros/plain/contrib/edmac`)
- [Lüc03] Uwe Lück. ‘ednotes—critical edition typesetting with LaTeX’. *TUGboat*, **24**, 2, pp. 224–236, 2003. (Code available from CTAN in `macros/latex/contrib/ednotes`)
- [Sul92] Wayne G. Sullivan. *The file 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
\@gobblefour	1
\@gobbleseven	1
\@gobblethree	1
\@h	1
\@hangingsymbol	1
\@iiiminipage	1
\@insertstanzaumber	1
\@k	1
\@l@dttempcnta	1
\@l@dttempcntb	1
\@lab	1
\@led@testifnofoot	1
\@lemma	1
\@line@num	1
\@lock	1
\@lopL	1
\@lopR	1
\@makechapterhead	1
\@makeschapterhead	1
\@mem@extranofeet	1
\@mem@old@ssect	1
\@mpfnpos	1
\@msd	1
\@msd@c	1
\@msd@options@iffullpage	1
\@msdata@list	1
\@nl	1
\@nl@reg	1
\@opXfeet	1
\@pend	1
\@pendR	1
\@ref	1
\@ref@reg	1
\@ref@reg@parse	1
\@sect	1
\@series	1
\@set	1
\@sidenotesep	1
\@ssect	1
\@startstanza	1
\@stopmsd	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1

\@xloop	1
\@xympar	1
CLASSarticle	67
CLASSbook	67, 337
CLASSmemoir	194, 195, 254–256, 289, 337, 356, 417, 421
CLASSscrbook	421
COMMAND*footnote	68
COMMAND\...@footnotemark...	198
COMMAND\...d@ta	146
COMMAND\<hook	
@<series	243
COMMAND\<hookname	
<pseudoseries	245
COMMAND\<type	
footfmt	185
COMMAND\@@line	176
COMMAND\@MM	162, 418
COMMAND\@Rlineflag	290, 418
COMMAND\@Serefprefix	268
COMMAND\@Serefprefixmore	268
COMMAND\@add@	327
COMMAND\@adv	106
COMMAND\@apprefprefixmore	268
COMMAND\@apprefprefixsingle	268
COMMAND\@beforeinsertofthisedtext	128
COMMAND\@bsphack	257
COMMAND\@doclearpage	255, 256, 411, 421
COMMAND\@doreinfectX	421
COMMAND\@dprintingcolumns	418
COMMAND\@edindex@hyperref	291, 292
COMMAND\@edtext@level	125
COMMAND\@esphack	257
COMMAND\@fnpos	216, 251
COMMAND\@footnotemark	195, 411, 421
COMMAND\@footnotetext	195, 196, 411
COMMAND\@gobble	123, 124, 238
COMMAND\@gobblefive	420
COMMAND\@gobblefour	417
COMMAND\@gobbleseven	239
COMMAND\@gobblethree	410
COMMAND\@h	179
COMMAND\@hangingsymbol	296
COMMAND\@iiiminipage	280, 282, 410, 421
COMMAND\@iiiminpage	280
COMMAND\@l	416
COMMAND\@l@dtmncnta	148, 150, 157
COMMAND\@l@dtmncntb	150
COMMAND\@l@reg	416
COMMAND\@lab	102, 257, 260, 264, 410
COMMAND\@ldunboxmpfoot	283

COMMAND\@led@extranofeet	357
COMMAND\@ledinnote@command	287
COMMAND\@lemma	128, 130
COMMAND\@lock	96, 296
COMMAND\@lopL	411
COMMAND\@lopR	411
COMMAND\@makecol	251, 252, 254, 255, 421
COMMAND\@mpfnpos	216
COMMAND\@msd	304, 305
COMMAND\@msd@c	304
COMMAND\@msd@options@iffullpage	311
COMMAND\@msdata@list	305, 306
COMMAND\@nl	102–105, 107, 116, 260, 410, 411
COMMAND\@nl@reg	103, 354, 411, 416
COMMAND\@opXfeet	411
COMMAND\@opfeetX	421
COMMAND\@opxtrafeeti	421
COMMAND\@page	104, 260
COMMAND\@pend	411
COMMAND\@pendR	411
COMMAND\@ref	102, 112–114, 117, 118, 123
COMMAND\@ref@later	113, 119
COMMAND\@ref@reg	112, 411
COMMAND\@ref@reg@parsearg	113
COMMAND\@reinserts	251, 253–255, 421
COMMAND\@secondoftwo	69
COMMAND\@sect	339
COMMAND\@series	241
COMMAND\@set	106
COMMAND\@sidenotesep	279
COMMAND\@stopmsd	305
COMMAND\@sw	114, 131, 134, 135
COMMAND\@tag	124, 126, 129
COMMAND\@tempcnta	82
COMMAND\@tempcntb	82
COMMAND\@toksa	88
COMMAND\@toksb	88
COMMAND\@xloop	158
COMMAND\@xympar	272, 421
COMMAND\Aendnote	16, 25
COMMAND\Afootfmt	162
COMMAND\Afootgroup	162
COMMAND\Afootnote	8, 16, 23, 24, 27, 126, 170, 194, 217, 232, 420
COMMAND\Afootstart	162
COMMAND\AtBeginDocument	255
COMMAND\AtEveryPend	18, 50, 140, 418, 419, 421
COMMAND\AtEveryPstart	18, 50, 418, 419, 421, 424
COMMAND\AtEveryStanza	50, 425
COMMAND\AtEveryStopStanza	50, 425
COMMAND\Bendnote	16, 23

COMMAND\Bfootnote	8, 16, 194, 217, 232
COMMAND\Centering	44
COMMAND\Cfootnote	194
COMMAND\Columns	83, 167
COMMAND\Dfootnote	194
COMMAND\Efootnote	194
COMMAND\Gls	61
COMMAND\Hy@raisedlink	349
COMMAND\Hy@raisedlink@left	349
COMMAND\LTR	44
COMMAND\NR	339
COMMAND\Pages	83, 252, 253
COMMAND\ProcessOptionsX	73
COMMAND\RL	43
COMMAND\RaggedLeft	44
COMMAND\RaggedRight	44
COMMAND\SEonlypage	266, 423
COMMAND\SEref	54–56, 266, 269, 424, 426
COMMAND\SErefonlypage	54–56, 360, 423
COMMAND\SErefwithpage	54, 56, 266, 269, 360, 423, 425
COMMAND\Stanza	416
COMMAND\Waklam	328
COMMAND\X@doreinfeet	253, 421
COMMAND\XR@prefix	272
COMMAND\XR@test	272
COMMAND\XR@test@mac	272
COMMAND\XR@test@mac@test	272
COMMAND\XXXXXXfmt	353
COMMAND\XXXXXXfmt	353
COMMAND\Xafterlemmaseparator	41, 357
COMMAND\Xafternote	44, 45, 356, 357
COMMAND\Xafternumber	39, 357
COMMAND\Xafterrule	46, 217, 357, 416, 419
COMMAND\Xaftersymlinenum	39, 357
COMMAND\Xarrangement	35, 45, 68, 163, 164, 243, 357
COMMAND\Xarrangement@footparagraph	169
COMMAND\Xarrangement@normal	164
COMMAND\Xarrangement@paragraph	168
COMMAND\Xbeforeinserting	43, 44
COMMAND\Xbeforelemmaseparator	41, 357
COMMAND\Xbeforenotes	45, 217, 357, 416, 419
COMMAND\Xbeforenumber	37, 39, 357
COMMAND\Xbeforesymlinenum	39, 357
COMMAND\Xbhookendnote	358
COMMAND\Xbhookgroup	45, 423, 424
COMMAND\Xbhooknote	43, 358, 359, 421, 422
COMMAND\Xboxendlinenum	40, 358, 420
COMMAND\Xboxlinenum	39, 40, 358
COMMAND\Xboxlinenumalign	40, 358, 420
COMMAND\Xboxstartlinenum	40, 358, 420

COMMAND\Xboxsymlinenum	39, 40, 358
COMMAND\Xcolalign	44, 419
COMMAND\Xdo@feet	421, 426
COMMAND\Xend	239
COMMAND\XendXafterlemmaseparator	358
COMMAND\XendXbeforelemmaseparator	358
COMMAND\XendXinplaceoflemmaseparator	358
COMMAND\XendXlemmaseparator	358
COMMAND\XendXmorethantwolines	358
COMMAND\XendXtwolines	358
COMMAND\Xendafterenumber	39, 422
COMMAND\Xendafterlemmaseparator	41, 358
COMMAND\Xendafternote	47, 424
COMMAND\Xendafternumber	41
COMMAND\Xendafterpagenumbe	425
COMMAND\Xendafterpagenumber	40, 56
COMMAND\XendafterpagenumberSereonlypage	360
COMMAND\Xendaftersymlinenum	39, 41, 422
COMMAND\Xendahookinplaceofnumber	41, 422
COMMAND\Xendahooklinenumber	41, 422
COMMAND\Xendbeforelemmaseparator	41, 358
COMMAND\Xendbeforelinenumber	40
COMMAND\Xendbeforenumber	39, 422
COMMAND\Xendbeforepagenumber	40, 55, 56
COMMAND\XendbeforepagenumberSereonlypage	55
COMMAND\Xendbeforesymlinenum	39, 41, 422
COMMAND\Xendbhookinplaceofnumber	41, 422
COMMAND\Xendbhooklinenumber	40, 422
COMMAND\Xendbhooknote	43
COMMAND\Xendboxendlinum	40, 420
COMMAND\Xendboxlinenum	40, 358, 418
COMMAND\Xendboxlinenumalign	40, 358, 420
COMMAND\Xendboxstartlinenum	40, 420
COMMAND\Xendboxsymlinenum	40, 422
COMMAND\Xendhangindent	43, 422, 424
COMMAND\Xendinplaceoflemmaseparator	25, 41, 358
COMMAND\Xendinplaceofnumber	39, 421
COMMAND\Xendinplaceofpagenumber	37, 426
COMMAND\Xendinsertsep@	223
COMMAND\Xendlemmadisablefontselection	42
COMMAND\Xendlemmafont	42, 423
COMMAND\Xendlemmaseparator	25, 41, 358
COMMAND\Xendletboxendlinum	358
COMMAND\Xendlineflag	56
COMMAND\Xendlineprefixmore	40, 56
COMMAND\Xendlineprefixsingle	40, 56
COMMAND\Xendlinangeseparator	37, 56, 183, 360, 422
COMMAND\Xendmorethantwolines	25, 37, 56, 358, 419, 420
COMMAND\Xendnonumber	38, 421
COMMAND\Xendnote	220, 238, 239, 419

COMMAND\Xendnotefontsize	42, 357
COMMAND\Xendnotenumfont	40–42, 357
COMMAND\Xendnotes	223
COMMAND\Xendnumberonlyfirstinline	36, 422
COMMAND\Xendnumberonlyfirstintwolines	36, 422
COMMAND\Xendpagenumberonlyfirst	36, 426
COMMAND\Xendpagenumberonlyfirstifsingle	36, 426
COMMAND\Xendpagenumberonlyfirstintwo	36, 426
COMMAND\Xendparagraph	47, 416
COMMAND\Xendsep	47
COMMAND\Xendsublinesep	38, 56, 183
COMMAND\Xendsymlinenum	36, 422
COMMAND\Xendsympagenum	37, 426
COMMAND\Xendtwolines	25, 37, 56, 358, 419, 420
COMMAND\Xendtwolinesbutnotmore	37, 56, 419, 420
COMMAND\Xendtwolinesonlyinsamepage	38, 56, 419, 420
COMMAND\Xendwrapcontent	43, 425
COMMAND\Xendwraplemma	43, 425
COMMAND\Xfootfmt	359
COMMAND\Xfootgroup	168
COMMAND\Xfootins	167
COMMAND\Xfootnote	53, 59, 124, 353, 413, 417, 419, 423, 425
COMMAND\Xfootstarts	168
COMMAND\Xgroupbyline	45, 156, 193
COMMAND\Xgroupbylines	426
COMMAND\Xgroupbylineseparetwolines	45
COMMAND\Xhangindent	43, 422
COMMAND\Xhsize	360, 423, 424
COMMAND\Xsizethreecol	44, 47, 358
COMMAND\Xsizetwocol	44, 47, 244, 358
COMMAND\Xinplaceoflemmaseparator	24, 41, 358
COMMAND\Xinplaceofnumber	39, 358, 419, 420
COMMAND\Xinsertparafootsep	173, 175
COMMAND\Xledsetnormalparstuff	355, 356, 419
COMMAND\Xlemmadisablefontselection	42
COMMAND\Xlemmafont	42, 423
COMMAND\Xlemmaseparator	41, 183, 246, 248, 251, 358
COMMAND\Xlineflag	55, 360, 361, 425
COMMAND\Xlinerangeseparator	37, 56, 183, 360, 422
COMMAND\Xmaxhnotes	46, 67, 68, 357, 358, 416, 418
COMMAND\Xmorethantwolines	24, 37, 38, 56, 358, 418
COMMAND\Xnoindent	422
COMMAND\Xnolemmaseparator	41, 251, 358
COMMAND\Xnonbreakableafternumber	39, 358, 414
COMMAND\Xnonumber	38, 358
COMMAND\Xnonumberinfootnote	358
COMMAND\Xnotefontsize	42, 357
COMMAND\Xnotefontsize@(<s>)	173, 177, 179
COMMAND\Xnotenumfont	42, 357
COMMAND\Xnoteswidthliketwocolumns	47, 417

COMMAND\Xnumberonlyfirstinline	36, 37, 45, 99, 184, 245, 246, 248, 358, 413, 418, 427
COMMAND\Xnumberonlyfirstintwolines	36, 45, 358, 413, 427
COMMAND\Xonlypstart	38, 358, 413, 418
COMMAND\Xpagelinesep	39, 426
COMMAND\Xparafootsep	45, 99, 357, 358, 425, 427
COMMAND\Xparafootsep@series	173
COMMAND\Xparinden	360
COMMAND\Xparindent	43, 419, 422, 424
COMMAND\Xprenotes	46, 217, 425
COMMAND\Xprenotes@	167, 217, 413
COMMAND\Xpstart	38, 355, 358, 413, 418
COMMAND\Xpstarteverytime	38, 355, 358, 418
COMMAND\Xragged	45
COMMAND\Xrigidbalance	176, 360, 423
COMMAND\Xstanza	38, 51
COMMAND\Xstanzaseparator	38
COMMAND\Xstorelineinfo	184
COMMAND\Xsublinesep	21, 38, 39, 56, 183, 360
COMMAND\Xsublinesepside	21, 38
COMMAND\Xsymlinenum	36, 45, 357, 358, 420, 427
COMMAND\Xtoendnotes	26, 239
COMMAND\Xtwolines	24, 37, 38, 56, 189, 190, 245, 358, 418
COMMAND\Xtwolinesappref	245
COMMAND\Xtwolinesbutnotmore	37, 38, 56, 358, 419
COMMAND\Xtwolinesbutnotmoreappref	245
COMMAND\Xtwolinesonlyinsamepage	37, 38, 56, 358, 419
COMMAND\Xtxtbeforenotes	45, 358, 425, 426
COMMAND\Xunvxh	171, 359
COMMAND\Xwidth	47, 360, 424
COMMAND\Xwrapcontent	43, 425
COMMAND\Xwraplemma	42–44, 425
COMMAND\&	357
COMMAND\<XXX>vfootnote	193
COMMAND\absline@num	96, 147
COMMAND\accent	123
COMMAND\actionlines@list	97, 148
COMMAND\actions@list	97
COMMAND\add@Xgroupbyline	156
COMMAND\add@inserts	97, 155, 156
COMMAND\add@inserts@next	155, 156
COMMAND\add@msd@	304
COMMAND\add@msdata	305, 306
COMMAND\add@msdata@firstlineofpage	308
COMMAND\add@msddata	304
COMMAND\add@penalties	147, 157
COMMAND\addcontentsline	335
COMMAND\addfootins	353, 356
COMMAND\addfootinsX	353, 356
COMMAND\addtoendnotes	239
COMMAND\advancelabel@refs	259

COMMAND\advanceline	22, 98, 106, 120, 421
COMMAND\affixlin@num	279
COMMAND\affixline@num	150, 152, 154, 411
COMMAND\affixpstart@num	154
COMMAND\afterXrule	357
COMMAND\afterXsymlinenum	357
COMMAND\afterenumber	39
COMMAND\aftergroup	123, 127
COMMAND\afterlemmaseparator	357
COMMAND\afternote	357
COMMAND\afternoteX	45, 356, 357
COMMAND\afternumberinfootnote	357
COMMAND\afterruleX	46, 416, 419
COMMAND\applabel	55, 261, 262, 419, 425
COMMAND\appref	53, 55, 56, 266, 269, 423, 424
COMMAND\apprefprefixmore	55, 357
COMMAND\apprefprefixsingle	55, 357
COMMAND\apprefwithpage	55, 56, 266, 269, 420, 423
COMMAND\arrangementX	35, 68, 198, 243, 357
COMMAND\arrangementX@normal	204
COMMAND\article	15
COMMAND\at@every@pend	140
COMMAND\autopar	17, 18, 138, 141, 142, 214, 412, 414, 415, 419
COMMAND\ballast	68
COMMAND\ballast@count	147, 157
COMMAND\baselineskip	35, 169, 173
COMMAND\beforeXnotes	357
COMMAND\beforeXsymlinenum	357
COMMAND\beforeelectedchapter	10, 65, 336
COMMAND\beforeinsertingX	43
COMMAND\beforelemmaseparator	357
COMMAND\beforenotesX	46, 359, 415, 416, 419
COMMAND\beforenumberinfootnote	357
COMMAND\begin	312
COMMAND\beginnumbering	16–19, 83, 84, 86, 95, 100, 115, 141, 219, 304, 413, 416, 420, 421, 426, 427
COMMAND\bf	413
COMMAND\bfseries	42, 413
COMMAND\bhookXnote	358
COMMAND\bhookgroupX	45, 423
COMMAND\bhooknoteX	43, 359, 421, 422
COMMAND\body	296
COMMAND\bodyfootmarkA	32
COMMAND\book	15
COMMAND\boxXendlinenum	358
COMMAND\boxXendlinenumalign	358
COMMAND\boxXendstartlinenum	358
COMMAND\boxendlinenum	358
COMMAND\boxlinefootnote	187
COMMAND\boxlinenum	358
COMMAND\boxlinenumalign	358

COMMAND\boxstartlinenum	358
COMMAND\boxsymlinenum	358
COMMAND\break	35, 171
COMMAND\brokenpenalty	157
COMMAND\centering	44
COMMAND\ch@ck@l@ck	411
COMMAND\ch@cksub@l@ck	152, 411
COMMAND\chapter	64, 337, 341, 416, 419, 421, 425
COMMAND\chaptermark	335
COMMAND\check@pb@in@verse	348
COMMAND\colalignX	44, 419
COMMAND\collect@body	313
COMMAND\color	425
COMMAND\colorbox	69
COMMAND\columns	47
COMMAND\columnwidth	169, 417
COMMAND\command names	244, 245
COMMAND\copyright	124
COMMAND\correct@Xfootins@box	418
COMMAND\correct@footinsX@box	418
COMMAND\count	177
COMMAND\critex	412
COMMAND\critext	130, 352, 353, 356
COMMAND\csname	74, 133
COMMAND\ctab	328, 333
COMMAND\ctabtext	333
COMMAND\dc col	322
COMMAND\def	71
COMMAND\detokenize	133
COMMAND\dimen	177
COMMAND\dimexpr	47
COMMAND\discretionary	171
COMMAND\displaywidowpenalty	157
COMMAND\do@Xfeet	252, 411, 421, 426
COMMAND\do@actions	147, 149, 411
COMMAND\do@actions@fixedcode	411
COMMAND\do@actions@next	147, 148
COMMAND\do@ballast	147, 157
COMMAND\do@feet@custom@order	251
COMMAND\do@insidelinehook	414
COMMAND\do@line	97, 122, 139, 142, 145, 146, 155, 157, 296, 411, 412, 414, 416
COMMAND\do@linehook	411
COMMAND\do@lockoff	98
COMMAND\do@lockon	98
COMMAND\dodoreinextrafeet	410
COMMAND\doendnotes	25, 223, 224, 359, 420, 427
COMMAND\doendnotesbysection	25, 224, 240, 420, 427
COMMAND\doennotes	427
COMMAND\doinsidelinehook	23, 417
COMMAND\dolinehook	23, 417

COMMAND\doreintrafeeti	421
COMMAND\doreintrafeetii	421
COMMAND\doxtrafeet	251, 410
COMMAND\doxtrafeeti	421
COMMAND\doxtrafeetii	421
COMMAND\dummy@ref	123
COMMAND\edaftertab	63, 328, 329
COMMAND\edatleft	63, 326
COMMAND\edatright	63, 64, 326
COMMAND\edbeforetab	63, 328
COMMAND\edfilldimen	327
COMMAND\edfont@info	129
COMMAND\edgls	61, 286
COMMAND\edglsadd	427
COMMAND\edgls...	424
COMMAND\edindex	59–61, 285, 286, 289, 291, 292, 317, 414, 417, 418, 421, 422, 426, 427
COMMAND\edindexlab	60
COMMAND\edlabel	52–55, 57, 124, 257, 259, 260, 263, 264, 271, 286, 317, 349, 361, 410, 413–415, 418, 423
COMMAND\edlabelE	54, 262
COMMAND\edlabelS	54, 262
COMMAND\edlabelSE	54
COMMAND\edlineref	52, 257, 358, 418, 420, 423, 427
COMMAND\edmakelabel	54, 271
COMMAND\edpageref	52, 257, 263, 271
COMMAND\edrowfill	328
COMMAND\edsublineref	52
COMMAND\edtabcolsep	321
COMMAND\edtext	6, 23, 24, 26, 27, 29, 30, 48, 52, 53, 55, 59, 61, 68, 69, 97, 112, 114, 117, 118, 121–130, 132– 134, 136, 261, 262, 265, 316, 318, 337, 349, 352, 353, 356, 410, 412, 414, 416–420, 425–427
COMMAND\edtext@level	420
COMMAND\edtextlater	114
COMMAND\edvertdots	64, 327
COMMAND\edvertline	64, 326
COMMAND\eledchapter	65
COMMAND\eled@sectioning@out	342
COMMAND\eledchapter	64, 356, 417, 421
COMMAND\eledchapter*	64
COMMAND\eledmac@error	410
COMMAND\eledsection	6, 16, 65, 123, 145, 336, 356, 419
COMMAND\eledsection*	65
COMMAND\eledsubsection	65, 356
COMMAND\eledsubsection*	65
COMMAND\eledsubsubsection	65, 356
COMMAND\eledsubsubsection*	65
COMMAND\eledxxx	10, 65, 342, 416
COMMAND\eledxxxx	335
COMMAND\eledsection	342
COMMAND\else	285, 336

COMMAND\empty	82, 151, 257
COMMAND\end	312
COMMAND\end@lemmas	123
COMMAND\endashchar	183
COMMAND\endgraf	139, 173, 214
COMMAND\endlock	21, 98, 121, 301
COMMAND\endminipage	280, 282, 410, 421
COMMAND\endmsdata	32
COMMAND\endnotes	419, 423
COMMAND\endnumbering	16, 17, 19, 83, 84, 86, 411, 420, 426, 427
COMMAND\endprint	220, 222, 239, 355
COMMAND\endstanzaextra	357
COMMAND\endsub	21, 98, 119
COMMAND\endverse	359
COMMAND\everypar	141
COMMAND\extensionchars	66, 83
COMMAND\externaldocument	57, 272
COMMAND\f@x@l@cks	411
COMMAND\falseverse	356, 414, 416
COMMAND\fi	336
COMMAND\firstlinenum	20, 150, 412
COMMAND\firstsublinenum	20, 412
COMMAND\fix@page	103, 104, 411
COMMAND\flag@end	117, 118, 129, 416
COMMAND\flag@end@RTL	118
COMMAND\flag@end@later	119
COMMAND\flag@start	117, 118, 129, 416, 417
COMMAND\flag@start@RTL	118
COMMAND\flag@start@later	119
COMMAND\flagstanza	51
COMMAND\floatingpenalty	162, 418
COMMAND\flush@notes	158
COMMAND\fnpos	34, 216, 415, 426
COMMAND\footfmt	162, 164
COMMAND\footfmt...	199
COMMAND\footfootmarkA	33
COMMAND\footfudgefactor	171
COMMAND\footfudgefiddle	68, 169, 410
COMMAND\footgroup	162
COMMAND\footins	167
COMMAND\footnormal	244, 357, 410
COMMAND\footnormalX	357
COMMAND\footnote	32, 68, 194–196, 354, 411
COMMAND\footnote@lang	183
COMMAND\footnoteA	16, 32
COMMAND\footnoteB	16
COMMAND\footnoteC	23
COMMAND\footnoteE	32
COMMAND\footnoteX	8, 236, 238, 425
COMMAND\footnoteX@reading	237

COMMAND\footnoteXmk	250
COMMAND\footnote⟨X⟩	124
COMMAND\footnotelang@lua	161
COMMAND\footnotelang@poly	161
COMMAND\footnoteoption@	160, 422
COMMAND\footnoterule	177
COMMAND\footnotesize	42
COMMAND\footparagraph	169, 244, 357, 416
COMMAND\footparagraphX	209, 357, 416
COMMAND\footsplitskips	411, 418
COMMAND\footstart	162, 167, 177
COMMAND\footstrut	173
COMMAND\footthreecol	357
COMMAND\footthreecolX	357, 420
COMMAND\foottwocol	357
COMMAND\foottwocolX	357, 420
COMMAND\foreignlanguage	43
COMMAND\fullstop	360
COMMAND\get@edindex@hyperref	291
COMMAND\get@edindex@ledinnote@command	287
COMMAND\get@fnmark	196
COMMAND\get@index@command	415
COMMAND\get@linelistfile	411
COMMAND\get@thisfootnote	202
COMMAND\getline@num	146, 148
COMMAND\gl@p	88
COMMAND\global	102
COMMAND\globaldefs	102
COMMAND\gls	61, 293
COMMAND\hangindentX	43, 419, 422
COMMAND\hangingsymbol	357, 412
COMMAND\hbox	171
COMMAND\hfill	415
COMMAND\hidenumbering	22, 111, 419
COMMAND\hidenumberingonleftpage	23, 111, 425
COMMAND\hidenumberingonrightpage	23, 111, 425
COMMAND\hline	61
COMMAND\hrulefill	328
COMMAND\hsize	35, 167, 169–171, 178, 180, 215, 411, 417
COMMAND\hsizeX	360, 423, 424
COMMAND\hsizethreecol	358
COMMAND\hsizethreecolX	44, 47
COMMAND\hsizetwocol	358
COMMAND\hsizetwocolX	44, 47
COMMAND\hyperlinkR	290
COMMAND\hyperlinkformat	290
COMMAND\hyperlinkformatR	290
COMMAND\if@RTL	75, 127
COMMAND\if@edtext@	417, 420
COMMAND\if@eled@sectioning	342

COMMAND\if@firstlineofpage	75
COMMAND\if@firstlineofpageR	75
COMMAND\if@msd@options@fullpage	311
COMMAND\if@msdata@insertedfrompreviouspage	308
COMMAND\if@nobreak	140
COMMAND\if@noneed@Footnote	117
COMMAND\ifXnote@	83
COMMAND\ifbypage@	89
COMMAND\ifbypage@R	89
COMMAND\ifbypstart@	89
COMMAND\ifbypstart@R	89
COMMAND\iffirst@linenum@out@	115, 116
COMMAND\ifindtl@innote	83
COMMAND\ifindtl@notenumber	83
COMMAND\ifinserthangingsymbol	296
COMMAND\ifinstanza	296
COMMAND\ifistwofollowinglines	190, 191
COMMAND\ifl@d@Xmorethantwolines	188, 418
COMMAND\ifl@d@Xtwolines	188
COMMAND\ifl@d@dash	188
COMMAND\ifl@d@elin	188
COMMAND\ifl@d@esl	188
COMMAND\ifl@d@pnum	187
COMMAND\ifl@d@ssub	188
COMMAND\ifl@dend@X	238
COMMAND\ifl@dmemoir	410
COMMAND\ifl@dpaging	417
COMMAND\ifl@dpairing	82, 412
COMMAND\ifl@dprintingpages	418
COMMAND\ifl@dskipnumber	150
COMMAND\ifl@dstartendok	327
COMMAND\ifl@imakeidx	75
COMMAND\ifledRcol	83, 412
COMMAND\ifledRcol@	83, 416
COMMAND\iflemmacommand@	417
COMMAND\ifnoend@	224
COMMAND\ifnoledgroup@	285
COMMAND\ifnoteschanged@	99
COMMAND\ifnumberedpar@	137
COMMAND\ifnumbering	84, 86
COMMAND\ifnumberingR	83, 412
COMMAND\ifnumberline	128, 150
COMMAND\ifpst@rted	412
COMMAND\ifpst@rtedL	84
COMMAND\ifseriesbefore	242
COMMAND\ifstopmsdata@inserted@	304
COMMAND\ifsublines@	96, 108
COMMAND\iftrue	420
COMMAND\ifvmode	259
COMMAND\ifxxx	336

COMMAND\ignorespaces	126
COMMAND\imki@wrindexentry	75
COMMAND\immediate	115, 116, 219
COMMAND\indent	18, 141, 359
COMMAND\index	293, 426
COMMAND\indtl@wrindexentry	75
COMMAND\initnumbering@quote	334, 421
COMMAND\initnumbering@reg	411
COMMAND\initnumbering@sectcmd	421
COMMAND\inplaceoflemmaseparator	358
COMMAND\inplaceofnumber	358
COMMAND\insert	128, 155, 162, 164, 165, 178, 193, 199
COMMAND\insert@Xtxtbeforenotes	159, 178
COMMAND\insert@count	112, 117, 126
COMMAND\insert@countR	126
COMMAND\insert@msdata	305, 311
COMMAND\inserthangingsymbol	415
COMMAND\insertlines@list	97, 112
COMMAND\insertparafootsepX	212
COMMAND\inserts@list	122, 138, 155, 170
COMMAND\interAfootnotelinepenalty	412
COMMAND\interfootnotelinepenalty	412
COMMAND\interlinepenalty	162
COMMAND\interparanoteglue	356
COMMAND\justifying	44
COMMAND\l@advance@parledegroupp@beforenormalnotes	421
COMMAND\l@d@@@wrindexhyp	417
COMMAND\l@d@add	131
COMMAND\l@d@end	219, 238
COMMAND\l@d@nums	126, 129–131, 187, 188
COMMAND\l@d@section	220
COMMAND\l@d@set	107, 120
COMMAND\l@dampcount	318
COMMAND\l@dbfnote	196, 411
COMMAND\l@dcheckstartend	327
COMMAND\l@dchset@num	107
COMMAND\l@dcolcount	318, 319
COMMAND\l@dcollect@@body	312
COMMAND\l@dcollect@body	312
COMMAND\l@dcsnote	416
COMMAND\l@dcsnotetext	146, 276
COMMAND\l@dcsnotetext@l	146, 276, 277
COMMAND\l@dcsnotetext@r	146, 276
COMMAND\l@ddodoreinextrafeet	253, 410
COMMAND\l@ddoxtrafeet	253, 410
COMMAND\l@emptyd@ta	412
COMMAND\l@dend@close	219
COMMAND\l@dend@open	219
COMMAND\l@dend@stuff	219
COMMAND\l@denvbody	312

COMMAND\l@dfeetbeginmini	410
COMMAND\l@dfeetendmini	410
COMMAND\l@dgetline@margin	412
COMMAND\l@dgetlock@disp	412
COMMAND\l@dgetref@num	264, 265
COMMAND\l@dgetsidenote@margin	273, 412
COMMAND\l@dgobbeloptarg	417
COMMAND\l@dgobblearg	417
COMMAND\l@dgobbleoptarg	317
COMMAND\l@dlabel@parse	264, 265
COMMAND\l@dld@ta	150, 152
COMMAND\l@dlp@rbox	278
COMMAND\l@dlsn@te	412
COMMAND\l@dlsnote	416
COMMAND\l@dmake@labels	260, 272
COMMAND\l@dmake@labelsR	272
COMMAND\l@dnumpsstartsL	84, 412
COMMAND\l@dp@rsefootspec	188
COMMAND\l@dparsefootspec	188
COMMAND\l@dpush@begins	312
COMMAND\l@drd@ta	150, 152
COMMAND\l@dref@undefined	264
COMMAND\l@drsn@te	412
COMMAND\l@drsnote	416
COMMAND\l@dtabaddcols	327
COMMAND\l@dtabnoexpands	410
COMMAND\l@dumboxmpfoot	421
COMMAND\l@dunboxmpfoot	412
COMMAND\l@dzeropenalties	412, 417
COMMAND\l@pb	348
COMMAND\l@prev@nopb	347
COMMAND\l@prev@pb	347
COMMAND\l@reg	354
COMMAND\label	19, 54, 57, 60, 257, 258, 265
COMMAND\label@refs	257
COMMAND\labelstarttrue	19, 413
COMMAND\labelref@list	257, 260
COMMAND\language	171
COMMAND\last@page@num	411
COMMAND\lastbox	141
COMMAND\lastskip	119
COMMAND\latex@makecol	255
COMMAND\leavevmode	18, 141
COMMAND\led@check@nopb	348
COMMAND\led@check@pb	348
COMMAND\led@nopb	347, 348
COMMAND\led@nopbnum	347
COMMAND\led@pb	347, 348
COMMAND\led@pb@macro	347
COMMAND\led@pbnum	347

COMMAND\led@reinit@index@fornote	293
COMMAND\led@set@index@fornote	293
COMMAND\ledRflag	290
COMMAND\ledchapter	356, 414
COMMAND\ledfootinsdim	357
COMMAND\ledinnernote	57, 274, 416, 425
COMMAND\ledinnote	288, 355, 420
COMMAND\ledinnotemark	59, 355, 419
COMMAND\ledleftnote	57, 274
COMMAND\ledlinenum	94, 412
COMMAND\ledllfill	146
COMMAND\ledlsnotefontsetup	425
COMMAND\ledlsnotesep	58
COMMAND\ledlsnotewidth	57
COMMAND\lednob	66, 347
COMMAND\lednobinverse	348
COMMAND\lednobinverse>true	50, 66
COMMAND\lednobnum	347
COMMAND\ledouternote	57, 274, 416, 425
COMMAND\ledpb	66, 347, 356
COMMAND\ledpbnum	347
COMMAND\ledpbsetting	66, 347, 422
COMMAND\ledrightnote	57, 274
COMMAND\ledrsnotefontsetup	425
COMMAND\ledrsnotesep	58
COMMAND\ledrsnotewidth	57
COMMAND\ledsection	356
COMMAND\ledsectnomark	335
COMMAND\ledsectnotoc	335
COMMAND\ledsetnormalparstuff	355, 356, 419
COMMAND\ledsetnormalparstuff@common	214
COMMAND\ledsetnormalparstuffX	355, 356, 419
COMMAND\ledsidenote	57, 274, 276, 277
COMMAND\ledsubsection	356
COMMAND\ledsubsubsection	356
COMMAND\ledxxx	416
COMMAND\left	63
COMMAND\leftctab	328
COMMAND\leftheadline	94
COMMAND\leftlinenum	21, 94, 410, 412
COMMAND\leftltab	328
COMMAND\leftnoteupfalse	58
COMMAND\leftpstartnum	154
COMMAND\leftftab	328
COMMAND\leftsidenote	276
COMMAND\leftskip	167, 170, 171
COMMAND\lemma	3, 24, 26, 27, 29, 30, 122, 125–127, 129, 130, 132, 352, 412, 413, 420, 421, 423
COMMAND\lemmaseparator	358
COMMAND\let	27, 48, 124, 128, 238, 255, 301, 410
COMMAND\letboxXendendlinenum	358

COMMAND\line	176, 179
COMMAND\line@list	97, 113, 129
COMMAND\line@list@stuff	84, 100, 115, 410, 412
COMMAND\line@list@version	102
COMMAND\line@margin	90, 151, 273
COMMAND\line@num	95, 96, 98, 150, 410
COMMAND\line@set	130, 131
COMMAND\lineation	20, 89
COMMAND\linebreak	35
COMMAND\linenum	24, 27, 53, 55, 122, 130, 263, 265, 271, 352, 424
COMMAND\linenum@out	115, 257, 260
COMMAND\linenumberlist	20, 82, 151, 410
COMMAND\linenumberstyle	22, 94, 410, 427
COMMAND\linenumincrement	20, 412
COMMAND\linenummargin	20, 21, 90, 273
COMMAND\linenumr@p	94, 410, 412
COMMAND\linenumrep	94, 412
COMMAND\linenumsep	21, 58, 94, 274
COMMAND\linerrangesep@	250
COMMAND\lineref	257, 263, 271, 358, 418
COMMAND\list@clear	88
COMMAND\list@clearing@reg	412
COMMAND\list@create	88
COMMAND\lock@disp	93
COMMAND\lock@off	109
COMMAND\lock@on	108
COMMAND\lockdisp	22, 93
COMMAND\loop	158, 296
COMMAND\ltab	328, 329, 333
COMMAND\ltabtext	333
COMMAND\m@mmf@prepare	195
COMMAND\makeatletter	146
COMMAND\makehboxoffhboxes	172, 173
COMMAND\makeindex	59, 289
COMMAND\makelabel	271
COMMAND\managestanza@modulo	297
COMMAND\marginpar	57, 68, 272, 273, 411
COMMAND\marginparwidth	57, 274
COMMAND\markboth	145
COMMAND\mathchardef	297
COMMAND\maxhXnotes	358
COMMAND\maxhnotesX	46, 68, 357, 415, 416, 418–420
COMMAND\maxlinesinpar@list	100
COMMAND\measurebody	330
COMMAND\measuretbody	330
COMMAND\memorybreak	19
COMMAND\morenoexpands	68, 69, 122, 124
COMMAND\morethantwolines	358
COMMAND\mpfnpos	34, 216, 415, 426
COMMAND\mpnormalfootgroup	411

COMMAND\mpnormalvfootnote	411
COMMAND\msdata	31, 32, 302, 304, 305, 427
COMMAND\msdataposition	32
COMMAND\multfootsep	33, 195
COMMAND\multiplefootnotemark	195
COMMAND\musixtex	416
COMMAND\n@num	412, 419
COMMAND\n@num@ref	419
COMMAND\new@line	116, 411
COMMAND\new@series	124
COMMAND\newcommand	27, 71, 194, 260
COMMAND\newcommandx	27
COMMAND\newhookarg@specific	250
COMMAND\newhookcommand@series	244, 246, 419
COMMAND\newhookcommand@series@reload	246
COMMAND\newhookcommand@toggle@reload	245, 417
COMMAND\newhooktoggle@series	245, 419
COMMAND\newhooktoggle@specific	250
COMMAND\newif	419
COMMAND\newline	35
COMMAND\newlinechar	238
COMMAND\newseries	33, 353, 356, 357
COMMAND\newseries@	230, 243
COMMAND\newverse	50, 51, 356, 416
COMMAND\next	296
COMMAND\next@action	101
COMMAND\next@actionline	101
COMMAND\next@insert	155
COMMAND\nl@regR	103
COMMAND\no@expands	68, 124, 129, 238, 410
COMMAND\noXlemmaseparator	358
COMMAND\nobreak	187
COMMAND\nocritical	231
COMMAND\noeledsec	65, 356
COMMAND\noendnotes	356
COMMAND\noexpand	354
COMMAND\nofamiliar	247
COMMAND\noindent	18, 141, 422
COMMAND\noindentX	422
COMMAND\nomk@	250
COMMAND\nonbreakableafternumber	358
COMMAND\nonumberinfootnote	358
COMMAND\normal@footnotemarkX	199
COMMAND\normal@page@break	347
COMMAND\normal@pars	214
COMMAND\normalbfnoteX	412
COMMAND\normalbodyfootmarkX	199
COMMAND\normalfont	426
COMMAND\normalfootfmt	48, 166, 173, 183, 220, 424
COMMAND\normalfootfmtX	200

COMMAND\normalfootfootmarkX	200
COMMAND\normalfootgroup	168
COMMAND\normalfootgroupX	201
COMMAND\normalfootnoterule	163
COMMAND\normalfootstart	166, 170
COMMAND\normalfootstartX	200
COMMAND\normalvfootnote	164, 165
COMMAND\normalvfootnote@inserted	165
COMMAND\normalvfootnoteX	199
COMMAND\notbool	336
COMMAND\notfontsetup	357
COMMAND\notfontsizeX	42, 357
COMMAND\notenumfont	357
COMMAND\notenumfontX	42, 357
COMMAND\notesXwidthliketwocolumns	358
COMMAND\noteswidthliketwocolumnsX	47, 358, 417, 419
COMMAND\num@lines	137, 157
COMMAND\numberlinefalse	20
COMMAND\numberlinetrue	20
COMMAND\numberonlyfirstinline	242, 358
COMMAND\numberonlyfirstintwolines	358
COMMAND\numberpstartfalse	18
COMMAND\numberpstarttrue	18, 38, 355, 413, 421, 427
COMMAND\numberstanza	38
COMMAND\numberstanzafalse	51
COMMAND\numberstanzatrue	51
COMMAND\numlabfont	21, 47, 94
COMMAND\one@line	137
COMMAND\onehalfspacing	422
COMMAND\onlyXpstart	358
COMMAND\onlysideX	237
COMMAND\page@action	98, 107
COMMAND\page@start	98, 412
COMMAND\pagecontents	98
COMMAND\pageparbreak	356
COMMAND\pageref	54, 263
COMMAND\par	26, 35, 141, 214
COMMAND\par@line	137, 157
COMMAND\para@footgroup	170
COMMAND\para@footgroupX	212
COMMAND\para@footsetup	169, 410
COMMAND\para@footsetupX	209, 410, 417
COMMAND\para@vfootnoteX	210
COMMAND\parafootfmt	172, 173, 424
COMMAND\parafootfmtX	211
COMMAND\parafootftm	175
COMMAND\parafootftmX	212
COMMAND\parafootftmsep	357
COMMAND\parafootsep	358, 415, 420
COMMAND\parafootsepX	45, 99, 357, 358, 425

COMMAND\parafootstart	170
COMMAND\parafootstartX	210
COMMAND\paravfootnote	170, 173
COMMAND\parfillskip	173
COMMAND\parindent	422
COMMAND\parindentX	43, 360, 422, 424
COMMAND\parshape	68
COMMAND\parskip	141
COMMAND\pausenumbering	19, 86, 87, 100, 102, 142, 415, 417, 424
COMMAND\penalty	173
COMMAND\pend	2, 6, 17–19, 21, 22, 65, 120, 122, 125, 131, 137–142, 154, 155, 354, 415, 416, 425, 426
COMMAND\preXnotes	419, 425
COMMAND\prenotesX	46, 218, 419
COMMAND\prepare@Xgroupbyline	193
COMMAND\prepare@Xprenotes	217
COMMAND\prev@nopb	347
COMMAND\prev@pb	347
COMMAND\prevlineX	99
COMMAND\prevpageX@num	99
COMMAND\print@Xfootnoterule	419
COMMAND\print@Xnotes	252, 253
COMMAND\print@Xnotes@forpages	418
COMMAND\print@eledsection	145
COMMAND\print@footnoteXrule	419
COMMAND\print@leftmargin@eledsection	336
COMMAND\print@lemma	184
COMMAND\print@line	143
COMMAND\print@notesX@forpages	418
COMMAND\print@rightmargin@eledsection	336
COMMAND\printendlines	224, 230, 269, 360, 410, 412
COMMAND\printlinefootnote	185, 186, 418
COMMAND\printlinefootnotearea	186, 187, 418
COMMAND\printlinefootnotenumbers	185
COMMAND\printlines	166, 183, 187, 189, 224, 269, 360, 410, 412, 418, 423
COMMAND\printnpnum	355, 360
COMMAND\printpstart	183
COMMAND\protect	124, 354
COMMAND\providecommand	194, 410
COMMAND\pstart	2, 6, 17–19, 21, 22, 64, 65, 107, 120, 125, 131, 137–141, 145, 155, 341, 354, 412–414, 416, 417, 419–421, 424–426
COMMAND\pstartinfootnote	358
COMMAND\pstartinfootnoteeverytime	358
COMMAND\pstartnum	154
COMMAND\pstartref	52, 257, 264, 415, 427
COMMAND\pstarts	413
COMMAND\raggedX	45
COMMAND\raggedleft	44
COMMAND\raggedright	44
COMMAND\raw@text	137, 138
COMMAND\rbracket	41

COMMAND\read@linelist	100–102
COMMAND\ref	54, 57, 60
COMMAND\reformatted@	269
COMMAND\reformattedwithpage	269
COMMAND\relax	18, 107, 128, 147, 155, 301, 317, 354
COMMAND\renewcommand	68, 357, 360
COMMAND\reset@msd@options@iffullpage	311
COMMAND\resetprevline@	99
COMMAND\resetprevpage@	99
COMMAND\resumenumbering	19, 83, 86, 87, 100, 102, 142, 412, 416, 417, 424
COMMAND\right	63
COMMAND\rightctab	328
COMMAND\rightlinenum	21, 94, 410, 412
COMMAND\rightltab	329
COMMAND\rightnoteupfalse	58
COMMAND\rightrtab	329
COMMAND\rightrightnote	276
COMMAND\rightrightskip	167, 170, 171, 173
COMMAND\rightrightstartnum	154
COMMAND\rigidbalance	175–177, 179, 360, 423
COMMAND\rigidbalanceX	176, 360, 423
COMMAND\robustify	36
COMMAND\roman	322, 423
COMMAND\rtab	328, 329, 333
COMMAND\rtabtext	330, 333
COMMAND\sameword	28–30, 114, 131–133, 136, 418, 420, 422, 425
COMMAND\sameword@inedtext	132, 133
COMMAND\saweword	132
COMMAND\scriptsize	94
COMMAND\section	64, 412, 426
COMMAND\section@num	83
COMMAND\sectionmark	335
COMMAND\select@lemmafont	47, 48, 160
COMMAND\series	230
COMMAND\series@	230
COMMAND\seriesatbegin	33, 241, 419
COMMAND\seriesatend	33, 242, 419
COMMAND\set@Xtxtbeforenotes	159
COMMAND\set@line	129
COMMAND\set@line@action	98, 108
COMMAND\setSErefonlypageprefixmore	55, 268, 424, 427
COMMAND\setSErefonlypageprefixsingle	55, 268, 424, 427
COMMAND\setSErefprefixmore	55
COMMAND\setSErefprefixsingle	55
COMMAND\setapprefprefixmore	55, 357
COMMAND\setapprefprefixsingle	55, 357, 423
COMMAND\setcommand@series	244
COMMAND\sethangingsymbol	50, 296, 357, 422
COMMAND\sethanginsymbol	48
COMMAND\setistwofollowinglines	190

COMMAND\setl@dlprbox	278
COMMAND\setline	22, 98, 103, 106, 120, 124, 139, 421
COMMAND\setlinenum	22, 103, 107, 120, 410
COMMAND\setmsdatalabel	32
COMMAND\setmsdataposition	427
COMMAND\setmsdataseries	32
COMMAND\setprintendlines	225, 226, 412
COMMAND\setprintlines	189, 190, 225, 412
COMMAND\setsidenoteseq	58
COMMAND\setsidenotsep	357
COMMAND\setstanzaindent	298
COMMAND\setstanzaindents	49, 297, 354
COMMAND\setstanzapenalties	297
COMMAND\setstanzavalues	297
COMMAND\settoggle@series	243, 413, 417
COMMAND\showlemma	123, 410, 411
COMMAND\showwordrank	30, 133
COMMAND\sidenote@margin	411
COMMAND\sidenotemargin	57, 411, 416
COMMAND\sidenoteseq	357
COMMAND\sidepstartnumtrue	18
COMMAND\skip	167
COMMAND\skipnumbering	22, 110, 121, 412, 419, 420
COMMAND\skipnumbering@reg	419
COMMAND\small	42
COMMAND\special	12
COMMAND\splitmaxdepth	162, 177
COMMAND\splitoff	176
COMMAND\splittopskip	162, 177, 179
COMMAND\stanza	21, 22, 50, 51, 301, 357, 359, 422
COMMAND\stanza@hang	300
COMMAND\stanza@line	300
COMMAND\stanzaindent	49, 298, 418
COMMAND\stanzaindent*	49
COMMAND\stanzaindentbase	297
COMMAND\stanzanumwrapper	51
COMMAND\startlock	21, 98, 121, 301
COMMAND\startstanzahook	357
COMMAND\startsub	21, 98, 119
COMMAND\stopmsd	305
COMMAND\stopmsdata	31, 303, 304
COMMAND\strip@pt	170
COMMAND\strutbox	177
COMMAND\sub@action	98, 108
COMMAND\sub@lock	96
COMMAND\sub@off	105, 260
COMMAND\sub@on	105, 260
COMMAND\subline@num	95, 96, 98
COMMAND\sublinenum@rep	410
COMMAND\sublinenumberstyle	22, 94, 410

COMMAND\sublinenumincrement	20
COMMAND\sublinenumr@p	94, 410, 412
COMMAND\sublinenumrep	94, 412
COMMAND\sublineref	52, 257, 263
COMMAND\subsectionmark	335
COMMAND\sw@inthisedtext	126
COMMAND\sw@list@inedtext	129, 136
COMMAND\symlinenum	358
COMMAND\symplinenum	357
COMMAND\sza@penalty	300
COMMAND>tag	418
COMMAND\text	352
COMMAND\text<language>	43
COMMAND\textcolor	69
COMMAND\textheight	68
COMMAND\the	410
COMMAND\thefootnoteA	32
COMMAND\thefootnoteX	414
COMMAND\thelabidx	291, 292
COMMAND\thepage	103
COMMAND\thepstart	18
COMMAND\thepstartL	413
COMMAND\thepstartR	413
COMMAND\thestanza	51
COMMAND\this@line@list@version	115
COMMAND\thisfootnote	202
COMMAND\threecol@begin@insert	178
COMMAND\threecolfootfmt	178, 424
COMMAND\threecolfootfmtX	208
COMMAND\threecolfootgroup	177
COMMAND\threecolfootgroupX	208
COMMAND\threecolfootsetup	177
COMMAND\threecolfootsetupX	207
COMMAND\threecolvfootnote	177
COMMAND\threecolvfootnote@inserted	177
COMMAND\threecolvfootnoteX	207
COMMAND\toendnotes	26, 223, 425
COMMAND\twocolfootfmt	424
COMMAND\twocolfootfmtX	205
COMMAND\twocolfootgroupX	206
COMMAND\twocolfootsetupX	205
COMMAND\twocolvfootnoteX	205
COMMAND\twolines	243, 358
COMMAND\twolines@A	243
COMMAND\twolines@B	243
COMMAND\twolines@C	243
COMMAND\twolinesbutnotmore	358
COMMAND\twolinesonlyinsamepage	358
COMMAND\txbeforeXnotes	358
COMMAND\unhbox	171

COMMAND\unpenalty	172, 173
COMMAND\unskip	173
COMMAND\unvxh	173, 359
COMMAND\unvxhX	359
COMMAND\upbracefill	328
COMMAND\usingcritext	353, 356
COMMAND\usingdtext	353, 356
COMMAND\VAfootnote	162
COMMAND\vadjust	118
COMMAND\variant	27
COMMAND\vbox	139, 141, 171, 176, 216
COMMAND\vfootnote	162, 167, 170, 177
COMMAND\vl@dbfnote	196, 411
COMMAND\vl@disnote	277
COMMAND\vl@dlsnote	277
COMMAND\vl@dosnote	277
COMMAND\vl@drsnote	277
COMMAND\vnumfootnoteX	412
COMMAND\vsizer	46, 68
COMMAND\vsplit	157
COMMAND\waklam	328
COMMAND\waklamec	328
COMMAND\wapunktel	328
COMMAND\wastricht	328
COMMAND\widthX	47, 360, 424
COMMAND\wrap@edcrossref	263, 417
COMMAND\wrapcontentX	43, 425
COMMAND\wrapped@bodyfootmarkX	213
COMMAND\wrapped@footfootmarkX	213
COMMAND\x...	53
COMMAND\xabslineref	264
COMMAND\xdef	88, 301
COMMAND\xflagref	53, 264, 360, 423
COMMAND\xleft@appenditem	88, 123
COMMAND\xlineref	53, 360, 423
COMMAND\xpageref	53
COMMAND\xpstartref	53, 415
COMMAND\xr	57
COMMAND\xright@appenditem	88
COMMAND\xsublineref	53
COMMAND\xxref	53, 265, 271, 415, 418, 425
COMMAND\zz@@@	410
ENVIRONMENTastanza	423
ENVIRONMENTedarrayc	333
ENVIRONMENTedarrayl	333
ENVIRONMENTedarrayr	333
ENVIRONMENTedtabularc	333
ENVIRONMENTedtabularl	333
ENVIRONMENTedtabularr	333
ENVIRONMENTledgroup	74, 283, 360, 423

ENVIRONMENTledgroupsize	284
PACKAGE(r)(e)ledmac	33
PACKAGEEledmac	11, 71, 98, 289, 355, 356, 418, 419
PACKAGEEledpar	419
PACKAGEEtoolbox	74
PACKAGEParallel	362
PACKAGEREledmac	359, 360
PACKAGEamsgen	311
PACKAGEamsmath	311, 312
PACKAGEbabel	43, 69, 322, 423
PACKAGEbiblatex	67
PACKAGEbidi	43, 44, 75, 422
PACKAGEccaption	82
PACKAGEcolor	69
PACKAGEedmac	1, 6, 10, 12, 13, 71, 187, 194, 257, 297, 352, 362, 410
PACKAGEedstanza	1, 13, 295
PACKAGEeledmac	1, 10, 14–16, 59, 194, 285, 289, 315, 337, 350, 353, 355–357, 414, 416, 418
PACKAGEeledpar	82, 162, 335, 362, 412, 416–418
PACKAGEetex	422
PACKAGEetoolbox	87, 88, 131, 230, 243, 251, 276, 336, 347
PACKAGEfancyhdr	255, 426
PACKAGEfloatrow	67, 360
PACKAGEfootmisc	33, 69, 75, 194, 362, 426
PACKAGEgeometry	15
PACKAGEglossaries	60, 61, 293, 423
PACKAGEhandout	417
PACKAGEhyperlink	237
PACKAGEhyperref	53, 125, 213, 258, 291, 339, 349, 415–417, 424, 427
PACKAGEifluatex	74
PACKAGEifxetex	74
PACKAGEimakeidx	58, 59, 67, 75, 285, 289, 356, 414–416, 418
PACKAGEindextols	293
PACKAGEindextool	356
PACKAGEindextools	58, 59, 67, 75, 83, 285, 289, 293, 356, 418, 423, 426
PACKAGEinputenc	133
PACKAGEledarab	70
PACKAGEledmac	1, 10, 13, 14, 69, 87, 289, 352, 353, 356, 359
PACKAGEledpar	69
PACKAGEMemoir	74, 289, 356, 362, 417
PACKAGEMorewrites	67
PACKAGEMusixtex	416
PACKAGEperpage	423
PACKAGEpolyglossia	41, 69, 161, 183, 423
PACKAGERagged2e	44, 74
PACKAGEREledmac	1, 2, 10–12, 14–17, 19, 20, 23, 25, 27–29, 31, 33, 36, 37, 40, 43, 44, 46, 48, 50, 51, 53–55, 57, 58, 60, 61, 65–69, 71, 72, 89, 91, 97, 98, 102, 105, 115, 124, 127, 128, 131, 155, 163, 167, 171, 183, 194, 220, 231, 235, 243, 251, 263, 264, 266, 271, 272, 289, 308, 315, 335, 336, 348, 349, 356, 357, 359–361, 421, 424, 426
PACKAGEREledpar	1, 4, 6, 8, 15, 18, 47, 53, 55, 56, 66, 67, 69, 72, 82, 89, 100, 105, 126, 128, 164, 167, 214, 215, 230, 236, 250, 252, 253, 285, 295, 422, 423, 425, 426

PACKAGESuffix	74
PACKAGETabmac	1, 13, 362
PACKAGEuninormalize	28
PACKAGExargs	27, 74
PACKAGExkeyval	72, 250
PACKAGEXr	5, 57, 272, 424
PACKAGEXref	271
PACKAGEXstring	74, 291

A

\absline@num	1
Abu Kamil Shuja' b. Aslam	13
\actionlines@list	1
\actions@list	1
\add@inserts	1
\add@inserts@next	1
\add@msd@	1
\add@msdata	1
\add@msdata@firstlineofpage	1
\add@penalties	1
\add@Xgroupbyline	1
\addtol@denbody	1
Adelard II	13
\advancelabel@refs	1
\advanceline	1, 22
\Aendnote	24
\affixline@num	1
\affixpstart@num	1
\affixside@note	1
\Afootnote	24
\afternoteX	44
\afterruleX	46
\ampersand	1, 51
\append@notesX	1
\append@Xnotes	1
\applabel	1, 55
\appref	1, 55
\apprefwithpage	1, 55
\arrangementX	1, 35
\arrangementX@normal	1
\arrangementX@threecol	1
\arrangementX@twocol	1
\at@every@pend	1
\AtEveryPend	1, 18
\AtEveryPstart	1, 18
\AtEveryStanza	1
\AtEveryStopStanza	1
\autopar	1, 17

B

<code>\ballast</code>	68
<code>\ballast@count</code>	<u>1</u>
Beeton, Barbara Ann Neuhaus Friend	18
<code>\beforeedchapter</code>	<u>1</u>
<code>\beforeinsertingX</code>	43
<code>\beforenotesX</code>	46
<code>\beginnumbering</code>	<u>1</u> , 16
<code>\Bendnote</code>	24
<code>\Bfootnote</code>	24
<code>\bhookgroupX</code>	45
<code>\bhooknoteX</code>	43
<code>\bodyfootmarkA</code>	32
<code>\boxfootnotenumbers</code>	<u>1</u>
Bredon, Simon	13
Breger, Herbert	13, 315
Brey, Gerhard	13
Busard, Hubert L. L.	13
<code>\bypage@false</code>	<u>1</u>
<code>\bypage@true</code>	<u>1</u>
<code>\bypstart@false</code>	<u>1</u>
<code>\bypstart@true</code>	<u>1</u>

C

<code>\c@addcolcount</code>	<u>1</u>
<code>\c@ballast</code>	<u>1</u>
<code>\c@firstlinenum</code>	<u>1</u>
<code>\c@firstsublinenum</code>	<u>1</u>
<code>\c@labidx</code>	<u>1</u>
<code>\c@linenumincrement</code>	<u>1</u>
<code>\c@sublinenumincrement</code>	<u>1</u>
<code>\Cendnote</code>	24
<code>\Cfootnote</code>	24
<code>\ch@ck@l@ck</code>	<u>1</u>
<code>\ch@cksub@l@ck</code>	<u>1</u>
<code>\chapter</code>	<u>1</u>
<code>\check@pb@in@verse</code>	<u>1</u>
Chester, Robert of	13
Claassens, Geert H. M.	13
<code>\colalignX</code>	44
Copernicus, Nicolaus	13
<code>\critext</code>	352
<code>\ctab</code>	<u>1</u>
<code>\ctabtext</code>	<u>1</u>

D

Dekker, Dirk-Jan	69
<code>\Dendnote</code>	24
<code>\Dfootnote</code>	24
<code>\disable@familiarnotes</code>	<u>1</u>

<code>\disable@notes</code>	1
<code>\disable@sidenotes</code>	1
<code>\disablel@dtabfeet</code>	1
<code>\do@actions</code>	1
<code>\do@actions@fixedcode</code>	1
<code>\do@actions@next</code>	1
<code>\do@ballast</code>	1
<code>\do@feet@custom@order</code>	1
<code>\do@feetX</code>	1
<code>\do@insidelinehook</code>	1
<code>\do@line</code>	1
<code>\do@linehook</code>	1
<code>\do@lockoff</code>	1
<code>\do@lockoffL</code>	1
<code>\do@lockon</code>	1
<code>\do@lockonL</code>	1
<code>\do@Xfeet</code>	1
<code>\doedindexlabel</code>	1
<code>\doendnotes</code>	1, 25
<code>\doendnotesbysection</code>	1, 25
<code>\doinsidelinehook</code>	1, 23
<code>\dolinehook</code>	1, 23
<code>\dosplits</code>	1
Downes, Michael	68, 171, 173
<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, 328
<code>edarrayc</code> (environment)	61
<code>edarrayl</code> (environment)	61
<code>edarrayr</code> (environment)	61
<code>\edatleft</code>	1, 63
<code>\edatright</code>	1, 63
<code>\edbeforetab</code>	1, 63, 328
<code>\edfilldimen</code>	1
<code>\edfont@info</code>	1
<code>\edindex</code>	1, 58
<code>\edindexlab</code>	1, 60
<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, 54
<code>\edpageref</code>	1, 52
<code>\edrowfill</code>	1, 62

<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 62
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc</code> (environment)	61
<code>edtabularl</code> (environment)	61
<code>edtabularr</code> (environment)	61
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 23
<code>\edvertdots</code>	1, 64
<code>\edvertline</code>	1, 64
<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>	61
<code>edarrayl</code>	61
<code>edarrayr</code>	61
<code>edtabularc</code>	61
<code>edtabularl</code>	61
<code>edtabularr</code>	61
<code>ledgroup</code>	51
<code>ledgroupsize</code>	52
<code>minipage</code>	51

Euclid	13
\extensionchars	<u>1</u> , 66

F

\f@x@l@cks	<u>1</u>
Fairbairns, Robin	33
\first@linenum@out@false	<u>1</u>
\first@linenum@out@true	<u>1</u>
\firstlinenum	<u>1</u> , 20
\firstseriesX@	<u>1</u>
\firstsublinenum	<u>1</u> , 20
\firstXseries@	<u>1</u>
\fix@page	<u>1</u>
\flag@end	<u>1</u>
\flag@end@later	<u>1</u>
\flag@end@RTL	<u>1</u>
\flag@start	<u>1</u>
\flag@start@later	<u>1</u>
\flag@start@RTL	<u>1</u>
\flagstanza	<u>1</u> , 51
\flush@notes	<u>1</u>
\fnpos	<u>1</u> , 33
Folkerts, Menso	13
\footfootmarkA	32
\footfudgefiddle	<u>1</u> , 68
\footnote	<u>1</u>
\footnoteA	32
\footnoteB	32
\footnoteC	32
\footnoteD	32
\footnoteE	32
\footnotelang@lua	<u>1</u>
\footnotelang@poly	<u>1</u>
\footnoteoptions@	<u>1</u>
\footplitskips	<u>1</u>
\fullstop	<u>1</u>

G

Gädeke, Nora	13
\get@edindex@hyperref	<u>1</u>
\get@edindex@ledinnote@command	<u>1</u>
\get@fnmark	<u>1</u>
\get@fnmarkX	<u>1</u>
\get@index@command	<u>1</u>
\get@linelistfile	<u>1</u>
\get@sw@txt	<u>1</u>
\get@thisfootnote	<u>1</u>
\get@thisfootnoteX	<u>1</u>
\getline@num	<u>1</u>
\gl@p	<u>1</u>

H

\h@num	1
\hangindentX	43
\hidenumbering	1, 22
\hidenumberingonleftpage	1, 23
\hidenumberingonrightpage	1
\Hilfsbox	1
\hilfsbox	1
\hilfscount	1
\HILFSskip	1
\Hilfsskip	1
\hilfsskip	1
\hsizethreecolX	44
\hsizetwocolX	44
\Hy@raisedlink@left	1
\hyperlinkformat	1
\hyperlinkformatR	1
\hyperlinkR	1

I

\if@addsw	1
\if@edtext@secondarg@	1
\if@eled@sectioning	1
\if@firstlineofpage	1
\if@led@nofoot	1
\if@ledgroup	1
\if@lemmacommand@	1
\if@msdata@insertedfrompreviouspage	1
\if@noeled@sec	1
\if@noneed@Footnote	1
\if@RTL	1
\ifautopar@pause	1
\ifbypage@	1
\ifbypage@R	1
\ifbypstart@	1
\ifbypstart@R	1
\ifeledmaccompat@	1
\iffirst@linenum@out@	1
\ifindtl@innote	1
\ifindtl@notenumber	1
\ifinserthangingsymbol	1
\ifinstanza	1
\ifl@d@dash	1
\ifl@d@elin	1
\ifl@d@esl	1
\ifl@d@pnum	1
\ifl@d@ssub	1
\ifl@d@Xmorethantwolines	1
\ifl@d@Xtwolines	1
\ifl@dend@X	1

<code>\ifl@dhidenumber</code>	1
<code>\ifl@dmemoir</code>	1
<code>\ifl@dpaging</code>	1
<code>\ifl@dpairing</code>	1
<code>\ifl@dprintingcolumns</code>	1
<code>\ifl@dprintingpages</code>	1
<code>\ifl@dskipnumber</code>	1
<code>\ifl@dskipversenumber</code>	1
<code>\ifl@dstartendok</code>	1
<code>\ifl@footmisc</code>	1
<code>\ifl@imakeidx</code>	1
<code>\ifl@indextools</code>	1
<code>\ifledfinal</code>	1, 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 158

L

\l@advance@parledgroup@beforenormalnotes 1
 \l@d@add 1
 \l@d@nums 1
 \l@d@section 1
 \l@d@set 1
 \l@d@Xend 1
 \l@dampcount 1
 \l@dbfnote 1
 \l@dcheckcols 1
 \l@dcheckstartend 1
 \l@dchset@num 1
 \l@dcolcount 1
 \l@dcollect@body 1
 \l@dcollect@body 1
 \l@dcolwidth 1
 \l@dcsnote 1
 \l@dcsnotetext 1
 \l@dcsnotetext@l 1
 \l@dcsnotetext@r 1
 \l@ddodoreinextrafeet 1
 \l@dedbeginmini 1
 \l@dedendmini 1
 \l@demptyd@ta 1
 \l@dend@close 1
 \l@dend@open 1
 \l@dend@stuff 1
 \l@dend@Xfalse 1
 \l@dend@Xtrue 1
 \l@denvbody 1
 \l@desnote 1
 \l@dfambeginmini 1
 \l@dfamendmini 1
 \l@dfeetbeginmini 1
 \l@dfeetendmini 1
 \l@dgetline@margin 1
 \l@dgetlock@disp 1
 \l@dgetref@num 1
 \l@dgetsidenote@margin 1
 \l@dobblearg 1
 \l@disnote 1
 \l@dlabel@parse 1
 \l@dld@ta 1
 \l@dlp@rbox 1

<code>\ldlsn@te</code>	1
<code>\ldlsnote</code>	1
<code>\ldmake@labels</code>	1
<code>\ldmodforedtext</code>	1
<code>\ldnullfills</code>	1
<code>\ldnumpstartsL</code>	1
<code>\ldoldold@footnotetext</code>	1
<code>\ldp@rsefootspec</code>	1
<code>\ldparsedendline</code>	1
<code>\ldparsedendpage</code>	1
<code>\ldparsedendsub</code>	1
<code>\ldparsedstartline</code>	1
<code>\ldparsedstartpage</code>	1
<code>\ldparsedstartsub</code>	1
<code>\ldparsefootspec</code>	1
<code>\ldpush@begins</code>	1
<code>\ldrd@ta</code>	1
<code>\ldref@undefined</code>	1
<code>\ldrestorefills</code>	1
<code>\ldrestoreforedtext</code>	1
<code>\ldrp@rbox</code>	1
<code>\ldrsn@te</code>	1
<code>\ldrsnote</code>	1
<code>\ldsetmaxcolwidth</code>	1
<code>\ldskipnumberfalse</code>	1
<code>\ldskipnumbertrue</code>	1
<code>\ldtabaddcols</code>	1
<code>\ldtabnoexpands</code>	1
<code>\ldunboxmpfoot</code>	1
<code>\ldunhbox@line</code>	1
<code>\ldzeropenalties</code>	1
<code>\label</code>	54
<code>\labelstartfalse</code>	1
<code>\labelstarttrue</code>	1, 19
<code>\labelref@list</code>	1
<code>\labelrefsparseline</code>	1
<code>\labelrefsparsesubline</code>	1
<code>\last@page@num</code>	1
Lavagnino, John	12
<code>\led@check@nopb</code>	1
<code>\led@check@pb</code>	1
<code>\led@err@AutoparNotNumbered</code>	1
<code>\led@err@BadAction</code>	1
<code>\led@err@edtextoutsidepstart</code>	1
<code>\led@err@EdtextWithoutFootnote</code>	1
<code>\led@err@FootnoteNotInSecondArgEdtext</code>	1
<code>\led@err@HighEndColumn</code>	1
<code>\led@err@LineationInNumbered</code>	1
<code>\led@err@LowStartColumn</code>	1
<code>\led@err@ManyLeftnotes</code>	1

\led@err@ManyRightnotes	1
\led@err@ManySidenotes	1
\led@err@NumberingNotStarted	1
\led@err@NumberingShouldHaveStarted	1
\led@err@NumberingStarted	1
\led@err@NumberingWithoutPstart	1
\led@err@PendNoPstart	1
\led@err@PendNotNumbered	1
\led@err@PstartInPstart	1
\led@err@PstartNotNumbered	1
\led@err@ReverseColumns	1
\led@err@toendnotes@outsidenumbering	1
\led@err@TooManyColumns	1
\led@err@UnequalColumns	1
\led@error@fail@patch@@doclearpage	1
\led@error@fail@patch@@iiiminipage	1
\led@error@fail@patch@@makecol	1
\led@error@fail@patch@@reinserts	1
\led@error@fail@patch@endminipage	1
\led@error@PackageAfterEledmac	1
\led@mess@NotesChanged	1
\led@mess@SectionContinued	1
\led@nopb	1
\led@nopbnum	1
\led@pb	1
\led@pb@setting	1
\led@pbnum	1
\led@reinit@index@fornote	1
\led@set@index@fornote	1
\led@toksa	1
\led@toksb	1
\led@warn@AppLabelOutSecondArgEdtext	1
\led@warn@BadAction	1
\led@warn@BadAdvancelineLine	1
\led@warn@BadAdvancelineSubline	1
\led@warn@BadLineation	1
\led@warn@BadLinenummargin	1
\led@warn@BadLockdisp	1
\led@warn@BadSetline	1
\led@warn@BadSetlinenum	1
\led@warn@BadSidenotemargin	1
\led@warn@BadSublockdisp	1
\led@warn@DuplicateLabel	1
\led@warn@edinde@outsidenumbering	1
\led@warn@LineFileObsolete	1
\led@warn@NoFile	1
\led@warn@NoIndexFile	1
\led@warn@NoMarginpars	1
\led@warn@RefUndefined	1
\led@warn@SeriesStillExist	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</code> (environment)	51
<code>ledgroupsize</code> (environment)	52
<code>\ledinnernote</code>	1, 57
<code>\ledinnote</code>	1
<code>\ledinnotehyperpage</code>	1
<code>\ledinnotemark</code>	1
<code>\ledleftnote</code>	1, 57
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 58
<code>\ledlsnotesep</code>	1, 58
<code>\ledlsnotewidth</code>	1, 57
<code>\lednopb</code>	1, 66
<code>\lednopbinversetrue</code>	66
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	1, 57
<code>\ledpb</code>	1, 66
<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 66
<code>\ledrightnote</code>	1, 57
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 58
<code>\ledrsnotesep</code>	1, 58
<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, 57
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 21
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	58
<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, 27

<code>\linenum@out</code>	1
<code>\linenumberlist</code>	1, 20
<code>\linenumberstyle</code>	1, 22
<code>\linenumincrement</code>	1, 20
<code>\linenummargin</code>	1, 20
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 21
<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	73

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>\measurembody</code>	1
<code>\measuremcell</code>	1
<code>\measuremrow</code>	1
<code>\measuresetbody</code>	1
<code>\measuresetcell</code>	1
<code>\measuresetrow</code>	1
Middleton, Thomas	13, 96
<code>minipage</code> (environment)	51
Mittelbach, Frank	12
<code>\morenoexpands</code>	1, 69
<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
<code>\msdata@c</code>	1
<code>\msdata@cR</code>	1
<code>\multfootsep</code>	1, 33
<code>\multiplefootnotemarker</code>	1

N

<code>\n@num</code>	1
<code>\n@num@stanza</code>	1
<code>\new@line</code>	1
<code>\newhookarg@specific</code>	1
<code>\newhookcommand@series</code>	1
<code>\newhookcommand@series@reload</code>	1
<code>\newhooktoggle@series</code>	1
<code>\newhooktoggle@series@reload</code>	1
<code>\newhooktoggle@specific</code>	1
<code>\newseries@</code>	1
<code>\newverse</code>	1
<code>\NEXT</code>	1
<code>\no@expands</code>	1
<code>\noeledsec</code>	65
<code>\nomk@</code>	1
<code>\normal@footnotemarkX</code>	1
<code>\normal@page@break</code>	1
<code>\normal@pars</code>	1
<code>\normalbfnoteX</code>	1
<code>\normalbodyfootmarkX</code>	1
<code>\normalfootfmt</code>	1
<code>\normalfootfmtX</code>	1
<code>\normalfootfootmarkX</code>	1
<code>\normalfootgroup</code>	1
<code>\normalfootgroupX</code>	1
<code>\normalfootnoterule</code>	1
<code>\normalfootnoteruleX</code>	1
<code>\normalfootstart</code>	1
<code>\normalfootstartX</code>	1
<code>\normalvfootnote</code>	1
<code>\normalvfootnote@inserted</code>	1
<code>\normalvfootnoteX</code>	1
<code>\notefontsizeX</code>	42
<code>\notenumfontX</code>	42

<code>\noteschanged@false</code>	1
<code>\noteschanged@true</code>	1
<code>\noteswidthliketwocolumnsX</code>	47
<code>\nulledindex</code>	1
<code>\nullsetzen</code>	1
<code>\num@lines</code>	1
<code>\numberedpar@false</code>	1
<code>\numberedpar@true</code>	1
<code>\numberingfalse</code>	1
<code>\numberingtrue</code>	1
<code>\numberlinefalse</code>	20
<code>\numberlinetrue</code>	20
<code>\numberpstartfalse</code>	1, 18
<code>\numberpstarttrue</code>	1, 18
<code>\numberstanzafalse</code>	51
<code>\numberstanzatrue</code>	51
<code>\numlabfont</code>	1, 47

O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>optionauxdir</code>	15, 425
<code>optioncontinuousnumberingwithcolumns</code>	424
<code>optioninnnote</code>	423
<code>optioninnnote</code>	423
<code>optionlinrangesep</code>	250
<code>optionnocritical</code>	423
<code>optionnoeledsec</code>	341, 425
<code>optionnoend</code>	423
<code>optionnopenalties</code>	68
<code>optionnotenumber</code>	423

P

<code>\page@action</code>	1
<code>\page@num</code>	1
<code>\pagelinesep</code>	1, 59
<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>	45
<code>\parafootstart</code>	1
<code>\parafootstartX</code>	1
<code>\paravfootnote</code>	1
<code>\parindentX</code>	43

<code>\pausenumbering</code>	1, 19
<code>\pend</code>	1, 17
Plato of Tivoli	13
<code>\postbodyfootmark</code>	1
<code>\prebodyfootmark</code>	1
<code>\prenotesX</code>	46
<code>\prepare@edindex@fornote</code>	1
<code>\prepare@prenotesX</code>	1
<code>\prepare@Xgroupbyline</code>	1
<code>\prepare@Xprenotes</code>	1
<code>\prev@ncpb</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, 17
<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>	45
<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, 99
<code>\resetprevpage@</code>	1
<code>\resetprevpage@num</code>	99
<code>\restore@familiarnotes</code>	1
<code>\restore@notes</code>	1
<code>\restore@sidenotes</code>	1
<code>\resumenumbering</code>	1, 19
<code>\rightctab</code>	1
<code>\rightlinenum</code>	1, 21
<code>\rightltab</code>	1
<code>\rightnoteupfalse</code>	58
<code>\rightrtab</code>	1
<code>\rightstartnum</code>	1
<code>\rigidbalance</code>	1
<code>\rigidbalanceX</code>	1
<code>\rtab</code>	1
<code>\rtabtext</code>	1

S

Sacrobosco	13
<code>\sameword</code>	1, 28
<code>\sameword@inedtext</code>	1
Schöpf, Rainer	12
<code>\section@num</code>	1
<code>\select@lemmafонт</code>	1
<code>\select@lemmafонт</code>	1, 47
<code>\SEref</code>	1, 54
<code>\SErefonlypage</code>	54
<code>\SErefwithpage</code>	1, 54
<code>\series</code>	1
<code>\seriesatbegin</code>	1, 33
<code>\seriesatend</code>	1, 33
<code>\set@line</code>	1
<code>\set@line@action</code>	1
<code>\set@Xtxtbeforenotes</code>	1
<code>\setapprefprefixmore</code>	55
<code>\setapprefprefixsingle</code>	55
<code>\setcommand@series</code>	1
<code>\sethangingsymbol</code>	1, 50
<code>\setistwofollowinglines</code>	1
<code>\setl@dlp@rbox</code>	1
<code>\setl@drpr@box</code>	1
<code>\setline</code>	1, 22
<code>\setlinenum</code>	1, 22

<code>\setmcellcenter</code>	1
<code>\setmcellleft</code>	1
<code>\setmcellright</code>	1
<code>\setmrowcenter</code>	1
<code>\setmrowleft</code>	1
<code>\setmrowright</code>	1
<code>\setmsdatalabel</code>	1, 32
<code>\setmsdataposition</code>	1, 32
<code>\setmsdataseries</code>	1, 32
<code>\setnoteswidthliketwocolumnsX@</code>	1
<code>\setnotesXpositionliketwocolumns@</code>	1
<code>\setprintendlines</code>	1
<code>\setprintlines</code>	1
<code>\setSErefonlypageprefixmore</code>	55
<code>\setSErefonlypageprefixsingle</code>	55
<code>\setSErefprefixmore</code>	55
<code>\setSErefprefixsingle</code>	55
<code>\setsidenotesep</code>	58
<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, 57
<code>\sidepstartnumtrue</code>	18
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 22
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 62
<code>\spreadtext</code>	1, 62
<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>\step1@dcollcount</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, 20
<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, 68, 82, 86, 171, 172, 257, 295
<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, 51
<code>\thestartpageline</code>	1
<code>\this@line@list@version</code>	1
<code>\threecol@begin@insert</code>	1
<code>\threecolfootfmt</code>	1
<code>\threecolfootfmtX</code>	1
<code>\threecolfootgroup</code>	1
<code>\threecolfootgroupX</code>	1
<code>\threecolfootsetup</code>	1
<code>\threecolfootsetupX</code>	1
<code>\threecolvfootnote</code>	1
<code>\threecolvfootnote@inserted</code>	1
<code>\threecolvfootnoteX</code>	1
<code>\toendnotes</code>	1, 26
<code>\toendnotes*</code>	1
<code>\twocolfootfmt</code>	1
<code>\twocolfootfmtX</code>	1
<code>\twocolfootgroup</code>	1
<code>\twocolfootgroupX</code>	1
<code>\twocolfootsetup</code>	1

<code>\twocolfootsetupX</code>	1
<code>\twocolvfootnote</code>	1
<code>\twocolvfootnote@inserted</code>	1
<code>\twocolvfootnoteX</code>	1

U

<code>\unvxhX</code>	1
----------------------------	---

V

Vamana	13
<code>\variab</code>	1
<code>\vbfnoteX</code>	1
<code>\vl@dbfnote</code>	1
<code>\vl@dcnote</code>	1
<code>\vl@disnote</code>	1
<code>\vl@dlsnote</code>	1
<code>\vl@dosnote</code>	1
<code>\vl@drsnote</code>	1
<code>\vnumfootnoteX</code>	1

W

Whitney, Ron	12
<code>\widthX</code>	47
<code>\wrap@edcrossref</code>	1
<code>\wrapcontentX</code>	43
<code>\wrapped@bodyfootmarkX</code>	1
<code>\wrapped@footfootmarkX</code>	1
Wujastyk, Dominik	12

X

<code>\X@atbegininsertion</code>	1
<code>\X@beforeinsertion</code>	1
<code>\X@doreinfeet</code>	1
<code>\xabslineref</code>	1
<code>\Xafterlemmaseparator</code>	41
<code>\Xafternote</code>	44
<code>\Xafternumber</code>	39
<code>\Xafterrule</code>	46
<code>\Xaftersymlinenum</code>	39
<code>\Xarrangement</code>	1, 35
<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>	41
<code>\Xbeforenotes</code>	45
<code>\Xbeforenumber</code>	37, 39
<code>\Xbeforesymlinenum</code>	39
<code>\Xbhookgroup</code>	45

\Xbhooknote	43
\Xboxlinenum	39
\Xboxlinenumalign	40
\Xboxsymlinenum	39
\Xcolalign	44
\xedindex	1
\xedlabel	1
\xedtext	1
\Xendafterrenumber	39
\Xendafterlemmaseparator	41
\Xendafternote	47
\Xendafterpagenumber	40
\Xendaftersymlinenum	39
\Xendahookinplaceofnumber	41
\Xendahooklinenum	41
\Xendbeforelemmaseparator	41
\Xendbeforenumber	39
\Xendbeforepagenumber	40
\Xendbeforesymlinenum	39
\Xendbhookinplaceofnumber	41
\Xendbhooklinenum	40
\Xendbhooknote	43
\Xendboxendlinenumalign	40
\Xendboxlinenum	40
\Xendboxlinenumalign	40
\Xendboxstartlinenumalign	40
\Xendboxsymlinenum	40
\Xendhangindent	43
\Xendinplaceofflemmaseparator	41
\Xendinplaceofnumber	39
\Xendinplaceofpagenumber	37
\Xendlemmadisablefontselection	42
\Xendlemmafont	42
\Xendlemmaseparator	41
\Xendlineprefixmore	40
\Xendlineprefixsingle	40
\Xendlinerangeseparator	37
\Xendmorethantwolines	37
\Xendnonumber	38
\Xendnotefontsize	42
\Xendnotenumfont	42
\Xendnumberonlyfirstinline	36
\Xendnumberonlyfirstintwolines	36
\Xendpagenumberonlyfirst	36
\Xendparagraph	47
\Xendsep	47
\Xendsublinesep	38
\Xendsymlinenum	36
\Xendsympagenum	37
\Xendtwolines	37

\Xendtwolinesbutnotmore	37
\Xendwrapcontent	43
\xflagref	<u>1</u>
\Xgroupbyline	45
\Xgroupbylineseparatetwolines	45
\Xhangindent	43
\Xhsizethreecol	44
\Xhsizetwocol	44
\Xinplaceoflemmaseparator	41
\Xinplaceofnumber	39
\Xinsertparafootsep	<u>1</u>
\Xledsetnormalparstuff	<u>1</u>
\xleft@appenditem	<u>1</u>
\Xlemmadisablefontselection	42
\Xlemmafont	42
\Xlemmaseparator	41
\Xlinrangeseparator	37
\xlineref	<u>1</u> , 53
\Xmaxhnotes	46
\Xmorethantwolines	37
\Xnolemmaseparator	<u>1</u> , 41
\Xnonbreakableafternumber	39
\Xnonumber	38
\Xnotefontsize	42
\Xnotenumfont	42
\Xnoteswidthliketwocolumns	47
\Xnumberonlyfirstinline	36
\Xnumberonlyfirstintwolines	36
\Xonlypstart	38
\Xpagelinesep	39
\xpageref	<u>1</u> , 53
\Xparafootsep	45
\Xparindent	43
\Xprenotes	<u>1</u> , 46
\Xprenotes@	<u>1</u>
\Xpstart	38
\Xpstarteverytime	38
\xpstartref	<u>1</u> , 53
\XR@test	<u>1</u>
\XR@test@mac	<u>1</u>
\XR@test@mac@test	<u>1</u>
\Xragged	45
\xright@appenditem	<u>1</u>
\Xrigidbalance	<u>1</u>
\Xstanza	38
\Xstanzaseparator	38
\Xstorelineinfo	<u>1</u>
\xsublineref	<u>1</u> , 53
\Xsublinesep	38
\Xsymlinenum	36

<code>\Xtoendnotes</code>	26
<code>\Xtwolines</code>	37
<code>\Xtwolinesonlyinsamepage</code>	37
<code>\Xtxtbeforenotes</code>	45
<code>\Xunvxh</code>	<u>1</u>
<code>\Xwidth</code>	47
<code>\Xwrapcontent</code>	43
<code>\Xwrapendlemma</code>	43
<code>\Xwraplemma</code>	42
<code>\xxref</code>	<u>1</u> , 53

Z

<code>\zz@@@</code>	<u>1</u>
---------------------------	----------

Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
\ifl@dmemoir: Added \ifl@dmemoir for memoir class having been used	74
\morenoexpands: Added \l@dtabnoexpands to \no@expands	124
\reledmac@error: Added \eledmac@error and replaced error messages	76
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	260
\doxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	251
\edlabel: Tweaked \edlabel to get correct page numbers	257
\l@ddodoreinxtrafeet: Renamed \dodoreinxtrafeet to \l@ddodoreinxtrafeet	253
\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	124
\zz@@@: Minor change to \zz@@@	257
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	254
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	169
\line@list@stuff: Added initial write of page number in \line@list@stuff	116
\para@footsetup: Added \footfudgefiddle to \para@footsetup	169
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	209
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	82
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	260
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	103
\ledlinenum: Added \linenumr@p and \sublinenum@rep to \leftlinenum and \rightlinenum	94
\linenumberlist: Added \linenumberlist mechanism	82
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	226
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	192
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	94
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	282
\Xarrangement@normal: Added minpage footnote setup to \footnormal	164
\edtext: Added \showlemma to \edtext	125
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	280

\mpnormalfootgroup: Added \mpnormalfootgroup	168
\mpnormalvfootnote: Added \mpnormalvfootnote	165
\showlemma: Added \showlemma	82
General: Added final/draft options	72
Added ledgroup environment	283
Added ledgroupsize environment	284
Added minipage, etc., support	1
v0.4.1.	
\do@Xfeet: Changed \do@Xfeet code for easier extensions	252
\edindex: Let eledmac take advantage of memoir's indexing	289
\print@Xnotes: Added \op@Xfeet	252
General: Added code for changing \doclearpage	255
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	195
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	196
\@xympar: Eliminated \marginpar disturbance	272
General: Added left and right side notes	273
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	150
\l@edgetsidenote@margin: Added \sidenotemargin and \sidenote@margin	273
General: Added moveable side note	273
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	284
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	105
\@nl@reg: Added \fix@page to \@nl	103
Extended \@nl to include the page number	103
\fix@page: Added \last@page@num and \fix@page	104
\get@thisfootnote: Changed \l@dbfnote and \vl@dbfnote as originals could give incorrect markers in the footnotes	196
\new@line: Extended \new@line to output page numbers	116
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	103
\@ref@reg: Added \@ref@reg	112
\affixline@num: Added skipnumering to \affixline@num	150
\do@actions@fixedcode: Added \do@actions@fixedcode	149
\do@actions@next: Added number skipping to \do@actions	147
\do@insidelinehook: Added \do@linehook for use in \do@line	145
\endnumbering: Changed \endnumbering for eledpar	86
\fix@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fix@l@cks	152
\footplitskips: Added \footplitskips for use in many footnote styles	162
\get@linelistfile: Added \get@linelistfile	102
\initnumbering@reg: Added \initnumbering@reg	84

\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	283
\l@dcsnotetext@r: Added \l@emptyd@ta	146
\l@dgetline@margin: Added \l@dgetline@margin	90
\l@dgetlock@disp: Added \l@dgetlock@disp	93
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	273
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	82
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	146
\l@dzeropenalties: Added \l@dzeropenalties	141
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum ..	94
\line@list@stuff: Deleted \page@start from \line@list@stuff	116
\list@clearing@reg: Added \list@clearing@reg	101
\n@num: Added \n@num	110
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	202
\resumenumbering: Changed \resumenumbering for eledpar	87
\setprintendlines: Added \setprintendlines for use by \printendlines ...	225
\setprintlines: Added \setprintlines for use by \printlines	189
\skipnumbering: Added \skipnumbering and supports	121
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	92
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	94
Using \sublinenumrep instead of \sublinenumr@p	94
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	203
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	75
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	82
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.	298
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses.	49
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.	257
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.	150
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	296
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code>	191
General: Line numbering can be reset at each <code>pstart</code>	89
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.	296
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	124
General: <code>\lemma</code> can contain commands.	26
Debug in lineation command	20
New generic commands to customize footnote display.	34, 242
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code>	24
Options of <code>\Xfootnotes</code>	160
Possibility to have commands in sidenotes.	57
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.	36
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code>	217
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	243
General: Add <code>\labelpstarttrue</code>	19
Add <code>\Xnumberonlyfirstintwolines</code>	36
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	38
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	241
Sectioning commands.	64

v1.2.0.	
<code>\Xprenotes</code> : Debug in familiar footnotes (bug introduced by v1.1).	217
<code>\endquote</code> : Compatibility of <code>\ledchapter</code> with the <i>memoir</i> class.	334
v1.3.0.	
<code>\endquote</code> : <i>Quotation</i> and quote environment inside numbered sections.	334
v1.4.0.	
<code>\edtext</code> : Compatibility of <code>\edtext</code> with the right-to-left direction (with Polyglossia).	125
<code>\ledsetnormalparstuffX</code> : Direction of footnotes with polyglossia.	214
<code>\newseries@</code> : Remembers the language of the lemma, in order to create a correct direction for the footnote separator.	232
<code>\rbracket</code> : Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX).	183
General: Compatibility with LuaTeX of RTL notes.	1
v1.4.1.	
<code>\affixside@note</code> : Remove spurious spaces.	279
<code>\endquote</code> : New option <i>noquotation</i> .	334
<code>\get@thisfootnote</code> : Compatibility of standard footnotes with <code>eledmac</code> when these footnotes contain any commands.	196
<code>\labelrefsparsesubline</code> : Fix bug with <code>\edlabel</code> .	259
v1.4.2.	
General: Debug with some special classes.	1
v1.4.3.	
General: Add <code>\Xnonbreakableafternumber</code> .	39
Spurious space after familiar footnotes.	1
v1.4.4.	
General: Label inside familiar footnotes.	1
v1.4.5.	
General: Bug with <code>komasscript</code> + <code>eledpar</code> + <code>chapter</code> .	1
v1.4.6.	
General: Bug with <i>memoir</i> class introduced by 1.4.5.	1
v1.4.7.	
<code>\endquote</code> : Compatibility of sectioning commands with <code>\autopar</code> .	334
v1.4.8.	
General: Corrects a bug with parallel texts introduced by 1.1.	1
v1.4.9.	
<code>\normalbfnoteX</code> : Allow to redefine <code>\thefootnoteX</code> with <code>alph</code> when some packages are loaded.	202
v1.5.0.	
<code>\do@insidelinehook</code> : Added <code>\do@insidelinehook</code> for use in <code>\do@line</code>	145
<code>\edindex</code> : Compatibility with <code>imakeidx</code> package, and possibility to use multiple index with <code>\edindex</code> .	289
General: Correct indexing when the call is made in critical notes.	285
v1.5.1.	
<code>\managestanza@modulo</code> : Correct <code>stanzaindentsrepetition</code> counter	298
<code>\normalvfootnoteX</code> : Fix bug with normal familiar footnotes when mixing RTL and LTR text.	199
v1.6.0.	
<code>\newverse</code> : Add <code>\falseverse</code> macro.	301
v1.6.1.	
<code>\AtEveryPstart</code> : Spurious space in <code>\pstart</code> .	138

\ifinserthangingsymbol: Hang verse is now not automatically flush right.	296
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill inside.	142
\pend: Spurious space in \pend.	139
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.	66
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts.	334
\get@index@command: Debug \get@index@command and compatibility with hyperref package.	288
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work.	246
\prevpage@num: Correct \parafootsep when using with ledgroup.	175
General: Compatibility with parledgroup option ofeledpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	285
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	219
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.	285
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes.	251
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup.	280
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	334
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.	42
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.	103
\AtEveryPstart: New optional argument for \pstart, to execute code before it. . .	138
\edindex: Use correctly default index when imakeidx is loaded.	289
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	334
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar. .	84
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	273
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in eledmac, including needs for eledpar case.	274
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	82
\l@dunhbox@line: \do@line is split in more little commands.	143
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	246
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	246
\pend: New optional argument for \pend, to execute code after it.	139
\stanza: &can have an optional argument: content to be printed after.	301
\Stanza can have an optional argument: content to be printed before.	301
Add \newverse macro, \falseverse deprecated.	301
General: Add \ledinnernote and \ledouternote commands.	57
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	285
New hooks: \Xafterrule and \afterruleX	46
New options for ragged-paragraph notes	45
New sectioning commands.	64
Optional arguments for \pstart and \pend.	18

v1.12.1.

\wrap@edcrossref: Fix spurious spaces.	263
--	-----

v1.12.2.

\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	142
--	-----

v1.12.3.

\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	117
\flag@start send a error message when a \edtext is done without insert (note)	117
\reledmac@error: Replaced error messages	76
General: Add macros for new messages since v0.7	75
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.	57
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	291
\@ssect: Debug \eledchapter in parallel with memoir	337
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	146
\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	86
\l@dgobblearg: \l@dgobblearg becomes \l@gobbeloptarg	317
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	317
\resumenumbering: Debug \resumenumbering	87
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	263
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	117
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	245
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	209
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	243
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	47
Added widthliketwocolumns option	72
v1.13.1.	
\thepstart: Add \l@dzeropenalties in \pstart	138
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	82
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	289
General: Debug spurious characters before endnotes.	219
Delete previous override of \l@d@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	292
Move gobbling command	82
Provide \@gobblefour	82
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref	
package.	339
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	337
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	125
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	209
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	119
\if@lemmacommand@: New boolean \iflemmacommand@.	130

General: Fix bug with footnotes layout when using some options of the geometry package (bug add by v1.13.0).	1
New commands <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> .	18
New tools to prevent ambiguous references in lemma	28
v1.15.1.	
<code>\line@list@stuff</code> : Revert modification of 1.5.2 which makes bug with numbering. Leave vertical mode to solve spurious space before minipage.	116
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.	125
Compatibility of standard footnotes with some biblatex styles.	1
New <code>\stanzaindent</code> command.	1
v1.16.1.	
<code>\xlineref</code> : <code>\lineref</code> is not defined if defined by some other package, like <code>lineno</code> . <code>Eledmac</code> provides <code>\edlineref</code> instead.	263
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph.	125
v1.18.0.	
<code>\@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools + hyperref</code> .	291
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code>	257
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes.	220
<code>\l@dnumpstartsL</code> : Add <code>\ifl@dprintingpages</code> and <code>\@dprintingcolumns</code> for <code>eledpar</code>	82
<code>\print@line</code> : Compatibility with <code>Lua\TeX</code> RTL languages.	143
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code>	185
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns (<code>eledpar</code>)	183
General: Add <code>\Xpstarteverytime</code>	38
Compatibility with <code>Lua\TeX</code> RTL languages.	1
Debug <code>\Xonlypstart</code> when using <code>\Xnumberonlyfirstinline</code> and the current line number differs from the previous.	38
v1.19.0.	
<code>\footssplitskips</code> : <code>\footssplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages.	162
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty.	265
General: <code>\Xmaxhnotes</code> and <code>\maxhnotesX</code> work now for both two-columns and three-columns setting.	1
Compatibility with <code>eledpar v1.13.0</code> .	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>	192
<code>\stanza</code> : <code>&</code> and <code>&</code> can be preceded by spaces.	301
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (fix bug added in 1.19.0).	265
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	291
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	220
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code>	288
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code>	214
<code>\n@num</code> : <code>\n@num@ref</code> deleted	110
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	110
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	245
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	245
<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>	216
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	242
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	121
Correct <code>\skipnumbering</code> for stanza.	121
Delete <code>\skipnumbering@reg</code>	121
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	241
Add <code>\applabel</code> and related	55
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	44
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code>	37
Add <code>\Xparindent</code> and <code>\hangindentX</code>	43
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>	82
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	288
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>	175
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>	125
<code>\SErefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code>	269
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	129
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	232
<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.	133
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code>	40
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code>	40
Allow use of <code>\sameword</code> with <code>inputenc</code> managing of UTF-8.	1
Compatibility between <code>nofamiliar/nocriticals</code> option and <code>minipage/ledgroup</code>	1
Error message when using <code>\beginnumbering... \endnumbering</code> without <code>\pstart</code>	1
Fix bug with <code>\sameword</code> when the lemma overlaps multiple line.	28
Fix bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code>	28
Fix bug with <code>\skipnumbering</code> called immediately after a <code>\pstart</code>	1
Fix error of <code>\iftrue</code> not closed.	1
Fix spurious space with <code>\skipnumbering</code> (bug added on v1.21.0).	1
New tools to ensure the line-list file uses the right version of commands when upgrading the <code>eledmac</code> version.	1
Optional argument of <code>\sameword</code> can be a comma-separated list of <code>\edtext</code> depth.	28
v1.23.1.	
General: Fix bug with <code>\lemma</code> command in the right side.	1
v1.23.2.	
General: Compatibility with L ^A T _E X's release 2015.	1

v1.24.0.	
General: We can reinitialize <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> providing to it an empty argument.	1
v1.24.1.	
General: <code>\lemma</code> is disabled when using ‘ <code>nocritical</code> ’ option.	1
v1.24.2.	
General: Fix incompatibility between ‘ <code>nofamiliar</code> ’ option and ‘ <code>memoir</code> ’ package.	1
v1.24.3.	
General: Restore marginal numbers and notes with sectioning command (bug introduced in v1.21.0)	1
v1.24.4.	
General: Fix spurious space with <code>\edindex</code> when using <code>xindy+hyperref</code> option.	1
v1.24.5.	
General: Fix bug of indent, when a added in 1.1.0, when a <code>\beginnumbering</code> immediately follow a sectioning command.	1
v2.0.0.	
<code>\@iiiminipage</code> : Patch <code>\@iiiminipage</code> instead of redefining it.	282
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	272
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	282
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	333
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some conde of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	283
<code>\newseries@</code> : One endnotes file by series.	238
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined	254
<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>	254
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	195
Suppress indexing command specific to <code>memoir</code>	289
v2.0.1.	
General: Fix bug in <code>eledmac-compat</code> option	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code>	1
v2.1.0.	
General: Fix bug with <code>\advanceline</code> at the beginning of a <code>\pstart</code>	1
Fix bug with <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting with <code>scrbook</code>	1
Fix bug with <code>\eledchapter</code> in parallel typesetting with <code>scrbook</code>	1
Fix bug with <code>\setline</code> at the beginning of a <code>\pstart</code>	1
Fix spacing bug with <code>\Xhooknote</code> and <code>\hooknoteX</code> when using them to insert text and not to execute code.	1

New tools to number stanzas	1
v2.1.1.	
General: Fix bug with <code>\ledpbsetting{before}</code>	1
v2.1.2.	
General: Fix bug with lineation by <code>pstart</code> and <code>tabular</code> environments (added in 2.1.0).	1
v2.1.3.	
<code>\ledsetnormalparstuffX</code> : Replaced <code>\noindent</code> with <code>\parindent</code> set to 0pt.	214
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.	160
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>	37

Option linerangesep for critical footnotes and endnotes.	37
v2.4.1.	
General: Fix bug with \appref and \apprefwithpage (introduced in v2.4.0).	1
Fix bug with tabular environments when using babel or polyglossia languages that override \LaTeX \roman command, like Greek language.	1
Fix bug with tabular environments when using babel or polyglossia languages that override \LaTeX \roman command, like Greek.	1
v2.5.0.	
\SErefwithpage: Debug \setapprefprefixsingle	268
\edlabel: Fix bug when calling \edlabel in a footnotes of the rightside	257
\l@d@section: \endnotes take six arguments.	220
\printlines: \printlines takes an eighth argument: the line flag	191
\xlineref: \xlineref does not include anymore the side flag. Use \xflagref to get it. Not that \edlineref still contains the flag.	263
General: \apprefwithpage and \appref print double quotation mark when the label was not defined.	1
\apprefwithpage and \appref work with right side crossref.	1
\apprefwithpage works also when noend option is enabled.	1
\appref and \apprefwithpage can take linerangesep optional argument.	1
\edlabel works now in \Xfootnote.	1
\lemma can be used even when the nocritical is enabled.	1
Compatibility with new hook and tools of reledpar 2.6.0.	1
Fix spurious vertical space in astanza environment (reledpar)	1
Log now states ‘There were undefined references’ when using wrong references in \edlineref or edpageref.	1
New hooks to customize page and line number appearance in endnotes.	1
New hooks: \Xhookgroup and \bhookgroupX.	1
New tools to easily make cross-reference to a passage defined by a start and an end line	54
v2.6.0.	
General: Adds compatibility with innnote and notenumber options of indextools package.	1
Fix bug with footnote counter in ledgroup (added in v2.5.0).	1
Fix bug, introduced in v2.5.0, with footnote numbering in parallel typesetting when using perpage package.	1
v2.7.0.	
\@k: \rigidbalance is split in \Xrigidbalance and \rigidbalanceX.	176
\l@d@section: \endnotes take seven arguments.	220
General: Add dash as default page range separator for \SEonlypage	1
Debug \SErefonlypage when referring to only one page.	1
Delete parenthesis after \SErefonlypage.	1
Fix (again) bugs with footnote numbering in parallel typesetting while using ledgroup environments (bug added in v2.5.0).	1
Fix bug with \SErefwithpage.	1
Fix bugs in compatibility with innote and notenumber options of indextools package, when indexing outside of a ledgroup.	1
New commands to make glossaries connected to page and linenumber with the glossaries package	1
New hooks: \Xhsize and \hsizeX	47
New hooks: \Xlemmafont and \Xendlemmafont	42

New setting commands: <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code>	1
Warning for duplicate and undefined labels are parsable by latexmk	1
Warning for duplicate labels does not send any more a false line and page number	1
When using <code>hyperref</code> package, add link in familiar footnotes between the footnote marks in the text and the footnote marks in the footnote	1
When using <code>hyperref</code> package, add links for <code>\SEref</code> and <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>	47
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.	220
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>	184

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> .	43
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	149
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>	254
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1
Add <code>\Xpagelinesep</code> hook.	39
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.	254
New options for <code>\fnpos</code> and <code>\mpfnpos</code> to set a customized order for familiar and critical footnotes.	1
When <code>\edindex</code> is called outside of a <code>\beginnumbering... \endnumbering</code> structure, it is automatically switched to <code>\index</code> , with a warning message.	1
When indexing texts in familiar footnotes with <code>\edtext</code> , refer to the line number where the footnote is called.	1
When indexing texts in sidenotes with <code>\edtext</code> , refer to the line number where the sidenote is called.	1
v2.16.1.	
General: Fix bug with redefinition of the style of the footnote number (bug added in v2.12.0)	1
v2.16.2.	
General: Error message if <code>footmisc</code> is loaded after <code>reledmac</code>	1
Fix bug introduced by v2.16.1 when using non-expandable control sequence, like <code>\normalfont</code> , in the footnote number style.	1
v2.16.3.	
General: Fix bug with <code>\Seref</code> (bug added in v2.7.0).	1
v2.16.4.	
General: Fix bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.15.3).	1
v2.16.5.	
General: Fix potential spurious spaces in endnotes.	1

v2.16.6.	
General: Fix bug with the line number style in <code>\doennotes</code> when referring to right side	
line in parallel typesetting.	1
Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code>	1
v2.16.7.	
General: Fix bug with <code>\msdata</code> when using multiple	
<code>\beginnumbering... \endnumbering</code>	1
Fix bug with <code>\numberpstarttrue</code> when using multiple	
<code>\beginnumbering... \endnumbering</code>	1
v2.16.8.	
General: Fix bug with <code>\edindex</code> in footnotes in parallel typesetting.	1
v2.17.0.	
General: Add <code>\edglsadd</code> command.	1
Add <code>\setmsdataposition</code> setting.	1
v2.17.1.	
General: Fix spurious space in paragraphed footnotes when using <code>Lua\TeX</code> without using	
Right-To-Left text.	1
v2.17.2.	
General: Change log message when numbered files still don't exist, in order to improve	
compatibility with <code>latexmk</code>	1
v2.17.3.	
General: Fix bug with <code>\doendnotesbysection</code> and <code>\doendnotes</code>	1
v2.17.4.	
General: Fix bug with <code>\setSErefonlypageprefixsingle</code> and	
<code>\setSErefonlypageprefixmore</code>	1
v2.17.5.	
General: Fix bug with <code>\pstartref</code> when referring to the left side in parallel typesetting.	1
v2.18.0.	
General: Fix bug when using a <code>\edtext</code> in two lines or more in right-to-left typesetting	
with <code>X\TeX</code>	1
Fix bug when using both <code>\Xnumberonlyfirstintwolines</code> or	
<code>\Xnumberonlyfirstinline</code> and <code>\Xparafootsep</code> and <code>\Xsymlinenum</code>	1
v2.18.1.	
General: Fix bug when using <code>\msdata</code> with <code>Lua\TeX</code> or with <code>hyperref</code> package.	1