

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 v2.23.0, last revised 2017/07/31.

[†]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>	18
5.2.3 Content before specific <code>\pstart</code> and after specific <code>\pend</code>	18
5.2.4 Content before every <code>\pstart</code> and after every <code>\pend</code>	19
5.2.5 Numbering paragraphs (<code>\pstart</code>)	19
5.2.6 Languages written in Right to Left	20
5.2.7 Memory limits	20
5.3 Lineation commands	20
5.3.1 Disabling lineation	20
5.3.2 Setting lineation start and step	21
5.3.3 Setting lineation reset	21
5.3.4 Setting line number margin	21
5.3.5 Other settings	22
5.4 Changing the line numbers	22
5.4.1 Sublineation	22
5.4.2 Locking lineation	22
5.4.3 Setting and changing line number	22
5.4.4 Line number style	23
5.4.5 Skipping and hiding number	23
5.5 Executing code at each line	24
6 Apparatus commands	24
6.1 Terminology	24
6.2 Critical notes	24
6.2.1 The lemma	24
6.2.2 Footnotes	25
6.2.3 Endnotes	25
6.2.4 Paragraph in critical apparatus	27
6.2.5 Change lemma and line number	27
6.2.6 Changing the names of commands for critical apparatus	28
6.3 Disambiguation of identical words in the apparatus	28
6.3.1 Basic use	28

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

7.6.3 For all notes	45
7.7 Wrapping notes	45
7.7.1 Wrapping lemmas	45
7.7.2 Wrapping contents	45
7.8 Indent of notes content	45
7.9 Arbitrary code at the beginning of notes	46
7.10 Arbitrary code before inserting note	46
7.11 Options for footnotes in columns	46
7.11.1 Alignment	46
7.11.2 Size of the columns	47
7.12 Options for paragraphed footnotes and notes grouped by line	47
7.12.1 Mark separation of notes	47
7.12.2 Ragged text	47
7.13 Options for block of notes	47
7.13.1 Grouping notes by line	47
7.13.2 Text before notes	48
7.13.3 Code before notes	48
7.13.4 Spacing	48
7.13.5 Rule	48
7.13.6 Maximum height	49
7.13.7 Width	49
7.14 Footnotes and the reledpar columns	49
7.15 Endnotes in one paragraph	50
8 Fonts	50
9 Verse	50
9.1 Basic	50
9.2 Define stanza indents	51
9.3 Repeating stanza indents	51
9.4 Manual stanza indent	52
9.5 Stanza breaking	52
9.6 Hanging symbol	52
9.7 Long verse and page break	53
9.8 Content before/after verses	53
9.9 Numbering stanza	54
9.10 Various tools	54
9.11 Notes on empty lines	54
10 Grouping	55
11 Cross referencing	55
11.1 Basic use	55
11.2 Cross-referencing to a critical note	56
11.3 Cross-referencing which return a number in any case	56
11.3.1 Cross-referencing in order to define line number of a critical note	57

11.4 Not automatic cross-referencing	57
11.5 Normal \LaTeX cross-referencing	57
11.6 References to start and end lines	57
11.6.1 Reference to main text lines	57
11.6.2 References to lines that are commented on in the apparatus	58
11.6.3 Settings	58
11.7 Compatibility with <code>xr</code> package	60
12 Side notes	60
12.1 Basics	60
12.2 Setting	61
12.2.1 Width	61
12.2.2 Vertical position	61
12.2.3 Distance to the main text	61
12.2.4 Font	61
12.2.5 Separator between notes	61
13 Indexing	61
13.1 Basics	61
13.2 Referring to critical notes	62
13.3 Separator between page and line numbers	62
13.4 Using <code>xindy</code>	63
13.5 Advanced setting	63
14 Glossary	63
14.1 Preamble setting	64
14.2 Commands	64
15 Tabular material	64
16 Sectioning commands	67
16.1 Sectioning commands without line numbers or critical notes	67
16.2 Sectioning commands with line numbering and critical notes	68
16.3 Optimization	68
17 Quotation environments	69
18 Page breaks	69
18.1 Control page breaking	69
18.2 Prevent page break in a long verses	69
19 Miscellaneous	70
19.1 Known and suspected limitations	70
19.1.1 Non-standard geometry	70
19.1.2 <code>floatrow</code> package compatibility	70
19.1.3 ‘No room for a new’	70
19.1.4 Marginal notes	71

19.1.5 Paragraph shape	71
19.1.6 Paragraphed footnotes	71
19.1.7 Use with other packages	72
19.2 Parallel typesetting	73
I Implementation overview	74
II Preliminaries	74
II.1 Links with original edmac	74
II.2 Package declaration	74
II.3 Package options	75
II.4 Loading packages	77
II.5 Compatibility with LuaTeX	77
II.6 Boolean flags	77
II.7 Messages	79
II.8 Gobbling	85
II.9 Miscellaneous commands	85
II.10 Prepare reledpar	86
II.11 Booleans provided by other optional packages which are required in any case	86
III Sectioning commands	86
IV List macros	91
V Line counting	92
V.1 Choosing the system of lineation	92
V.2 Line number margin	94
V.3 Line number initialization and increment	95
V.4 Line number locking	96
V.5 Line number style	97
V.6 Line number printing	97
V.7 Line number counters and lists	98
V.8 Line number locking counter	100
V.9 Line number associated to lemma	100
V.10 Reading the line-list file	103
V.11 Commands within the line-list file	105
V.12 Writing to the line-list file	118
VI Marking text for notes	125
VI.1 \edtext itself	126
VI.2 Substitute lemma	133
VI.3 Substitute line numbers	133
VI.4 Lemma disambiguation	134

VII Paragraph decomposition and reassembly	141
VII.1 Boxes, counters, \pstart and \pend	141
VII.2 Processing one line	147
VII.2.1 General process	147
VII.2.2 Process for “normal” line	148
VII.2.3 Process for line containing \eledsection command	150
VII.2.4 Hooks	150
VII.2.5 Sidenotes and marginal line number initialization	151
VIII Line and page number computation	151
IX Line number printing	155
X Pstart number printing in side	159
XI Restoring footnotes and penalties	160
XI.1 Add insertions to the vertical list	160
XI.2 Penalties	162
XI.3 Printing leftover notes	163
XI.4 Text before notes	164
XII Critical footnotes	165
XII.1 Fonts	166
XII.2 Individual note options	166
XII.3 Notes language	167
XII.4 General survey of the way we manage notes	168
XII.5 General setup	168
XII.6 Footnotes arrangement	169
XII.6.1 User level macro	169
XII.6.2 Normal footnote	169
XII.6.3 Paragraphed footnotes	175
XII.6.4 Columnar footnotes	182
XII.7 Critical notes presentation	189
XII.7.1 Font tools	189
XII.7.2 Pstart number in footnote	189
XII.7.3 Lemma printing	190
XII.7.4 Line number printing	191
XII.7.5 Footnote grouped by line	199
XIII Familiar footnotes	201
XIII.1 Adjacent footnotes	201
XIII.2 Regular footnotes for numbered texts	202
XIII.3 Footnote formats	204
XIII.4 Footnote arrangement	205
XIII.4.1 User level macro	205
XIII.4.2 Normal footnotes	205
XIII.4.3 Two columns footnotes	211

XIII.4.4 Three columns footnotes	213
XIII.4.5 Paragraphed footnotes	215
XIII.5 Wrapping footnote marks in hyperlink	219
XIV Code common to both critical and familiar footnote in normal arrangement	220
XV Footnotes' width for two columns	221
XVI Footnotes' order	222
XVII Footnotes' rule	222
XVIII Specific skip for first series of footnotes	223
XVIII.0.1 Overview	223
XVIII.0.2 User level command	224
XVIII.0.3 Internal commands	224
XIX Endnotes	225
XIX.1 Internal commands	225
XIX.2 User level commands	229
XIX.2.1 Inserting contents to endnotes	229
XIX.2.2 Printing endnotes	230
XX Generate series of notes	236
XX.1 Test if series is still existing	237
XX.2 Init specific to <code>reledpar</code>	237
XX.3 For critical footnotes	237
XX.3.1 Options	237
XX.3.2 Create inserts, needed to add notes in foot	239
XX.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc.	239
XX.3.4 Set standard display	242
XX.4 For familiar footnotes	242
XX.4.1 Options	242
XX.4.2 Create tools for familiar footnotes (<code>\footnotex</code>)	243
XX.5 The endnotes	245
XX.5.1 The auxiliary file	245
XX.5.2 The main macro	245
XX.5.3 Tools	246
XX.5.4 Internal commands	246
XX.5.5 The options	247
XX.6 Init standards series (A,B,C,D,E)	248

XXI Setting series display	248
XXI.1 Change series order	248
XXI.2 Test series order	249
XXI.2.1 Get the first series	249
XXI.3 Series setting	249
XXI.3.1 General way of working	249
XXI.3.2 Tools to set options	250
XXI.3.3 Tools to generate options commands	251
XXI.3.4 Options for critical notes	253
XXI.3.5 Options for familiar notes	254
XXI.3.6 Options for endnotes	255
XXI.4 Hooks for a particular footnote	257
XXI.5 Alias	257
XXII Output routine	258
XXII.1 Extra footnotes output	258
XXII.2 Patching standard output's commands	261
XXIII Cross referencing	264
XXIII.1 Compatibility with xref	278
XXIV Side notes	279
XXV Minipages and such	287
XXVI Indexing	292
XXVI.1 Looking on package order	292
XXVI.2 Auxiliary macros for \edindex	293
XXVI.3 Code specific to \edindexin critical footnotes	294
XXVI.4 Analysis of command in indexed text	295
XXVI.5 Code for the formatted index	295
XXVI.6 Main code	296
XXVI.7 Hyperlink	298
XXVI.8 'innote' and 'notenumber' option of indextols package	300
XXVII Glossaries	301
XXVIII Verse	303
XXVIII.1 Hanging symbol management	303
XXVIII.2 Using & character	304
XXVIII.3 Code category setting	304
XXVIII.4 Stanza count and indent	304
XXVIII.5 Numbering stanza	306
XXVIII.6 Stanza number in note	307
XXVIII.7 Main work	307
XXVIII.8 Restore catcode and penalties	310

XXIX Apparatus of Manuscripts	311
XXIX.1 User level macro	311
XXIX.2 Setting macro	312
XXIX.3 Counters and lists	313
XXIX.4 Auxiliary file macros	313
XXIX.5 Action macro	314
XXIX.6 Inserting footnote	319
XXIX.7 Other	319
XXX Arrays and tables	320
XXX.1 Preamble: macro as environment	320
XXX.2 Tabular environments	323
XXX.2.1 Disabling and restoring commands	323
XXX.2.2 Counters, boxes and lengths	327
XXX.2.3 Tabular typesetting	330
XXX.2.4 Environments	341
XXXI Quotation's commands	342
XXXII Section's title commands	343
XXXII.1 Commands to disable some feature	343
XXXII.2 General overview	343
XXXII.3 \beforeeledchapter command	344
XXXII.4 Auxiliary commands	344
XXXII.5 Patching standard commands	345
XXXII.6 Main code of \eledxxx commands	350
XXXII.7 Macros written in the auxiliary file	353
XXXIII Page breaking or no page breaking depending of specific lines	355
XXXIV Long verse: prevents being separated by a page break	357
XXXV Tools for hyperref package	357
XXXVI Compatibility with eledmac	358
Appendix A Things to do when changing versions	361
Appendix A.1 Migrating from edmac to ledmac	361
Appendix A.2 Migration from ledmac to eledmac	362
Appendix A.3 Migration to eledmac 1.5.1	363
Appendix A.4 Migration to eledmac 1.12.0	363
Appendix A.5 Migration to eledmac 17.1	364
Appendix A.6 Migration to eledmac 1.21.0	364
Appendix A.6.1 \Xledsetnormalparstuff and \ledsetnormalparstuffX	364
Appendix A.6.2 Endnotes	364
Appendix A.7 Migration to eledmac 1.22.0	364
Appendix A.8 Migration to eledmac 1.23.0	364

Appendix A.9 Migration from <code>eledmac</code> to <code>reledmac</code>	365
Appendix A.9.1 Risk of ‘no room for a new’	365
Appendix A.9.2 Multiple indices with <code>memoir</code>	365
Appendix A.9.3 Deprecated commands and options	365
Appendix A.9.4 <code>\renewcommand</code> replaced by <code>command</code>	366
Appendix A.9.5 Commands the names of which have been changed	366
Appendix A.9.6 Endnotes	368
Appendix A.9.7 Z Series	368
Appendix A.9.8 Internal commands	368
Appendix A.10 Migration to <code>reledmac</code> 2.1.0	368
Appendix A.11 Migration to <code>reledmac</code> 2.1.3	368
Appendix A.12 Migration to <code>reledmac</code> 2.3.0	368
Appendix A.13 Migration to <code>reledmac</code> 2.4.0	369
Appendix A.14 Migration to <code>reledmac</code> 2.5.0	369
Appendix A.15 Migration to <code>reledmac</code> 2.7.0	369
Appendix A.16 Migration to <code>reledmac</code> 2.7.2	369
Appendix A.17 Migration to <code>reledmac</code> 2.8.0	369
Appendix A.18 Migration to <code>reledmac</code> 2.13.1	369
Appendix A.19 Migration to <code>reledmac</code> 2.18.0	370
Appendix A.20 Migration to <code>reledmac</code> 2.21.0	370
References	371
Index	371
Change History	420

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 `Elɛdmac` 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

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).

were both of great help in ironing out last-minute wrinkles, while Ron made some important suggestions which may help to make future versions of `edmac` even more efficient. Wayne Sullivan, in particular, provided several important fixes and contributions, including adapting the Mittelbach/Schöpf ‘New Font Selection Scheme’ for use with `PLAIN TEX` and `edmac`. Another project Wayne has worked on is a DVI post-processor which works with an `edmac` that has been slightly modified to output `\specials`. This combination enables you to recover to some extent the text of each line as ASCII code, facilitating the creation of concordances, an *index verborum*, etc.

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

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

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

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

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

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

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

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

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

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

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

¹²Being prepared at Poona and Lausanne Universities.

February 2004. Sidenotes and regular footnotes in numbered text were added in April 2004. This port was called `ledmac` (\LaTeX `edmac`).

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

1.2.3 `eledmac`

Important changes were put in version 1.0, to make `ledmac` more easily extensible (see 7 p. 37). 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. 362.

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. 365.

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. 49.

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. 366).

swcaseinsensitive make `\sameword` command case insensitive.

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

nopbinverse prevents page break within verse environment.

noquotation by default, the quotation environment is redefined within numbered text. You can disable this redefinition with `noquotation` (see 17 p. 69).

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. 63).

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. 68).

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. 55). However, such features use up computer memory, at the expense of other processing needs. So if you do not need this feature, use `noledgroup` option. This should make `reledmac` faster.

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

5 Text lines and paragraphs numbering

5.1 Text lines numbering

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

```
\beginnumbering
Text
\endnumbering
```


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

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

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

5.2 Paragraphs

5.2.1 Basics

`\pstart` Within a numbered section, each paragraph of numbered text must be marked using the `\pend` `\pstart` and `\pend` commands like this:

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

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

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

```

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

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

The lines of this paragraph are
not numbered.

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

```

5.2.2 Automatically producing \pstart ... \pend

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

```

\begingroup
  \beginnumbering
  \autopar

  A paragraph of numbered text.

  Another paragraph of numbered
  text.

  \endnumbering
\endgroup

```

`\autopar` fails, however, on paragraphs that start with a `{` or with any other command that starts a new group before it generates any text. Such paragraphs need to be started explicitly, before the new group is opened, using `\indent`, `\noindent`, or `\leavevmode`, or using `\pstart` itself.¹³

5.2.3 Content before specific \pstart and after specific \pend

Both `\pstart` and `\pend` can take a optional argument in brackets. Its content will be printed before the beginning of `\pstart` / after the end of `\pend` instead of the argument of `\AtEveryPstart` / `\AtEveryPend`.

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

Note that a `\noindent` will be automatically added before this argument, and, consequently, a `\parskip` will be inserted. You can use a second optional argument, in brackets, to not have this `\noindent`.

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

The second optional argument of `\pstart` / `\pend` replace 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. 50) or `reledpar` (see 19.2 p. 73).

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.

Note that a `\noindent` will be inserted before the argument, and, consequently, a `\parskip`. You can use the starred version of `\AtEveryPstart` and `\AtEveryPend` to no insert the `\noindent`.

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

```
\AtStartEveryPstart{*}
```

Instead of manually doing

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

5.2.5 Numbering paragraphs (`\pstart`)

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

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

On each `\beginnumbering` the numbering restarts.

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

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

5.2.6 Languages written in Right to Left

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

5.2.7 Memory limits

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

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

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

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

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

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

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

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

5.3 Lineation commands

5.3.1 Disabling lineation

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

5.3.2 Setting lineation start and step

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

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

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

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

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

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

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

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

5.3.3 Setting lineation reset

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

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

5.3.4 Setting line number margin

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

```
\linenummargin{left}
```

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

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

5.3.5 Other settings

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

5.4 Changing the line numbers

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

5.4.1 Sublineation

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

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

You can change the separator between line number and subline number either by using `\Xsublinesep` without any optional argument (7.2.9 p. 41) or by using `\Xsublinesepside`. But in the second case, it will change the separator only for line numbers in the margins, not in the footnotes.

5.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current
`\endlock` value, until you insert `\endlock`. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines. But in this case you may use the `\stanza` mechanism, see 9 p. 50.

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

5.4.3 Setting and changing line number

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

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

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

5.4.4 Line number style

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

Alph Uppercase letters (A ... Z).

alph Lowercase letters (a ... z).

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

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

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

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

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

5.4.5 Skipping and hiding number

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

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

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

5.5 Executing code at each line

`\dolinehook`
`\doinsidelinehook`

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

6 Apparatus commands

6.1 Terminology

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

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

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

So :

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

6.2 Critical notes

6.2.1 The lemma

`\edtext`

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

`\edtext{⟨lemma⟩}{⟨commands⟩}`

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

For example:

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

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

The $\langle lemma \rangle$ 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 \edtext{Smith}{\Afootnote{Jones C, D.}} on Tuesday.}{ \Bfootnote{The date was July 16, 1954.} }</pre>	<pre>1 I am happy : I saw my friend Smith on 2 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 $\langle lemma \rangle$ 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, $\langle commands \rangle$, 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{ $\langle text \rangle$ }`. When all of the six are used, the A notes appear in a layer just below the main text, followed by the rest in turn, down to the E notes at the bottom.

`\Bfootnote` These are the main macros that you will use to construct the critical apparatus of your text.

`\Cfootnote`

`\Dfootnote`

`\Efootnote`

If you need more series of critical notes, please look at 6.7.1 p. 35.

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. 39).
- `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. 43).
- `nosep` to disable the lemma separator for this note.
- `linerangesep= $\langle c \rangle$` to change to $\langle c \rangle$ the separator between start line and end line for this particular note.

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

6.2.3 Endnotes

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

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

`\Cendnote`

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

`\Eendnote` `\Aendnote[$\langle option \rangle$]{ $\langle text \rangle$ }`.

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

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 7.2.5 p. 39).
- `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. 44).
- `linangesep= $\langle c \rangle$` to change to $\langle c \rangle$ the separator between start line and end line for this particular note.

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

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

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

Note that by default inside endnotes no separator is used between the lemma and the content. However you can use the `\Xendlemmaseparator` macro to define one (7.5.2 p. 44).

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

toendnotes **Code between endnotes** Sometimes, it is useful to insert content between endnotes of the same series: for example to separate endnotes of different sections of the same text. In this case, you could use *inside numbered text* the command:

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

For example:

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

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

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

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

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

6.2.4 Paragraph in critical apparatus

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

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

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

6.2.5 Change lemma and line number

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

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

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

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

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

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

This command does not change the marginal line numbers in any way; it just changes the numbers passed to the notes. Its use comes in situations that `\edtext` has trouble dealing with for whatever reason. If you need notes for overlapping passages that aren't nested, for instance, you can use `\lemma` and `\linenum` to generate such notes despite the limitations of `\edtext`. If the *⟨lemma⟩* argument to `\edtext` is extremely long, you may run out of memory; here again you can specify a note with an abbreviated lemma using `\lemma` and `\linenum`. The numbers used in `\linenum` need not be entered manually; you can use the 'x-' symbolic cross-referencing commands below (11 p. 55) to compute them automatically.

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

6.2.6 Changing the names of commands for critical apparatus

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

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

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

6.3 Disambiguation of identical words in the apparatus

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

6.3.1 Basic use

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

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

with the `\sameword` command:

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

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

6.3.2 Case setting

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

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

6.3.3 Notes about input encoding with UTF-8 processor

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

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

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

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

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

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

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

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

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

6.3.4 Use with `\lemma` command

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

For example in the following example:

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

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

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

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

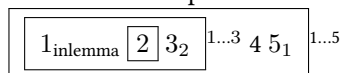
Note that only words that are actually referenced in a `\lemma` need the optional argument. Therefore, the first `\sameword` in the example above should have “1” as its optional argument, to be referenced correctly in the lemma.

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

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

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

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



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

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

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

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

Here, the corresponding code:

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

1	A (1) A (2) A (3) A (4) A (5)
1	A ¹ ...A ⁵] level 1
1	A ¹ ...A ³] level 2
1	A ² (2)] level 3

6.3.5 Sameword for a group of words

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

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

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

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

1 per causam tamen scire causam est per causam cognoscere causam

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

Here, it is not ambiguous what “per causam” refers to.

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

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

1 per causam tamen scire causam est per causam cognoscere causam

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

6.3.6 Customizing

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

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

6.3.7 Automatic sameword annotation

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

Please note that the maintainer of this script is not the maintainer of *reledmac*.

6.4 Apparatus of Manuscripts

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

To produce an apparatus of manuscripts with `reledmac`, you have to insert specific commands that are used to mark the sections for which only part of the manuscripts are relevant. These commands will be processed, and **after the second \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. 37). 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 L ^A T _E X footnotes generated via <code>\footnote</code> , the package also provides five series of additional footnotes called <code>\footnoteA</code> through <code>\footnoteE</code> . These
<code>\footnoteB</code>	have the familiar marker in the text, and the marked text at the foot of the page can be
<code>\footnoteC</code>	formatted using any of the styles described for the critical footnotes. Note that the 'reg-
<code>\footnoteD</code>	ular' footnotes have the series letter at the end of the macro name whereas the critical
<code>\footnoteE</code>	footnotes have the series letter at the start of the name.

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>\renewcommand*{\thefootnoteA}{\arabic{footnoteA}}</code>
<code>\footfootmarkA</code>	The appearance of the mark in the text is controlled by <code>\bodyfootmarkA</code> which is defined as: <code>\newcommand*{\bodyfootmarkA}{%</code> <code>\hbox{\textsuperscript{\normalfont\@nameuse{@thefnmarkA}}}}</code> The command <code>\footfootmarkA</code> controls the appearance of the mark at the start of the footnote text. It is defined as: <code>\newcommand*{\footfootmarkA}{\textsuperscript{\@nameuse{@thefnmarkA}}}</code>

There are similar command triples for the other series.

6.5.3 Separator for multiple footnotes

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

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

6.6 Printing the footnote mark without printing the footnote text

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

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

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

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

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

6.7 Changing series

6.7.1 Create a new series

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

6.7.2 Delete series

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

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

6.7.3 Series order

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

`\seriesatbegin`
`\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.

6.8 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:

`<series><type>`

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

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

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

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

7 Critical apparatus appearance

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

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

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

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

There is also name convention:

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

7.1 Notes arrangement in a series

`\Xarrangement`
`\arrangementX`

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

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

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

- `paragraph` formats all 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. 71.
- `twocol` formats them as separate paragraphs, but in two columns;
- `threecol`, in three columns.
- `normal`, restore normal arrangement.

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

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

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

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

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

7.2 Control line number printing

7.2.1 Print line number only at first time

`\Xnumberonlyfirstinline`

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

`\Xnumberonlyfirstintwolines`

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

`\Xsymlinenum`

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

`\Xendnumberonlyfirstinline`

`\Xendnumberonlyfirstintwolines`

`\Xendsymlinenum`

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

7.2.2 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.

`\Xendpagenumberonlyfirst`

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

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

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

In any case, you can use:

`\Xendsympagenum`

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

`\Xendinplaceofpagenumber`

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

7.2.3 Arbitrary text before line number

`\Xbeforenumber`

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

7.2.4 Separator for line range

`\Xlinrangeseparator`

`\Xendlinrangeseparator`

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 `\Xlinrangeseparator[⟨s⟩]{⟨text⟩}`, and with `\Xendlinrangeseparator[⟨s⟩]{⟨text⟩}` for critical endnotes.

7.2.5 Abbreviate line range

`\Xtwolines`

`\Xmorethantwolines`

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

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

```
\Xtwolines{sq.}
```

```
\Xmorethantwolines{sqq.}
```

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

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

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

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

`\Xtwolinesonlyinsamepage`

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

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

`\Xendtwolines`

`\Xendmorethantwolines`

`\Xendtwolinesbutnotmore`

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

7.2.6 Disable line number

`\Xnonumber`

`\Xendnonumber`

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

7.2.7 Printing pstart number

`\Xpstart`

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

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

`\Xpstarteverytime`

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

`\Xonlypstart`

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

7.2.8 Printing stanza number

`\Xstanza`

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

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

`\Xstanzaseparator`

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

7.2.9 Separator between line and subline numbers

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

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

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

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

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

7.2.10 Separator between page and line numbers

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

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

7.2.11 Space around number

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

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

`\Xendbeforenumber` `\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

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

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

`\Xendbeforeasymlinenumber` `\Xendbeforeasymlinenumber` and `\Xendaftersymlinenumber` are the equivalents of `\Xbeforeasymlinenumber` and `\Xaftersymlinenumber` 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}
```

`\Xboxsymlinenum` `\Xboxsymlinenum[⟨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⟩}`

`\Xendboxlinenumalign` `\Xendboxendlinenum[⟨s⟩]{⟨l⟩}` are the same as, respectively, `\Xboxlinenum` and

`\Xendboxstartlinenumalign` `\Xboxlinenumalign`, `\Xboxstartlinenum`, `\Xboxendlinenum` except in endnotes.

`\Xendboxendlinenumalign`

7.3 For endnotes

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

`\Xendafterpagenumber` `\Xendafterpagenumber[\langle s \rangle]{\langle text \rangle}` defines the text after the page number in endnotes. Default value is) (open parenthesis followed by a single space).

`\Xendlineprefixsingle` `\Xendlineprefixsingle[\langle s \rangle]{\langle text \rangle}` defines the text before the line number in endnotes, when there is only one line. Default value is empty.

`\Xendlineprefixmore` `\Xendlineprefixmore[\langle s \rangle]{\langle text \rangle}` 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[\langle s \rangle]{\langle code \rangle}` is used to execute code before line number in endnotes. The code is executed before the `\Xendbeforelinenumber` space and before the `\Xendnotenumfont` font setting.

`\Xendahooklinenumber` `\Xendahooklinenumber[\langle s \rangle]{\langle code \rangle}` is used to execute code after line number in endnotes. The code is executed after the `\Xendafternumber` space.

`\Xendbhookinplaceofnumber` `\Xendbhookinplaceofnumber[\langle s \rangle]{\langle code \rangle}` is used to execute code before space or symbol which replace line number in endnotes. The code is executed before the `\Xendbeforesymlinenum` space and before the `\Xendnotenumfont` font setting.

`\Xendahookinplaceofnumber` `\Xendahookinplaceofnumber[\langle s \rangle]{\langle code \rangle}` is used to execute code after space or symbol which replace line number in endnotes. The code is executed after the `\Xendaftersymlinenum` space.

7.5 Separator between the lemma and the note

7.5.1 For footnotes

`\Xlemmaseparator` By default, in a footnote, the separator between the lemma and the note is a right bracket (`\rbracket`)¹⁸. You can use `\Xlemmaseparator[\langle s \rangle]{\langle Xlemmaseparator \rangle}` to change it. The optional argument can be used to specify the series in which it is used. Note that there is a non-breakable space between the lemma and the separator, but a **breakable** space between the separator and the following text.

`\Xbeforelemmaseparator` Using `\Xbeforelemmaseparator[\langle s \rangle]{\langle l \rangle}` you can add some space between lemma and separator. If your lemma separator is empty, this space won’t be printed. The default value is 0 em.

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

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

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

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

7.5.2 For endnotes

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

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

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

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

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

7.6 Font style

7.6.1 For line number

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

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

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

7.6.2 For the lemma

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

`\Xendlemmafont`

`\Xlemmafont{\bfseries}`

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

7.6.3 For all notes

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

7.7 Wrapping notes

7.7.1 Wrapping lemmas

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

7.7.2 Wrapping contents

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

7.8 Indent of notes content

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

7.9 Arbitrary code at the beginning of notes

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

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

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

7.10 Arbitrary code before inserting note

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

They allow one to add any arbitrary code just before the footnotes are added in the list of footnotes. The main use is to insert text direction code. For example, if you edit right-to-left text with bidi, but want your critical footnote be left-to-right, use `\Xbeforeinserting\LTR`. You should also use `\Xwraplemma` to ensure your lemmas are right-to-left in a left-to-right paragraph (7.7.1 p. 45)).

Note that the changes are local to the footnote.

7.11 Options for footnotes in columns

7.11.1 Alignment

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

<code> must be one of the following command:

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

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

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

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

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

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

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

7.11.2 Size of the columns

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

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

7.12 Options for paragraphed footnotes and notes grouped by line

7.12.1 Mark separation of notes

<code>\Xafternote</code>	You can add some horizontal space after a note by using <code>\Xafternote[⟨s⟩]{⟨l⟩}</code> (for critical footnotes) or <code>\afternoteX[⟨s⟩]{⟨l⟩}</code> (for familiar footnotes). The default value is <code>1em plus .4em minus .4em</code> .
<code>\afternoteX</code>	
<code>\Xparafootsep</code>	For paragraphed footnotes (see below), you can choose the separator between each note by using <code>\Xparafootsep[⟨s⟩]{⟨text⟩}</code> for critical notes and <code>\parafootsepX</code> for familiar notes. A common separator is the double pipe (<code> </code>), which you can set by using <code>\Xparafootsep{\$\parallel\$}</code> .
<code>\parafootsepX</code>	
Note that if the symbol defined by <code>\Xsymlinenum</code> must be used at the beginning of a note, the <code>\Xparafootsep</code> / <code>\parafootsepX</code> is not used before this note.	

7.12.2 Ragged text

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

7.13 Options for block of notes

7.13.1 Grouping notes by line

<code>\Xgroupbyline</code>	If you do not use <code>\Xarrangement{paragraph}</code> , you may want to group all the critical footnotes related to the same line in the same paragraph. In this case, use <code>\Xgroupbyline[⟨series⟩]</code> . In many cases, you might like to use it in combination with <code>\Xnumberonlyfirstinline</code> (7.2.1 p. 38). Note that the <code>\Xafternote</code> and <code>\Xparafootsep</code> settings are used to determine space and content between footnotes (7.12 p. 47). Suppose you have two notes on line 1 which overlap lines 1 and 2. This last note will be printed, if you use <code>\Xgroupbyline</code>
<code>\groupbylineseparetwolines</code>	

in the same group as the previous one. In the case you want that note to be distinct, you must use both `\Xgroupbyline` and `\Xgroupbylineseparetwelines[⟨s⟩]`.

In many cases, you might like to use it in combination with `\Xnumberonlyfirstintwelines` (7.2.1 p. 38)

7.13.2 Text before notes

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

7.13.3 Code before notes

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

7.13.4 Spacing

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

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

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

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

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

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

7.13.5 Rule

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

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

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

Be careful, the standard \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[$\langle s \rangle$]{ $\langle l \rangle$ }`. 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[$\langle s \rangle$]{ $\langle l \rangle$ }` 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[$\langle s \rangle$]{ $\langle l \rangle$ }` sets the total width of critical footnotes. `\widthX[$\langle s \rangle$]{ $\langle l \rangle$ }` does the same for familiar footnotes.

$\langle l \rangle$ can be a length expression, parsable with `\dimexpr`. For example:

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

Note that changes the with of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.11.2 p. 47.

7.14 Footnotes and the `reledpar` columns

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

`\noteswidthliketwocolumnsX`

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

7.15 Endnotes in one paragraph

- `\Xendparagraph` By default, any new endnote starts a new paragraph. Use `\Xendparagraph[⟨s⟩]` to have all end notes of one given series set in one paragraph.
- `\Xendafternote` You can add some space after a endnote series by using `\Xendafternote[⟨s⟩]{⟨l⟩}`. The default value is `1em plus.4em minus.4em`.
- `\Xendsep` You can choose the separator between each note by `\Xendsep[⟨s⟩]{⟨text⟩}`. A common separator is the double pipe (`||`), which you can set by using `\Xendsep{\parallel}`.

8 Fonts

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

- `\numlabfont` Line numbers for the main text are usually printed in a smaller font in the margin. The `\numlabfont` macro is provided as a standard name for that font: it is initially defined as
- ```
\newcommand{\numlabfont}{\normalfont\scriptsize}
```
- You might wish to use a different font if, for example, you preferred to have these line numbers printed using old-style numerals.
- `\select@lemmafont` We will briefly discuss `\select@lemmafont` here because it is important to know about it now, although it is not one of the macros you would expect to change in the course of a simple job. Hence it is ‘protected’ by having the `@`-sign in its name.
- When you use the `\edtext` macro to mark a word in your text as a lemma, that word will normally be printed again in your apparatus. If the word in the text happens to be in a font such as italic or bold you would probably expect it to appear in the apparatus in the same font. This becomes an absolute necessity if the font is actually a different script, such as Arabic or Cyrillic. `\select@lemmafont` does the work of decoding `reledmac`’s data about the fonts used to print the lemma in the main text and calling up those fonts for printing the lemma in the note.
- `\select@lemmafont` is a macro that takes one long argument—the cluster of line numbers passed to the note commands. This cluster ends with a code indicating what fonts were in use at the start of the lemma. `\select@lemmafont` selects the appropriate font for the note using that font specifier.
- `reledmac` uses `\select@lemmafont` in a standard footnote format macro called `\normalfootfmt`. The footnote formats for each of the layers A to E are `\let` equal to `\normalfootfmt`. So all the layers of the footnotes are formatted in the same way.

## 9 Verse

### 9.1 Basic

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

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

If you need to add brackets directly after `\stanza`, `&` or `\&`, add `\norelax`. Otherwise, the brackets will be interpreted as delimitation of an optional argument (cf. 9.8 p. 53)

## 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. 53.

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. 363.**

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{stanzaindentrepetition}{0}
```

The macros make no restriction on the number of lines in a stanza. Stanza indentation values (and penalty values) obey T<sub>E</sub>X's grouping conventions, so if one stanza among several has a different structure, its indentations (penalties) may be set within a group; the prior values will be restored when the group ends.

## 9.4 Manual stanza indent

```
\stanzaindent
\stanzaindent*
```

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

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

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

## 9.5 Stanza breaking

```
\setstanzapenalties
```

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

The command

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

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

The first entry “1” is a control value. If it is zero, then no penalties are passed on to T<sub>E</sub>X, 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. 69 for further details.

## 9.8 Content before/after verses

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

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

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

### `\AtEveryStanza`

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

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

### `\AtStartEveryStanza`

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

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

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

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

- Use `\AtEveryPend{<arg>}` to automatically add content after verses (including the final one) and `\AtEveryPstart{<arg>}` to automatically add content before verses (including the first one).
- `\&` can take an optional argument (in brackets). Its content will be printed after the stanza.

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

## 9.9 Numbering stanza

- `\numberstanzatrue` If you want to automatically number stanzas, use `\numberstanzatrue`. In this case, the line number will restart at each `\stanza`.
- `\numberstanzafalse` If you want to disable this feature again, use `\numberstanzafalse`.  
You can use this feature in combination with `\Xstanza` (7.2.8 p. 40).
- `\thestanza` You can redefine `\thestanza` to change the aspect of stanza number. Default value is:  

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

- `\stanzanumwrapper` You can change the value of the stanza counter with the usual commands of  $\text{\LaTeX}$ .  
You can redefine `\stanzanumwrapper` in order to modify the way the stanza number is inserted in the flow of text. Default value is:  

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

## 9.10 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. 53).

## 10 Grouping

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

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

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

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

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

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

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

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

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

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

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

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

## 11 Cross referencing

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

### 11.1 Basic use

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

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

<sup>19</sup>More precisely, you should stick to characters in the  $\text{\TeX}$  categories of “letter” and “other”.

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

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

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

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

## 11.2 Cross-referencing to a critical note

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

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

If you want to refer to starting and ending lines, you should use `\appref` and related tools (11.6.2 p. 58).

## 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 `ℒTEX` 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.

For example, one might use the following:

```
\beginnumbering
```

```
\pstart
```

```
\edlabel{Queritur}Queritur utrum metaphysica sit scientia una.
```

```
\pend
```

```
\pstart
```

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

```
\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<sup>A</sup>T<sub>E</sub>X cross-referencing

`\label`

`\ref`

`\pageref`

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

## 11.6 References to start and end lines

### 11.6.1 Reference to main text lines

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

`\edlabelS`

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

|                             |                                                                                                                                                                                                                                          |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\edlabelE</code>      | Use <code>\edlabelE{&lt;label&gt;}</code> to mark the end the end line of the passage. These two commands just create to label which are named <code>&lt;label&gt;:start</code> and <code>&lt;label&gt;:end</code> .                     |
| <code>\edlabelSE</code>     | Use <code>\edlabelSE{&lt;label&gt;}</code> to mark just one location in the text. Contrary to a classical <code>\edlabel</code> , the <code>&lt;label&gt;</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{&lt;label&gt;}</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 endnotes.                                                                                                                                     |
| <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.

|                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\applabel</code>                               | If you use <code>\applabel{&lt;label&gt;}</code> inside the second argument of a <code>\edtext</code> , <code>reledmac</code> will add a <code>\edlabel</code> at the beginning and end of the marked passage. The label at the beginning of the passage will have the title <code>&lt;label&gt;:start</code> , while the label at the end will have the title <code>&lt;label&gt;:end</code> .<br>If you use <code>\linenum</code> (6.2.5 p. 27) to refer to these labels, <code>reledmac</code> will use your line settings to refer to the passage. |
| <code>\appref</code><br><code>\apprefwithpage</code> | You can also use <code>\appref{&lt;label&gt;}</code> and <code>\apprefwithpage{&lt;label&gt;}</code> 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

|                                                                          |                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\setapprefprefixsingle</code><br><code>\setapprefprefixmore</code> | <b>Specific to these tools</b> If you use <code>\apprefprefixsingle{&lt;prefix&gt;}</code> , <code>&lt;prefix&gt;</code> will be printed before the line numbers of a <code>\appref</code> -reference. If you use <code>\apprefprefixmore{&lt;prefix&gt;}</code> , <code>&lt;prefix&gt;</code> 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.

|                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\setSerefprefixsingle</code><br><code>\setSerefprefixmore</code><br><code>\setSerefonlypageprefixsingle</code><br><code>\setSerefonlypageprefixmore</code> | <code>\setSerefprefixsingle</code> and <code>\setSerefprefixmore</code> are similar for <code>\Seref</code> command.<br>Use <code>\setSerefonlypageprefixsingle{&lt;prefix&gt;}</code> to set the page prefix for <code>\Serefonlypage</code> when there is only one page. Use <code>\setSerefonlypageprefixmore{&lt;prefix&gt;}</code> 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. 59). 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 commandes 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. 25).

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

## 11.7 Compatibility with xr package

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

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

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

## 12 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)  
`\ledrsnotesep` margin. These lengths are initially set to the value of `\linenumsep`.

### 12.2.4 Font

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

These can of course be changed to suit.

### 12.2.5 Separator between notes

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

## 13 Indexing

### 13.1 Basics

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

- is the default separator used by the MAKEINDEX program.

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

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

However, you also have to configure your `.ist` index style file. For example if you use `:` as separator<sup>20</sup>.

```
page_compositor ":"
```

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

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

## 13.4 Using xindy

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

`pagenumber-linenummer`

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

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

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

If you 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.<sup>22</sup>

## 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.

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

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

## 14.1 Preamble setting

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

- Or set `.glossaries`:  
`\glsSetCompositor{-}`
- Or set `reledmac`:  
`\renewcommand{\pagelinesep}{-}`  
 In this case, the above will have consequences for your use of `\edindex` and you should set your `.ist` file (13.3 p. 62).

## 14.2 Commands

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

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

## 15 Tabular material

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

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

```

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

```

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

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

For example:

```

\beginnumbering
\pstart
\begin{edtabularl}

```



```

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

```

produces the following parallel pair of verses.

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

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

|                               |                                  |                      |
|-------------------------------|----------------------------------|----------------------|
| <code>\begin{edarrayl}</code> | 1 & 2 & & 3 & & 4 & \\           | 1 & 2 & 3 & 4        |
|                               | & \spreadmath{F+G+C} & & & & & & | $F + G + C$          |
|                               | a & bb & & ccc & & dddd          | <i>a bb ccc dddd</i> |
| <code>\end{edarrayl}</code>   |                                  |                      |

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

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

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

```

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

```

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

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

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

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

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

|   |    |   |   |
|---|----|---|---|
| 1 | 2  | 3 | 4 |
| a | └─ |   | d |
| A | B  | C | D |

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

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

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

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

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

For example:

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

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

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

**\edvertdots**

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

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

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

## 16 Sectioning commands

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

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

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

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

## 16.2 Sectioning commands with line numbering and critical notes

You have to use the following commands:

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

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

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

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

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

## 16.3 Optimization

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

## 17 Quotation environments

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

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

## 18 Page breaks

### 18.1 Control page breaking

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

`\ledpb`  
`\lednopb`

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

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

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

19.1 Known and suspected limitations

19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.13.6 p. 49 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. 68 in order to know which are the available options.

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

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

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

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

19.1.4 Marginal notes

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

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

19.1.5 Paragraph shape

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

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

19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.1 p. 37, and described in more detail on XII.6.3 p. 177, 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. 49), in order to define the maximum height relative to `\textheight` and not to `\vsize`, because the `\vsize` value is not the same inside and outside of the preamble.

19.1.7 Use with other packages

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

`\morenoexpands`

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

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

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

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

or

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

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

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

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

```
\makeatletter
```

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


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

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

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

19.2 Parallel typesetting

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

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

I Implementation overview

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

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

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

Next are commands for marking sections of the text for footnotes (Section VI), followed by the macros that take each paragraph apart, attach the line numbers and insertions, and send the result to the vertical list (Section VII). The footnote commands (Section 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}[2017/07/31 v2.23.0 typesetting critical editions
4 ]%
5 %

```

II.3 Package options

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

```

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

```

The `parledgroup` option is for `reledpar`. However, it has consequence on `reledmac` internal command. So we need to define the boolean now.

```

7 \newif\ifparledgroup
8 %

```

And now, the options of `reledmac`.

```

\DeclareOptionX{series}[A,B,C,D,E]{\xdef\default@series{#1}}
\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
78 \newif\ifsw@caseinsensitive%
79 \DeclareOptionX{swcaseinsensitive}{%
80   \sw@caseinsensitivetrue%
81 }%
82 %

```

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.

```
83 \ProcessOptionsX*\relax
84
85 %
```

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 `LuaTeX` or `XYTeX` is running, and `ragged2e` to manage ragged justification for paragraphed notes.

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

II.5 Compatibility with LuaTeX

Here, we enable some primitives for `LuaTeX`.

```
103 \ifx\directlua\undefined\else%
104 \directlua{tex.enableprimitives("",{"texdir","pardir","bodydir"})}
105 \fi
106 %
```

II.6 Boolean flags

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

```

107 \newif\ifl@dmemoir
108 \@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
109
110 %

```

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

```

111 \newif\if@ledgroup%
112 %

```

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

```

113 \newif\ifl@imakeidx
114 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value
115 %

```

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

```

116 \newif\ifl@indextools%
117 \@ifpackageloaded{indextools}{%
118   \l@indextoolstrue%
119   \l@imakeidxtrue%
120   \let\imki@wrindexentry\indtl@wrindexentry%
121 }{}%
122 %

```

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.

```

123 \newif\ifl@footmisc
124 \@ifpackageloaded{footmisc}{\l@footmisctrue}{}%False is the default value
125 %

```

\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.

```

126 \ifdef{\if@RTL}{\newif\if@RTL}
127 %

```

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

```

128 \newif\if@firstlineofpage%
129 \newif\if@firstlineofpageR%
130 %

```

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.

```
131 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
132 %
```

`\reledmac@error` Write an error message.

```
133 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
134 %
```

```
\led@err@NumberingStarted 135 \newcommand*{\led@err@NumberingStarted}{%
d@err@NumberingNotStarted 136 \reledmac@error{Numbering has already been started}{\@ehc}}
NumberingShouldHaveStarted 137 \newcommand*{\led@err@NumberingNotStarted}{%
138 \reledmac@error{Numbering was not started}{\@ehc}}
139 \newcommand*{\led@err@NumberingShouldHaveStarted}{%
140 \reledmac@error{Numbering should already have been started}{\@ehc}}
141 %
```

```
d@err@edtextoutsidestart 142 \newcommand*{\led@err@edtextoutsidestart}{%
143 \reledmac@error{\string\edtext\space outside numbered paragraph (\...pstart
\pend)}{\@ehc}}%
144 %
```

```
\led@mess@NotesChanged 145 \newcommand*{\led@mess@NotesChanged}{%
146 \typeout{reledmac reminder: }%
147 \typeout{ The number of the footnotes in this section
148 has changed since the last run.}%
149 \typeout{ You will need to run LaTeX two more times
150 before the footnote placement}%
151 \typeout{ and line numbering in this section are
152 correct.}}
153 %
```

```
\led@mess@SectionContinued 154 \newcommand*{\led@mess@SectionContinued}[1]{%
155 \message{Section #1 (continuing the previous section)}}
156 %
```

```
d@err@LineationInNumbered 157 \newcommand*{\led@err@LineationInNumbered}{%
158 \reledmac@error{You can't use \string\lineation\space within
159 a numbered section}{\@ehc}}
160 %
```

```

\led@warn@BadLineation61 \newcommand*\led@warn@BadLineation}{%
\led@warn@BadLinenummargin62 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp63 \newcommand*\led@warn@BadLinenummargin}{%
\led@warn@BadSublockdisp64 \reledmac@warning{Bad \string\linenummargin\space argument}}
165 \newcommand*\led@warn@BadLockdisp}{%
166 \reledmac@warning{Bad \string\lockdisp\space argument}}
167 \newcommand*\led@warn@BadSublockdisp}{%
168 \reledmac@warning{Bad \string\sublockdisp\space argument}}
169 %

\led@warn@NoFile70 \newcommand*\led@warn@NoFile}[1]{%
171 \reledmac@warning{File `#1' not found}}
172 %

\led@warn@LineFileObsolete73 \newcommand*\led@warn@Obsolete}[1]{%
174 \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
175 %

\led@warn@BadAdvancelineSubline76 \newcommand*\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine77 \reledmac@warning{\string\advanceline\space produced a sub-line
178 number less than zero.}}
179 \newcommand*\led@warn@BadAdvancelineLine}{%
180 \reledmac@warning{\string\advanceline\space produced a line
181 number less than zero.}}
182 %

\led@warn@BadSetline83 \newcommand*\led@warn@BadSetline}{%
\led@warn@BadSetlinenum84 \reledmac@warning{Bad \string\setline\space argument}}
185 \newcommand*\led@warn@BadSetlinenum}{%
186 \reledmac@warning{Bad \string\setlinenum\space argument}}
187 %

\led@err@PstartNotNumbered88 \newcommand*\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart89 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered90 numbered section %
\led@err@PendNoPstart91 (\string\...beginnumbering\string\endnumbering)}{\@ehc}}%
\led@err@AutoparNotNumbered92 \newcommand*\led@err@PstartInPstart}{%
\led@err@NumberingWithoutPstart93 \reledmac@error{\string\pstart\space encountered while another
194 \string\pstart\space was in effect}{\@ehc}}
195 \newcommand*\led@err@PendNotNumbered}{%
196 \reledmac@error{\string\pend\space must be used within a
197 numbered section}{\@ehc}}
198 \newcommand*\led@err@PendNoPstart}{%

```



```

199 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
200 \newcommand*{\led@err@AutoparNotNumbered}{%
201 \reledmac@error{\string\autopar\space must be used within a
202 numbered section}{\@ehc}}
203 \newcommand*{\led@err@NumberingWithoutPstart}{%
204 \reledmac@error{\string\beginnumbering...\string\endnumbering\space
without \string\pstart}{\@ehc}}%
205 %

```

```

\led@warn@BadAction06 \newcommand*{\led@warn@BadAction}{%
207 \reledmac@warning{Bad action code, value \next@action.}}
208 %

```

```

\led@warn@DuplicateLabel09 \newcommand*{\led@warn@DuplicateLabel}[1]{%
ppLabelOutSecondArgEdtext10 \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined11 \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined12 }%
213 \newcommand*{\led@warn@AppLabelOutSecondArgEdtext}[1]{%
214 \reledmac@warning{\string\applabel\space outside of the second argument
of an \string\edtext\space `#1' on page \thepage.}}%
215 \newcommand*{\led@warn@RefUndefined}[1]{%
216 \G@refundefinedtrue%
217 \reledmac@warning{Reference `#1' on page \thepage\space undefined.%
218 Using `000'.}%
219 \@latex@warning{Reference `#1' undefined\on@line}%
220 }%
221 \newcommand*{\led@warn@pairRefUndefined}[1]{%
222 \G@refundefinedtrue%
223 \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \thepage\
space undefined.
224 Using `??'.}%
225 \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
226 }
227 %

```

```

\led@warn@NoMarginpars28 \newcommand*{\led@warn@NoMarginpars}{%
229 \reledmac@warning{You can't use \string\marginpar\space in numbered text
}}
230 %

```

```

\led@warn@BadSidenotemargin31 \newcommand*{\led@warn@BadSidenotemargin}{%
232 \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
233 %

```

```

\led@warn@NoIndexFile34 \newcommand*{\led@warn@NoIndexFile}[1]{%
235 \reledmac@warning{Undefined index file #1}}
236 %

\led@warn@SeriesStillExist37 \newcommand{\led@warn@SeriesStillExist}[1]{%
238 \reledmac@warning{Series #1 is still existing !}%
239 }%
240 %

\led@err@BadAction41 \newcommand*{\led@err@StanzaIndentNotDefined}{%
242 \reledmac@error{You have not defined the indentation for the line \number
\stanza@count}{\@ehc}}%
243 %

\led@err@ManySidenotes44 \newcommand{\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes45 \ifledRcol{%
\led@err@ManyRightnotes46 \reledmac@warning{\itemcount@space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
\else%
247 \reledmac@warning{\itemcount@space sidenotes on line \the\line@num\
space p. \the\page@num}%
248 \fi%
249 }%
250 \newcommand{\led@err@ManyLeftnotes}{%
251 \ifledRcol{%
252 \reledmac@warning{\itemcount@space leftnotes on line \the\line@numR\
space p. \the\page@numR}%
253 \else%
254 \reledmac@warning{\itemcount@space leftnotes on line \the\line@num\
space p. \the\page@num}%
255 \fi%
256 }%
257 \newcommand{\led@err@ManyRightnotes}{%
258 \ifledRcol{%
259 \reledmac@warning{\itemcount@space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
260 \else%
261 \reledmac@warning{\itemcount@space rightnotes on line \the\line@num\
space p. \the\page@num}%
262 \fi%
263 }%
264 }%
265 %

\led@err@TooManyColumns66 \newcommand*{\led@err@TooManyColumns}{%
\led@err@UnequalColumns67 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn
\led@err@HighEndColumn
\led@err@ReverseColumns

```

```

268 \newcommand*\led@err@UnequalColumns}{%
269   \reledmac@error{Number of columns is not equal to the number
270     in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
271 \newcommand*\led@err@LowStartColumn}{%
272   \reledmac@error{Start column is too low}{\@ehc}}
273 \newcommand*\led@err@HighEndColumn}{%
274   \reledmac@error{End column is too high}{\@ehc}}
275 \newcommand*\led@err@ReverseColumns}{%
276   \reledmac@error{Start column is greater than end column}{\@ehc}}
277 %

```

```

endnotes@outsidenumbering 278 \newcommand\led@err@toendnotes@outsidenumbering}{%
279   \reledmac@error{\string\toendnotes\space and related commands must be
    called inside a numbered text (\string\...beginnumbering\string\endnumbering
    )}{\@ehc}%
280 }%
281 %

```

```

err@EdtextWithoutFootnote 282 \newcommand\led@err@EdtextWithoutFootnote}{%
283   \reledmac@error{edtext without Xfootnote. Check syntax}{\@ehc}%
284 }%
285 %

```

```

noteNotInSecondArgEdtext 286 \newcommand\led@err@FootnoteNotInSecondArgEdtext}[1]{%
287   \reledmac@error{#1 footnote outside of the second argument of an edtext.
    Check syntax}{\@ehc}%
288 }%
289 %

```

```

error@PackageAfterEledmac 290 \newcommand\led@error@PackageAfterEledmac}[1]{%
291   \reledmac@error{#1 must be loaded before reledmac}{\@ehc}%
292 }%
293 %

```

```

error@fail@patch@@makecol 294 \newcommand\led@error@fail@patch@@makecol}{%
295   \reledmac@error{Fail to patch \string\@makecol\space command}{\@ehc}%
296 }%
297 %

```

```

ror@fail@patch@@reinserts 298 \newcommand\led@error@fail@patch@@reinserts}{%
299   \reledmac@error{Fail to patch \string\@reinserts\space command}{\@ehc}%
300 }%
301 %

```

```

\led@error@fail@patch@@doclearpage02 \newcommand{\led@error@fail@patch@@doclearpage}{%
303   \reledmac@error{Fail to patch \string\@doclearpage\space command}{\@ehc}%
304 }%
305 %

```

```

\led@error@fail@patch@@iiiminipage06 \newcommand{\led@error@fail@patch@@iiiminipage}{%
307   \reledmac@error{Fail to patch \string\@iiiminipage\space command}{\@ehc}%
308 }%
309 %

```

```

\led@error@fail@patch@endminipage10 \newcommand{\led@error@fail@patch@endminipage}{%
311   \reledmac@error{Failed to patch the \string\endminipage\space command}{\@ehc}%
312 }%
313 %

```

```

\led@error@fail@patch@endminipage14 \newcommand{\led@error@fail@patch@makeindex}{%
315   \reledmac@error{Failed to patch the \string\makeindex\space command}{\@ehc}%
316 }%
317 %

```

```

\led@warn@edinde@outsidenumbering18 \newcommand{\led@warn@edinde@outsidenumbering}{%
319   \reledmac@warning{\string\edindex\space called outside of \string\
...beginnumbering\string\endnumbering. \MessageBreak Automatically switched
to \string\index.}%
320 }%
321 %

```

```

\led@warning@hsizeX@deprecated22 \newcommand{\led@warning@hsizeX@deprecated}{%
323   \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
324 }%
325 %

```

```

\led@warning@Xhsize@deprecated26 \newcommand{\led@warning@Xhsize@deprecated}{%
327   \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
328 }%
329 %

```

```

warning@msdatawithoutstop 30 \newcommand{\led@warning@msdatawithoutstop}{%
331 \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
332 }%
333 %

ning@preXnotes@deprecated 34 \newcommand{\led@warning@preXnotes@deprecated}{%
335 \reledmac@warning@preXnotes@deprecated%
336 }%
337 %

```

II.8 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree 38 \providecommand*\@gobblethree}[3]{}
\@gobblefour 39 \providecommand*\@gobblefour}[4]{}
\@gobbleseven 40 \providecommand*\@gobbleseven}[7]{}
341 %

```

II.9 Miscellaneous commands

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

```

342 \ifledfinal
343 \newcommand*\showlemma[1]{#1}
344 \else
345 \newcommand*\showlemma[1]{\underline{#1}}
346 \fi
347
348 %

```

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

Initialize it as `\empty`.

```

349 \let\linenumberlist=\empty
350
351 %

```

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

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

```

352 \newcount\@l@tempcnta \newcount\@l@tempcntb
353 %

```

II.10 Prepare reledpar

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

`\ifl@dpaging`

`\ifl@dprintingpages`

`\ifl@dprintingcolumns`

```

354 \newif\ifl@dpairing
355 \newif\ifl@dpaging%
356 \newif\ifl@dprintingpages%
357 \newif\ifl@dprintingcolumns%
358 \newif\ifpst@rtedL
359 \newcount\l@dnumpstartsL
360 %

```

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

`\ifledRcol@`

```

361 \newif\ifledRcol
362 \newif\ifledRcol@
363 %

```

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

```

364 \newif\ifnumberingR
365 %

```

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

```

366 \newif\ifXnote@%
367 %

```

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

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

`\ifindtl@notenumber`

```

368 \providebool{indtl@innote}%
369 \providebool{indtl@notenumber}%
370 %

```

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 `\resumenumbers` 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.

```

371 \newcount\section@num
372 \section@num=0
373 \let\extensionchars=\empty
374 %

```

`\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.

```

375 \newif\ifnumbering
376 %

```

`\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.

```

377 \newcommand*{\beginnumbering}{%
378   \ifnumbering
379     \led@err@NumberingStarted
380   \endnumbering
381 \fi
382 \global\numberingtrue
383 \global\advance\section@num \@ne
384 \initnumbering@reg
385 \message{Section \the\section@num }%
386 \line@list@stuff{\jobname.\extensionchars\the\section@num}%

```

```

387 \l@dend@stuff
388 \setcounter{pstart}{1}
389 \ifl@dpairing
390   \global\l@dnumpstartsL \z@
391   \global\pst@rtedLfalse
392 %

```

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.

```

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

```



```

427 \resetprevline@
428 \resetprevpage@num
429 \global\stopmsdata@inserted@true%
430 \global\let\@msdata@list\relax%
431 \global\csundef{\@msdata@\add@msd@c @data}%
432 }
433
434 %

```

\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.

```

435 \def\endnumbering{%
436   \ifnumbering
437     \global\numberingfalse
438     \normal@pars
439     \ifnum\l@dnumstart=L=0%
440       \led@err@NumberingWithoutPstart%
441     \fi%
442     \ifl@dpairing
443       \global\pst@rtedL=false
444     \else
445       \ifx\insertlines@list\empty\else
446         \global\noteschanged@true
447       \fi
448       \ifx\line@list\empty\else
449         \global\noteschanged@true
450       \fi
451     \fi
452     \ifnoteschanged@
453       \led@mess@NotesChanged
454     \fi
455   \else
456     \led@err@NumberingNotStarted
457   \fi
458   \autoparfalse
459   \if@noeled@sec\else%
460     \immediate\closeout\eled@sectioning@out%
461   \fi%
462   \ifl@dpairing\else
463     \global\l@dnumstart=L=\z@%
464   \endgroup
465 \fi
466 }
467 %

```

\pausenumbering The \pausenumbering macro is just the same as \endnumbering, but with the **\resumenumbering**

`\ifnumbering` flag set to true, to show that numbering continues across the gap.²⁴

```

468 \newcommand{\pausenumbering}{%
469   \ifautopar\global\autopar@pausetrue\fi%
470   \endnumbering\global\numberingtrue}
471 %

```

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

```

472 \newcommand*{\resumenumbering}{%
473   \ifnumbering
474     \ifautopar@pause\autopar\fi
475     \global\pst@rtedLtrue
476     \global\advance\section@num \@ne
477     \led@mess@SectionContinued{\the\section@num}%
478     \line@list@stuff{\jobname.\extensionchars\the\section@num}%
479     \l@dend@stuff
480     \ifl@dpairing\else%
481       \begingroup%
482       \initnumbering@quote%
483       \ifwidthliketwocolumns%
484         \csuse{setwidthliketwocolumns@\columns@position}%
485         \csuse{setpositionliketwocolumns@\columns@position}%
486       \fi%
487     \fi%
488     \set@continuousnumberingforL%
489   \else
490     \led@err@NumberingShouldHaveStarted
491     \endnumbering
492     \beginnumbering
493   \fi}
494
495
496 %

```

`\set@continuousnumberingforL` `\set@continuousnumberingforl` set the left line numbers and pstart counter at a `\beginnumbering` or a `\resumenumbering` in order to have continuous numbering with single column text.

```

497 \newcommand{\set@continuousnumberingforL}{%
498   \ifcontinuousnumberingwithcolumns%
499     \ifdefined\line@numR%
500       \ifnum\line@numR>\line@num%
501         \expandafter\setlinenum\expandafter{\the\line@numR}%
502       \fi%
503       \ifnum\last@page@numR>\last@page@num%

```

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

```

504     \global\last@page@num=\last@page@numR%
505     \fi%
506   \fi%
507   \ifl@dpairing%
508     \unless\ifl@dpaging%
509       \global\c@pstartL=\c@pstart%
510     \fi%
511   \fi%
512 \fi%
513 }%
514 %

```

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 \TeX 3 list, however such migration would take quite time with some risk of error, for a gain which will be minor.

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

```

515 \newcommand*{\list@create}[1]{%
516   \global\let#1=\empty%
517 }%
518 %

```

`\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 .

```

519 \newcommand*{\list@clear}[1]{%
520   \global\let#1=\empty%
521 }
522 %

```

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

```

523 \newtoks\led@toksa \newtoks\led@toksb
524 \global\led@toksa={\}
525 \long\def\xright@appenditem#1\to#2{%
526   \global\led@toksb=\expandafter{#2}%
527   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
528   \global\led@toksb={}}
529 %

```

\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`.

```

530 \long\def\xleft@appenditem#1\to#2{%
531   \global\led@toksb=\expandafter{#2}%
532   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
533   \global\led@toksb={}}
534 %

```

\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.

```

535 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
536 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
537
538 %

```

V Line counting

V.1 Choosing the system of lineation

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

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

\bypstart@true

\bypstart@false

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

\ifbypage@

\bypage@true

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

```

539 \newif\ifbypage@
540 \newif\ifbypstart@
541 %

```

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@R42 \newif\ifbypage@R
\ifbypstart@R43 \newif\ifbypstart@R
544 %

```

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

```

545 \newcommand*{\lineation}[1]{%
546 %

```

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

```

547 \ifnumbering
548 \led@err@LineationInNumbered
549 \else
550 %

```

If the argument is page.

```

551 \def\@tempa{#1}\def\@tempb{page}%
552 \ifx\@tempa\@tempb
553 \global\bypage@true
554 \global\bypstart@false
555 \unless\ifnocritical@%
556 \Xpstart[] [false]%
557 \fi%
558 %

```

If the argument is pstart.

```

559 \else
560 \def\@tempb{pstart}%
561 \ifx\@tempa\@tempb
562 \global\bypage@false
563 \global\bypstart@true
564 \unless\ifnocritical@%
565 \Xpstart%
566 \fi%
567 %

```

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

```

568 \else
569 \def\@tempb{section}
570 \ifx\@tempa\@tempb
571 \global\bypage@false
572 \global\bypstart@false
573 \unless\ifnocritical@%
574 \Xpstart[] [false]%
575 \fi%
576 %

```

In other case, it is an error.

```

577         \else
578             \led@warn@BadLineation
579         \fi
580     \fi
581 \fi
582 \fi}}
583 %

```

V.2 Line number margin

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

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

`\l@edgetline@margin`

```

584 \newcount\line@margin%
585 \newcount\line@margin@columns%Only for parallel typesetting
586 \line@margin@columns=\m@ne%
587
588 \newcommand*{\linenummargin}[1]{%
589     \l@edgetline@margin{#1}%
590     \ifnum\@l@tempcntb>\m@ne
591         \ifledRcol
592             \global\line@marginR=\@l@tempcntb
593             \led@warn@setting@in@rightside{\linenummargin}%
594         \else
595             \global\line@margin=\@l@tempcntb
596         \fi
597     \fi}}
598
599 \newcommand*{\l@edgetline@margin}[1]{%
600     \def\@tempa{#1}\def\@tempb{left}%
601     \ifx\@tempa\@tempb
602         \@l@tempcntb \z@
603     \else
604         \def\@tempb{right}%
605         \ifx\@tempa\@tempb
606             \@l@tempcntb \@ne
607         \else
608             \def\@tempb{outer}%
609             \ifx\@tempa\@tempb
610                 \@l@tempcntb \tw@
611             \else
612                 \def\@tempb{inner}%
613                 \ifx\@tempa\@tempb
614                     \@l@tempcntb \thr@@
615                 \else
616                     \led@warn@BadLinenummargin
617                     \@l@tempcntb \m@ne

```

```

618         \fi
619     \fi
620 \fi
621 \fi}
622
623 %

```

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; `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.

`\c@linenumincrement`

```

624 \newcounter{firstlinenum}
625 \setcounter{firstlinenum}{5}
626 \newcounter{linenumincrement}
627 \setcounter{linenumincrement}{5}
628 %

```

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

`\c@sublinenumincrement`

```

629 \newcounter{firstsublinenum}
630 \setcounter{firstsublinenum}{5}
631 \newcounter{sublinenumincrement}
632 \setcounter{sublinenumincrement}{5}
633
634 %

```

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

`\linenumincrement`

`\firstsublinenum`

`\sublinenumincrement`

```

635 \newcommand*{\firstlinenum}[1]{%
636     \ifledRcol%
637         \setcounter{firstlinenumR}{#1}%
638         \led@warn@setting@in@rightside{\firstlinenum}%
639     \else%
640         \setcounter{firstlinenum}{#1}%
641     \fi%
642 }
643
644 \newcommand*{\linenumincrement}[1]{%
645     \ifledRcol%
646         \setcounter{linenumincrementR}{#1}%
647         \led@warn@setting@in@rightside{\linenumincrement}%
648     \else%
649         \setcounter{linenumincrement}{#1}%
650     \fi%

```

```

651 }
652 \newcommand*{\firstsublinenum}[1]{%
653   \ifledRcol%
654     \setcounter{firstsublinenumR}{#1}%
655     \led@warn@setting@in@rightside{\firstsublinenum}%
656   \else%
657     \setcounter{firstsublinenum}{#1}%
658   \fi%
659 }
660 \newcommand*{\sublinenumincrement}[1]{%
661   \ifledRcol%
662     \setcounter{sublinenumincrementR}{#1}%
663     \led@warn@setting@in@rightside{\sublinenumincrement}%
664   \else%
665     \setcounter{sublinenumincrement}{#1}%
666   \fi%
667 }
668
669 %

```

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.

```

670 \newcount\lock@disp
671 \newcommand{\lockdisp}[1]{%
672   \l@getlock@disp{#1}%
673   \ifnum\l@ldtempcntb>\m@ne
674     \global\lock@disp=\l@ldtempcntb
675   \else
676     \led@warn@BadLockdisp
677   \fi}}
678 \newcommand*{\l@getlock@disp}[1]{
679   \def\@tempa{#1}\def\@tempb{first}%
680   \ifx\@tempa\@tempb
681     \l@ldtempcntb \z@
682   \else
683     \def\@tempb{last}%
684     \ifx\@tempa\@tempb
685       \l@ldtempcntb \@ne
686     \else
687       \def\@tempb{all}%
688       \ifx\@tempa\@tempb
689         \l@ldtempcntb \tw@
690       \else

```



```

691         \@l@dttempcntb \m@ne
692         \fi
693     \fi
694 \fi}
695
696 %

```

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

```

697 \newcount\sublock@disp
698 \newcommand{\sublockdisp}[1]{\{%
699     \l@dgetlock@disp{#1}%
700     \ifnum\@l@dttempcntb>\m@ne
701         \global\sublock@disp=\@l@dttempcntb
702     \else
703         \led@warn@BadSublockdisp
704     \fi}}
705
706 %

```

V.5 Line number style

`\linenumberstyle` We provide a mechanism for using different representations of the line numbers, not
`\linenumrep` just the normal arabic.
`\linenumr@p` NOTE: In v0.7 `\linenumrep` and `\sublinenumrep` replaced the internal `\linenumr@p`
`\sublinenumberstyle` and `\sublinenumr@p`.
`\sublinenumrep` `\linenumberstyle` and `\sublinenumberstyle` are user level macros for setting
`\sublinenumr@p` the number representation (`\linenumrep` and `\sublinenumrep`) for line and sub-line
 numbers.

```

707 \newcommand*{\linenumberstyle}[1]{\%
708     \def\linenumrep##1{\@nameuse{@#1}{##1}}
709 \newcommand*{\sublinenumberstyle}[1]{\%
710     \def\sublinenumrep##1{\@nameuse{@#1}{##1}}
711 %

```

Initialise the number styles to arabic.

```

712 \linenumberstyle{arabic}
713 \let\linenumr@p\linenumrep
714 \sublinenumberstyle{arabic}
715 \let\sublinenumr@p\sublinenumrep
716
717 %

```

V.6 Line number printing

`\leftlinenum` `\leftlinenum` and `\rightlinenum` are the macros that are called to print marginal
`\rightlinenum` line numbers on a page, for left- and right-hand margins respectively. They are made
`\linenumsep`
`\numlabfont`
`\ledlinenum`

easy to access and change, since you may want to change the styling in some way. These standard versions illustrate the general sort of thing that will be needed; they are based on the `\leftheadline` macro in *The TeXbook*, p. 416.

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

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

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

```

718 \newlength{\linenumsep}
719 \setlength{\linenumsep}{1pc}
720 \newcommand*{\numlabfont}{\normalfont\scriptsize}
721 \newcommand*{\ledlinenum}{%
722   \bgroup%
723   \ifluatex%
724     \texdir TLT%
725   \fi%
726   \numlabfont\linenumrep{\line@num}%
727   \ifsublines@
728     \ifnum\subline@num>0\relax
729       \unskip%
730       \Xsublinesep@side%
731       \sublinenumrep{\subline@num}%
732     \fi
733   \fi%
734   \egroup%
735 }%
736
737 \newcommand*{\leftlinenum}{%
738   \ledlinenum
739   \kern\linenumsep}
740 \newcommand*{\rightlinenum}{%
741   \kern\linenumsep
742   \ledlinenum}
743
744 %

```

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`.

```
745 \newcount\line@num
746 %
```

`\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.

```
747 \newcount\subline@num
748 %
```

`\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`

```
749 \newif\ifsublines@
750 %
```

`\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.

```
751 \newcount\absline@num
752 %
```

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.

```
753 \newcount\@lock
754 \newcount\sub@lock
755 %
```

V.9 Line number associated to lemma

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Here are the commands to create these lists:

```

756 \list@create{\line@list}
757 \list@create{\insertlines@list}
758 \list@create{\actionlines@list}
759 \list@create{\actions@list}
760
761 %

```

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

```

\endpage@num
\endline@num
\endsubline@num
762 \newcount\page@num
763 \newcount\endpage@num
764 \newcount\endline@num
765 \newcount\endsubline@num
766 %

```

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

```

767 \newif\ifnoteschanged@
768 %

```

`\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@ 769 \newcommand*{\resetprevline@}{%
770 \def\do##1{\global\csundef{prevline##1}}}%
771 \dolistloop{\@series}%
772 }
773 %

```

`\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.

```

774 \resetprevpage@ \newcommand*{\resetprevpage@num}{%
775     \def\do##1{%
776         \ifcsdef{prevpage##1@num}{%
777             \global\csname prevpage##1@num\endcsname=\z@%
778             \global\csname prevpage##1@numR\endcsname=\z@%
779         }%
780     }%
781     \ifcsdef{##1prevpage@num}{%
782         \global\csname ##1prevpage@num\endcsname=\z@%
783         \global\csname ##1prevpage@numR\endcsname=\z@%
784     }%
785     {}%
786 }%
787 \dolistloop{\@series}%
788 }
789 %

```

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.

```

790 \newread\@inputcheck
791 \newcommand*{\read@linelist}[1]{%
792     \iflabeledRcol%
793     \list@clearing@regR%
794     \else%
795     \list@clearing@reg%
796     \fi%
797 %

```

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).

```

798     \list@clear{\maxlinesinpar@list}
799 %

```

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

ignore carriage returns, since if we are in horizontal mode they can get interpreted as spaces to be printed.

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

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

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

```

800 \get@linelistfile{#1}%
801 \ifcontinuousnumberingwithcolumns
802     \global\page@numR=\page@numR\relax
803     \global\last@page@numR=\last@page@numR\relax
804     \global\page@num=\page@num\relax
805     \global\last@page@num=\last@page@num\relax
806 \fi
807 \@stopmsd%Security if last \endms{} is forgotten
808 \unless\ifledRcol%Get the last line of the last page
809     \csnumgdef{@lastabsline@forpage@the\page@num}{the\absline@num}%
810     \csnumgdef{@lastline@forpage@the\page@num}{the\line@num}%
811 \else%
812     \csnumgdef{@lastabsline@forpageR@the\page@numR}{the\absline@numR}%
813     \csnumgdef{@lastline@forpageR@the\page@numR}{the\line@numR}%
814 \fi%
815 \endgroup
816 %

```

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.

```

817 \ifledRcol
818     \global\page@numR=\m@ne
819     \ifx\actionlines@listR\empty
820         \gdef\next@actionlineR{1000000}%
821     \else
822         \glp\actionlines@listR\to\next@actionlineR
823         \glp\actions@listR\to\next@actionR
824     \fi
825 \else
826     \global\page@num=\m@ne
827     \ifx\actionlines@list\empty
828         \gdef\next@actionline{1000000}%
829     \else
830         \glp\actionlines@list\to\next@actionline
831         \glp\actions@list\to\next@action
832     \fi
833 \fi

```



```
834 }
835 %
```

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

```
836 \newcommand*{\list@clearing@reg}{%
837   \list@clear{\line@list}%
838   \list@clear{\insertlines@list}%
839   \list@clear{\actionlines@list}%
840   \list@clear{\actions@list}%
841   \list@clear{\linesinpar@listL}%
842   \list@clear{\linesonpage@listL}%
843 }%
844 %
```

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

```
845 \newcommand*{\get@linelistfile}[1]{%
846   \InputIfFileExists{\l@auxdir#1}{%
847     \global\noteschanged@false
848     \begingroup
849       \catcode`\[=1 \catcode`\]=2
850       \makeatletter \catcode`\^M=9}{%
851     \led@warn@NoFile{\l@auxdir#1}%
852     \global\noteschanged@true
853     \begingroup}%
854 }
855 %
856 %
```

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. 20 above).

V.11 Commands within the line-list file

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

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

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

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

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

```

857 \newcommand{\line@list@version}[1]{%
858   \IfStrEq{#1}{\this@line@list@version}%
859   {%
860     {\ifledRcol%
861       \led@warn@obsolete{\jobname.\extensionchars\the\section@num}%
862     \else%
863       \led@warn@obsolete{\jobname.\extensionchars\the\section@num}%
864     \fi%
865     \endinput%
866   }%
867 }%
868 %

```

`\@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`.

```

869 \newcommand*{\@nl}[2]{%
870   \fix@page{#1}%
871   \ifledRcol%
872     \@nl@regR%
873   \else%
874     \@nl@reg%
875   \fi%
876 }%

```

```

877 }
878 \newcommand*{\@nl@reg}{%
879   \ifx\l@dchset@num\relax \else
880     \advance\absline@num \@ne
881     \set@line@action
882     \let\l@dchset@num=\relax
883     \advance\absline@num \m@ne
884     \advance\line@num \m@ne
885   \fi
886 %

```

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

```

887   \advance\absline@num \@ne
888     \ifx\next@page@num\relax \else
889       \page@action
890       \let\next@page@num=\relax
891     \fi
892     \ifx\sub@change\relax \else
893       \ifnum\sub@change>\z@
894         \sublines@true
895       \else
896         \sublines@false
897       \fi
898       \sub@action
899       \let\sub@change=\relax
900     \fi
901 %

```

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

```

902   \ifcase\@lock
903     \or
904       \@lock \tw@
905     \or \or
906       \@lock \z@
907   \fi
908   \ifcase\sub@lock
909     \or
910       \sub@lock \tw@
911     \or \or
912       \sub@lock \z@
913   \fi
914 %

```

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

```

915   \ifsublines@
916     \ifnum\sub@lock<\tw@
917       \advance\subline@num \@ne

```

```

918         \fi
919     \else
920         \ifnum \@lock<\tw@
921             \advance\line@num \@ne \subline@num \z@
922         \fi
923     \fi}
924
925 %

```

`\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`.

```

926 \newcount\last@page@num
927 \last@page@num=-10000
928
929 \newcommand*{\fix@page}[1]{%
930     \ifledRcol
931         \ifnum #1=\last@page@numR
932         \else
933             \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
934             \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
935             \ifbypageR
936                 \line@numR \z@ \subline@numR \z@
937             \fi
938             \page@numR=#1\relax
939             \last@page@numR=#1\relax
940             \def\next@page@numR{#1}%
941         \fi
942     \else
943         \ifnum #1=\last@page@num
944         \else
945             \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
946             \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
947             \ifbypage@
948                 \line@num \z@ \subline@num \z@
949             \fi
950             \page@num=#1\relax
951             \last@page@num=#1\relax
952             \def\next@page@num{#1}%
953             \listxadd{\normal@page@break}{\the\absline@num}
954         \fi
955     \fi}
956 %

```

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

```

\@lopL
\@lopR
957 \newcommand*{\@pend}[1]{}
958 \newcommand*{\@pendR}[1]{}

```

```

959 \newcommand*{\@lopL}[1]{}
960 \newcommand*{\@lopR}[1]{}
961
962 %

```

\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.

\sub@off

```

963 \newcommand*{\sub@on}{\ifsublines@
964   \let\sub@change=\relax
965   \else
966     \def\sub@change{1}%
967   \fi}
968 \newcommand*{\sub@off}{\ifsublines@
969   \def\sub@change{-1}%
970   \else
971     \let\sub@change=\relax
972   \fi}
973
974 %

```

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

```

975
976 \newcommand*{\@adv}[1]{%
977   \ifsublines@
978     \ifledRcol
979       \advance\subline@numR by #1\relax
980       \ifnum\subline@numR<\z@
981         \led@warn@BadAdvancelineSubline
982         \subline@numR \z@
983       \fi
984     \else
985       \advance\subline@num by #1\relax
986       \ifnum\subline@num<\z@
987         \led@warn@BadAdvancelineSubline
988         \subline@num \z@
989       \fi
990     \fi
991   \else
992     \ifledRcol
993       \advance\line@numR by #1\relax
994       \ifnum\line@numR<\z@
995         \led@warn@BadAdvancelineLine
996         \line@numR \z@
997       \fi
998     \else
999       \advance\line@num by #1\relax

```

```

1000     \ifnum\line@num<\z@
1001         \led@warn@BadAdvancelineLine
1002         \line@num \z@
1003     \fi
1004 \fi
1005 \fi
1006 \set@line@action}
1007
1008 %

```

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

```

1009
1010 \newcommand*{\@set}[1]{%
1011     \ifledRcol
1012         \ifsublines@
1013             \subline@numR=#1\relax
1014         \else
1015             \line@numR=#1\relax
1016         \fi
1017         \set@line@action
1018     \else
1019         \ifsublines@
1020             \subline@num=#1\relax
1021         \else
1022             \line@num=#1\relax
1023         \fi
1024         \set@line@action
1025     \fi}
1026
1027 %

```

\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.

```

1028
1029 \newcommand*{\l@d@set}[1]{%
1030     \ifledRcol
1031         \line@numR=#1\relax
1032         \advance\line@numR \@ne
1033         \def\l@dchset@num{#1}
1034     \else
1035         \line@num=#1\relax
1036         \advance\line@num \@ne
1037         \def\l@dchset@num{#1}
1038     \fi}
1039 \let\l@dchset@num\relax

```

```
1040
1041 %
```

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

```
1042
1043 \newcommand*{\page@action}{%
1044   \ifledRcol
1045     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1046     \xright@appenditem{\next@page@numR}\to\actions@listR
1047   \else
1048     \xright@appenditem{\the\absline@num}\to\actionlines@list
1049     \xright@appenditem{\next@page@num}\to\actions@list
1050   \fi}
1051 %
```

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

```
1052
1053 \newcommand*{\set@line@action}{%
1054   \ifledRcol
1055     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1056     \ifsublines@
1057       \@l@tempcnta=-\subline@numR
1058     \else
1059       \@l@tempcnta=-\line@numR
1060     \fi
1061     \advance\@l@tempcnta by -5000\relax
1062     \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1063   \else
1064     \xright@appenditem{\the\absline@num}\to\actionlines@list
1065     \ifsublines@
1066       \@l@tempcnta=-\subline@num
1067     \else
1068       \@l@tempcnta=-\line@num
1069     \fi
1070     \advance\@l@tempcnta by -5000\relax
1071     \xright@appenditem{\the\@l@tempcnta}\to\actions@list
1072   \fi}
1073 %
```

\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.

```
1074
1075 \newcommand*{\sub@action}{%
1076   \ifledRcol
1077     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1078     \ifsublines@
```

```

1079 \xright@appenditem{-1001}\to\actions@listR
1080 \else
1081 \xright@appenditem{-1002}\to\actions@listR
1082 \fi
1083 \else
1084 \xright@appenditem{\the\absline@num}\to\actionlines@list
1085 \ifsublines@
1086 \xright@appenditem{-1001}\to\actions@list
1087 \else
1088 \xright@appenditem{-1002}\to\actions@list
1089 \fi
1090 \fi}
1091 %

```

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

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

```

1092 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1093
1094 \newcommand*{\do@lockon}{%
1095 \ifx\next\lock@off
1096 \global\let\lock@off=\skip@lockoff
1097 \else
1098 \ifledRcol
1099 \do@lockonR
1100 \else
1101 \do@lockonL
1102 \fi
1103 \fi}
1104
1105
1106 \newcommand*{\do@lockonL}{%
1107 \xright@appenditem{\the\absline@num}\to\actionlines@list
1108 \ifsublines@
1109 \xright@appenditem{-1005}\to\actions@list
1110 \ifnum\sub@lock=\z@
1111 \sub@lock \@ne
1112 \else
1113 \ifnum\sub@lock=\thr@@
1114 \sub@lock \@ne
1115 \fi
1116 \fi
1117 \else
1118 \xright@appenditem{-1003}\to\actions@list

```



```

1119 \ifnum \@lock=\z@
1120 \@lock \@ne
1121 \else
1122 \ifnum \@lock=\thr@@
1123 \@lock \@ne
1124 \fi
1125 \fi
1126 \fi}
1127
1128 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1129 \newcommand*{\do@lockoffL}{%
1130 \xright@appenditem{\the\absline@num}\to\actionlines@list
1131 \ifsublines@
1132 \xright@appenditem{-1006}\to\actions@list
1133 \ifnum\sub@lock=\tw@
1134 \sub@lock \thr@@
1135 \else
1136 \sub@lock \z@
1137 \fi
1138 \else
1139 \xright@appenditem{-1004}\to\actions@list
1140 \ifnum \@lock=\tw@
1141 \@lock \thr@@
1142 \else
1143 \@lock \z@
1144 \fi
1145 \fi}
1146
1147 \newcommand*{\do@lockoff}{%
1148 \ifledRcol
1149 \do@lockoffR
1150 \else
1151 \do@lockoffL
1152 \fi}
1153 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1154 \global\let\lock@off=\do@lockoff
1155
1156 %

```

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

```

1157 \newcommand*{\n@num}{%
1158 \ifledRcol%
1159 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1160 \xright@appenditem{-1007}\to\actions@listR
1161 \else%

```

```

1162 \xright@appenditem{\the\absline@num}\to\actionlines@list%
1163 \xright@appenditem{-1007}\to\actions@list%
1164 \fi%
1165 }%
1166
1167 %

```

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

```

1168 \newcommand*{\n@num@stanza}{%
1169 \ifledRcol%
1170 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1171 \xright@appenditem{-1008}\to\actions@listR%
1172 \else%
1173 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1174 \xright@appenditem{-1008}\to\actions@list%
1175 \fi%
1176 }
1177 %

```

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

`\hidenumberingonleftpage`

`\hidenumberingonrightpage`

```

1178 \newif\ifl@dhidenumber
1179 \newcommand*{\hidenumbering}{%
1180 \ifledRcol%
1181 \write\linenum@outR{\string\hide@num}%
1182 \else%
1183 \write\linenum@out{\string\hide@num}%
1184 \fi%
1185 }%
1186 \newcommand*{\hide@num}{%
1187 \ifledRcol%
1188 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1189 \xright@appenditem{-1009}\to\actions@listR%
1190 \else%
1191 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1192 \xright@appenditem{-1009}\to\actions@list%
1193 \fi%
1194 }
1195 \newcommand*{\hidenumberingonleftpage}{%
1196 \ifledRcol%
1197 \write\linenum@outR{\string\hide@num@left}%
1198 \else%
1199 \write\linenum@out{\string\hide@num@left}%
1200 \fi%
1201 }%
1202

```

```

1203 \newcommand*{\hide@num@left}{%
1204   \ifledRcol%
1205     \ifodd\page@numR\else%
1206       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1207       \xright@appenditem{-1009}\to\actions@listR%
1208     \fi%
1209   \else%
1210     \ifodd\page@num\else%
1211       \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1212       \xright@appenditem{-1009}\to\actions@list%
1213     \fi%
1214   \fi%
1215 }%
1216
1217 \newcommand*{\hidenumberingonrightpage}{%
1218   \ifledRcol%
1219     \write\linenum@outR{\string\hide@num@right}%
1220   \else%
1221     \write\linenum@out{\string\hide@num@right}%
1222   \fi%
1223 }%
1224
1225 \newcommand*{\hide@num@right}{%
1226   \ifledRcol%
1227     \ifodd\page@numR%
1228       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1229       \xright@appenditem{-1009}\to\actions@listR%
1230     \fi%
1231   \else%
1232     \ifodd\page@num%
1233       \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1234       \xright@appenditem{-1009}\to\actions@list%
1235     \fi%
1236   \fi%
1237 }%
1238
1239 %

```

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

\insert@count

- #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.

```

1240 \newcount\insert@count
1241 %

```

- #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.

```
1242 \newcommand*\dummy@ref}[2]{#2}
1243 %
```

\@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.

```
1244 \newcommand*\@ref}[2]{%
1245   \ifledRcol%
1246     \@ref@regR{#1}{#2}%
1247   \else%
1248     \@ref@reg{#1}{#2}%
1249   \fi%
1250 }%
1251 \newcommand*\@ref@reg}[2]{%
1252   \global\insert@count=#1\relax
1253   \global\advance\@edtext@level by 1%
1254   \loop\ifnum\insert@count>\z@
1255     \xright@appenditem{\the\absline@num}\to\insertlines@list
1256     \global\advance\insert@count \m@ne
1257   \repeat
1258 %
```

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.

```
1259 \begingroup
1260   \let\@ref=\dummy@ref
1261   \let\@lopL\@gobble
1262   \let\page@action=\relax
1263   \let\sub@action=\relax
1264   \let\set@line@action=\relax
1265   \let\@lab=\relax
1266   \let\@lemma=\relax%
1267   \let\@sw\@gobblethree%
1268   #2
1269   \global\endpage@num=\page@num
1270   \global\endline@num=\line@num
1271   \global\endsubline@num=\subline@num
1272 \endgroup
1273 %
```

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

```

1274 \xright@appenditem%
1275   {\the\page@num|\the\line@num|
1276    \ifsublines@ \the\subline@num \else 0\fi}%
1277   \the\endpage@num|\the\endline@num|
1278   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1279 %

```

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

```

1280 \@ref@reg@parse{#2}%
1281 %

```

Decrease edtext level counter.

```

1282 \global\advance\@edtext@level by -1%
1283 }
1284 %

```

`\@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.

```

1285 \newcommand{\@ref@reg@parse}[1]{%
1286   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
1287   @edtext@level\endcsname}%
1288   \providebool{lemmacommand@\the\@edtext@level}%
1289   \boolfalse{lemmacommand@\the\@edtext@level}%
1290 }
1291 %

```

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

```

1290 #1%
1291 %

```

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.

```

1292 \ifnum\@edtext@level>0%
1293   \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
1294   csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1295   \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level
1296   }{}%
1297   \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
1298   \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}%
1299   \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1300   \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
1301   \fi%
1302 }
1303 %

```

`\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.

```

1304 \newcommand{\@ref@later}[1]{%
1305   \global\advance\@edtext@level by \@ne%
1306   \ifledRcol%
1307     \@ref@reg@parseR{#1}%
1308   \else%
1309     \@ref@reg@parse{#1}%
1310   \fi%
1311   \global\advance\@edtext@level by -\@ne%
1312 }%
1313 %

```

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`.

```

1314 \newwrite\linenum@out
1315 %

```

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

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

```

1316 \newif\iffirst@linenum@out
1317 \iffirst@linenum@out@true
1318 %

```

`\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.

```

1319 \newcommand{\this@line@list@version}{7}%
1320 %

```

`\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.

```
1321 \newcommand*{\line@list@stuff}[1]{%
1322 %
```

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

```
1323 \global\newtoggle{notfirstrun@#1}%
1324 \IfFileExists{\l@auxdir#1}%
1325 {\global\toggletrue{notfirstrun@#1}}%
1326 {\global\togglefalse{notfirstrun@#1}}%
1327 %
```

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

```
1328 \read@linelist{#1}%
1329 %
```

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.

```
1330 \iffirst@linenum@out@
1331 \immediate\closeout\linenum@out%
1332 \global\first@linenum@out@false%
1333 \immediate\openout\linenum@out=\l@auxdir#1\relax%
1334 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}%
1335 \ifl@dpaging%
1336 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}%
1337 \fi%
1338 \else
1339 %
```

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.

```
1340 \if@minipage%
1341 \leavevmode%
1342 \fi%
1343 \closeout\linenum@out%
1344 \openout\linenum@out=\l@auxdir#1\relax%
1345 \write\linenum@out{\string\line@list@version{\this@line@list@version}}
%
1346 \ifl@dpaging%
1347 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}
%
1348 \fi%
1349 \fi}
1350 %
1351 %
```

\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.

```

1352 \newcommand*{\new@line}{%
1353   \IfStrEq{\led@pb@setting}{after}%
1354     {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1355       {\xifinlist{\the\absline@num}{\normal@page@break}%
1356         {\numdef{\@next@page}{\c@page+\@ne}%
1357           \write\linenum@out{\string\@nl[\@next@page][\@next@page]}}%
1358         }%
1359       {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1360     }%
1361     {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}
1362 {}%
1363 \IfStrEq{\led@pb@setting}{before}%
1364   {\numdef{\next@absline}{\the\absline@num+\@ne}%
1365     \xifinlist{\next@absline}{\l@prev@nopb}%
1366       {\xifinlist{\the\absline@num}{\normal@page@break}%
1367         {\numdef{\nc@page}{\c@page+\@ne}%
1368           \write\linenum@out{\string\@nl[\nc@page][\nc@page]}}%
1369         }%
1370       {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}
1371     }%
1372     {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}}
1373   }%
1374   {}%
1375   \IfStrEqCase{\led@pb@setting}{\{before\}{\relax}\{after\}{\relax}}{\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1376 }
1377
1378 %

```

\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

```

1379 \newif\if@noneed@Footnote%
1380
1381 \newcommand*{\flag@start}{%
1382   \ifledRcol%
1383     \edef\next{\write\linenum@out{%
1384       \string\@ref[\the\insert@countR]}}%
1385     \next%
1386     \ifnum\insert@countR<1%
1387       \if@noneed@Footnote\else%
1388         \led@err@EdtextWithoutFootnote%
1389       \fi%

```



```

1390 \fi%
1391 \else%
1392 \edef\next{\write\linenum@out{%
1393 \string\@ref[\the\insert@count] []}%
1394 \next%
1395 \ifnum\insert@count<1%
1396 \if@noneed@Footnote\else%
1397 \led@err@EdtextWithoutFootnote%
1398 \fi%
1399 \fi%
1400 \fi}%
1401
1402 \newcommand*{\flag@end}{%
1403 \ifledRcol%
1404 \write\linenum@outR[]}%
1405 \else%
1406 \write\linenum@out[]}%
1407 \fi}%
1408
1409
1410 %

```

`\flag@start@RTL` With \LaTeX , there is a problem when using RTL: the writing of a command in the numbered auxiliary files (.1, .2 etc) is reversed when the first argument of `\edtext` is 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.

```

1411 \newcommand{\flag@start@RTL}{%
1412 \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1413 \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%

```

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

```

1414 {000}%
1415 {}%
1416 {%
1417 \ifnumequal%
1418 {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1419 {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1420 {\flag@end}%
1421 {\flag@start}%
1422 }%
1423 }%
1424
1425 \newcommand{\flag@end@RTL}{%
1426 \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}%
1427 \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1428 {000}%
1429 {}%
1430 {%
1431 \ifnumequal%
1432 {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1433 {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1434 {\flag@start}%
1435 {\flag@end}%
1436 }%
1437 }%
1438 %

```

`\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

```

1439 \newcommand*{\flag@start@later}{%
1440 \ifledRcol%
1441 \write\linenum@outR{\string\@ref@later}%
1442 \else%
1443 \write\linenum@out{\string\@ref@later}%
1444 \fi%
1445 }%
1446 \newcommand{\flag@end@later}{%
1447 \ifledRcol%
1448 \write\linenum@outR{}}%
1449 \else%
1450 \write\linenum@out{}}%
1451 \fi%
1452 }
1453 %

```

`\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.

```

1454
1455
1456 \newcommand*{\startsub}{\dimen0\lastskip
1457 \ifdim\dimen0>0pt \unskip \fi
1458 \ifledRcol \write\linenum@outR{\string\sub@on}%
1459 \else \write\linenum@out{\string\sub@on}%
1460 \fi
1461 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1462 \def\endsub{\dimen0\lastskip
1463 \ifdim\dimen0>0pt \unskip \fi
1464 \ifledRcol \write\linenum@outR{\string\sub@off}%
1465 \else \write\linenum@out{\string\sub@off}%
1466 \fi
1467 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1468
1469 %

```

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

```

1470 \newcommand*{\advanceline}[1]{\leavevmode%
1471 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1472 \else \write\linenum@out{\string\@adv[#1]}%
1473 \fi%
1474 }
1475 %

```

`\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.

```

1476
1477 \newcommand*{\setline}[1]{%
1478 \leavevmode%
1479 \ifnum#1<\z@
1480 \led@warn@BadSetline
1481 \else
1482 \ifledRcol \write\linenum@outR{\string\@set[#1]}%
1483 \else \write\linenum@out{\string\@set[#1]}%
1484 \fi
1485 \fi}

```

```
1486
1487 %
```

\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.

```
1488
1489 \newcommand*{\setlinenum}[1]{%
1490   \ifnum#1<\z@
1491     \led@warn@BadSetlinenum
1492   \else
1493     \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1494     \else      \write\linenum@out{\string\l@d@set[#1]} \fi
1495   \fi}
1496
1497 %
```

\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.

```
1498
1499 \newcommand*{\startlock}{%
1500   \ifledRcol \write\linenum@outR{\string\lock@on}%
1501   \else      \write\linenum@out{\string\lock@on}%
1502   \fi}
1503 \def\endlock{%
1504   \ifledRcol \write\linenum@outR{\string\lock@off}%
1505   \else      \write\linenum@out{\string\lock@off}%
1506   \fi}
1507 %
```

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

\ifl@dskipversenumber

\l@dskipnumbertrue

\l@dskipnumberfalse

\skipnumbering

```
1508 \newif\ifl@dskipnumber
1509 \newif\ifl@dskipversenumber%
1510 \newcommand*{\skipnumbering}{%
1511   \leavevmode%
1512   \ifledRcol%
1513     \ifistanza%
1514       \write\linenum@outR{\string\n@num@stanza}%
1515     \else%
1516       \write\linenum@outR{\string\n@num}%
1517     \fi%
1518     \advanceline{-1}%
1519   \else%
1520     \ifistanza%
1521       \write\linenum@out{\string\n@num@stanza}%
1522     \else%
```

```

1523 \write\linenum@out{\string\n@num}%
1524 \fi%
1525 \advance\line{-1}%
1526 \fi%
1527 }%
1528
1529 %

```

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

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

VI.1 `\edtext` itself

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

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

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

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

```
1530 \list@create{\end@lemmas}
1531 %
```

`\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.

```
1532 \newcommand{\dummy@edtext}[2]{#1}
1533 %
```

`\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`.

```
1534 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1535 %
```

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

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

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 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 \LaTeX .)

```
1536 \newcommand*{\no@expands}{%
1537   \let\select@lemmafont=0%
```

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

```

1538 \let\startsub=\relax \let\endsub=\relax
1539 \let\startlock=\relax \let\endlock=\relax
1540 \let\edlabel=\@gobble
1541 \let\setline=\@gobble \let\advanceline=\@gobble
1542 \let\sameword\sameword@inedtext%
1543 \let\edtext=\dummy@edtext
1544 \let\edindex\dummy@edindex%
1545 \l@dtabnoexpands
1546 \morenoexpands}
1547 \let\morenoexpands=\relax
1548
1549 %

```

\@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.

```

1550 \newcommand{\@tag}{}
1551 %

```

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

```

1552 \newcount\@edtext@level%
1553 \@edtext@level=0%
1554 %

```

\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.

```

1555 \newif\if@edtext@secondarg%
1556 %

```

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

```

1557 \newcounter{edtext}
1558 \renewcommand{\theedtext}{edtxt\arabic{edtext}}%
1559 %

```

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

```

1560 \newcommand{\edtext}[2]{\leavevmode%
1561 %

```

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

```

1562 \ifnumberedpar%
1563 %

```

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


```

1564 \global\advance\@edtext@level by 1%
1565 %

```

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`.

```

1566 \stepcounter{edtext}%
1567 \csxdef{thisedtext@the\@edtext@level}{\theedtext}%
1568 %

```

By default, we do not use `\lemma`

```

1569 \global\@lemmacommand@false%
1570 %

```

```

1571 \begingroup%
1572 %

```

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`.

```

1573 \ifledRcol%
1574 \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1575 {\global\let\sw@inthisedtext\empty}%
1576 {\expandafter\glop\csname sw@list@edtextR@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1577 \else%
1578 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1579 {\global\let\sw@inthisedtext\empty}%
1580 {\expandafter\glop\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1581 \fi%
1582 %

```

`\@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.

```

1583 \global\renewcommand{\@tag}{%
1584 \noexpands #1%
1585 }%
1586 %

```

`\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`.

```

1587 \set@line%
1588 %

```

`\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`.

```

1589 \ifledRcol \global\insert@countR \z@%
1590 \else      \global\insert@count \z@ \fi%
1591 %

```

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.

```

1592 \@edtext@secondarg@true%
1593 \ignorespaces #2\relax%
1594 \@edtext@secondarg@false%
1595 %

```

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

```

1596 \if@RTL%
1597 \flag@start@RTL%
1598 \else%
1599 \flag@start%
1600 \fi%
1601 %

```

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

```

1602 \if@lemmacommand@%
1603 \ifledRcol%
1604 \write\linenum@outR{\string\@lemma}%
1605 \else%
1606 \write\linenum@out{\string\@lemma}%
1607 \fi%
1608 \fi%
1609 %

```

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.

```

1610 \endgroup%
1611 \ifdef{\hypertarget}%
1612 {%
1613 \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\
@edtext@level}:start}{}}%
1614 \showlemma{#1}%
1615 \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:
end}{}}%
1616 }%
1617 {%
1618 \showlemma{#1}%
1619 }%
1620 %

```

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.

```

1621 \ifx\end@lemmas\empty \else%
1622 \glp\end@lemmas\to\x@lemma%
1623 \x@lemma%
1624 \global\let\x@lemma=\relax%
1625 \fi%
1626 \if@RTL%
1627 \flag@end@RTL%
1628 \else%
1629 \flag@end%
1630 \fi%
1631 %

```

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`.

```

1632 \global\@noneed@Footnotefalse%
1633 \global\advance\@edtext@level by -1%
1634 \global\@lemmacommand@false%
1635 %

```

We also reset `\@beforeinsertofthisedtext`

```

1636 \global\let\@beforeinsertofthisedtext\relax%
1637 %

```

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

```

1638 \else%
1639 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidestart%
1640 \fi%
1641 }%
1642
1643
1644 %

```

\@beforeinsertofthisedtext The `\@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`.

```

1645 \let\@beforeinsertofthisedtext\relax
1646 %

```

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

```

1647 \newif\ifnumberline
1648 \numberlinetrue
1649 %

```

\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.

```

1650 \newcommand*\set@line{%
1651 \ifledRcol
1652 \ifx\line@listR\empty
1653 \global\noteschanged@true
1654 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1655 \else
1656 \glp\line@listR\to\@tempb
1657 \xdef\l@d@nums{\@tempb|\edfont@info}%
1658 \global\let\@tempb=\undefined
1659 \fi
1660 \else
1661 \ifx\line@list\empty
1662 \global\noteschanged@true
1663 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1664 \else
1665 \glp\line@list\to\@tempb
1666 \xdef\l@d@nums{\@tempb|\edfont@info}%

```

```

1667 \global\let\@tempb=\undefined
1668 \fi
1669 \fi}
1670
1671 %

```

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

```

1672 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1673
1674 %

```

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. 129).

```

1675 \newcommand*{\lemma}[1]{%
1676 \global\@lemmacommand@true%
1677 \global\renewcommand{\@tag}{%
1678 \no@expands #1%
1679 }%
1680 \ignorespaces%
1681 }%
1682 %

```

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

```

1683 \newcommand{\@lemma}{%
1684 \booltrue{lemmacommand@the\@edtext@level}%
1685 }%
1686 %

```

\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.

```

1687 \newif\if@lemmacommand@%
1688 %

```

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. 100): 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.

```
1689 \newcommand*{\linenum}[1]{%
1690   \xdef\@tempa{#1|}|}|}|}|noexpand\\l@d@nums}%
1691   \global\let\l@d@nums=\empty
1692   \expandafter\line@set\@tempa|\\ignorespaces}
1693 %
```

\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.

```
1694 \def\line@set#1|#2\\#3|#4\\{%
1695   \gdef\@tempb{#1}%
1696   \ifx\@tempb\empty
1697     \l@d@add{#3}%
1698   \else
1699     \l@d@add{#1}%
1700   \fi
1701   \gdef\@tempb{#4}%
1702   \ifx\@tempb\empty\else
1703     \l@d@add{|}\line@set#2\\#4\\%
1704   \fi}
1705 %
```

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

```
1706 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1707
1708 %
```

VI.4 Lemma disambiguation

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

The general mechanism is the following:

- **At the first run**, each `\sameword` command increments an `etoolbox` counter the name of which contains the argument of the `\sameword` commands.

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

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

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

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

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

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

$\bar{a}^1 \left[\boxed{\text{foo}^1 \text{foo}^2 \text{bar}^2 \text{foo}^3} \right] (\text{A})(\text{B}) \text{foo}^4 \text{bar}^3 \quad (\text{C}) \quad \left[\boxed{\text{foo}^5} \right] (\text{D}) \text{bar}^4 \quad (\text{E})$

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

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

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

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

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

After theory, implementation.

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

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

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 `\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.

```
1709 \newcommand{\get@sw@txt}[1]{%
1710   \begingroup%
1711   %.
```

Allow some macros inside .

```
1712   \let\sameword\@firstofone%Allow to have nested \sameword
1713   \let\emph\@firstofone%
1714   \let\textit\@firstofone%
1715   \let\textbf\@firstofone%
1716   \let\textsc\@firstofone%
1717   \let\framebox\@firstofone%
1718   %
```

Using case sensibility option.

```
1719   \ifsw@caseinsensitive%
1720     \def\@tmpa##1{\lowercase{##1}}%
1721   \else%
1722     \def\@tmpa##1{##1}%
1723   \fi%
1724   %
```

And now, define `\sw@txt`.

```
1725   \ifxetex%
1726     \@tmpa{\xdef\sw@txt{##1}}%
1727   \else%
1728     \@tmpa{\expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize
1729     {##1}}}%
1729   \fi%
1730   \endgroup%
1731 }%
1732 %
```

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

```
1733 \newcommandx{\sameword}[2][1,usedefault]{%
1734   \leavevmode%
1735   \get@sw@txt{##2}%
1736   %
```

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

```
1737   \unless\ifledRcol%
1738     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1739   %
```

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

Then, write its value to the numbered file.

```
1740 \protected@write\linenum@out{}\string\@sw{\sw@txt}{\csuse{sw@\sw@txt
1741 }}{#1}}%
%
```

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

```
1742 \else%
1743 \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1744 \protected@write\linenum@outR{}\string\@sw{\sw@txt}{\csuse{sw@\sw@txt
1745 }}{#1}}%
1746 \fi%
%
```

And print the word.

```
1747 #2%
1748 }%
1749 %
```

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

```
\if@addsw%
1751 %
```

\@sw The command printed in the auxiliary files.

```
1752 \newcommand{\@sw}[3]{%
1753 \get@sw@txt{#1}%
1754 \unless\ifledRcol%
1755 %
```

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

```
1756 \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
1757 %
```

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

```
1758 \numdef{\prev@line}{\the\absline@num-1}%
1759 \ifcsundef{sw@\sw@txt @\prev@line @\the\section@num}{%
1760 \csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
1761 }{}%
1762 %
```

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

```
1763 \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1764 %
```

And do the same thing for the right side.

```

1765 \else%
1766 \csxdef{sw@sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1767 \numdef{\prev@line}{\the\absline@numR-1}%
1768 \ifcsundef{sw@sw@txt @\prev@line @\the\section@numR @R}{%
1769 \csnumgdef{sw@sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1770 }{}%
1771 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R
}}%
1772 \fi%
1773 %

```

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

```

1774 \@tempcnta=\@edtext@level
1775 \@whilenum{\@tempcnta>0}\do{%
1776 \lcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1777 {%
1778 \@addswfalse%
1779 \notbool{lemmacommand@\the\@tempcnta}%
1780 {\@addswtrue}%
1781 {\IfStrEq{#3}{inlemma}%
1782 {\@addswtrue}%
1783 {%
1784 \def\do##1{%
1785 \ifnumequal{##1}{\the\@tempcnta}%
1786 {\@addswtrue\listbreak}%
1787 }%
1788 }%
1789 \docsvlist{#3}%
1790 }%
1791 }%
1792 \if@addsw%
1793 \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1794 \ifledRcol%
1795 \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
1796 \else%
1797 \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
1798 \fi%
1799 \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
1800 \fi%
1801 }%
1802 {}%
1803 \advance\@tempcnta by -1%
1804 }%
1805 }%
1806 %

```

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

```

1807 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1808 \get@sw@txt{#2}%

```

```

1809 \unless\ifledRcol@%
1810 %

```

Just a precaution.

```

1811 \ifx\sw@list@inedtext\empty%
1812 \def\the@sw{999}%
1813 \def\this@absline{-99}%
1814 \else%
1815 %

```

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

```

1816 \glp\sw@list@inedtext\to\@tmp%
1817 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1818 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1819 \fi%
1820 %

```

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

```

1821 \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
1822 \numdef{\prev@line}{\this@absline-1}%
1823 \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1824 }%
1825 {\numdef{\sw@atthisline}{0}}%
1826 %

```

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

```

1827 \ifnumgreater{\sw@atthisline}{1}%
1828 {\showwordrank{#2}{\the@sw}}%
1829 {#2}%
1830 %

```

And the same for right side.

```

1831 \else%
1832 \ifx\sw@list@inedtext\empty%
1833 \def\the@sw{999}%
1834 \def\this@absline{-99}%
1835 \else%
1836 \glp\sw@list@inedtext\to\@tmp%
1837 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1838 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1839 \fi%
1840 \ifcsdef{sw@\sw@txt @\this@absline @\the\section@numR @R}{%
1841 \numdef{\prev@line}{\this@absline-1}%
1842 \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
1843 }%

```

```

1844     {\numdef{\sw@atthisline}{0}}%
1845     \ifnumgreater{\sw@atthisline}{1}%
1846         {\showwordrank{#2}{\the@sw}}%
1847         {#2}%
1848     \fi%
1849 }%
1850 %

```

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

```

1851 \newcommand{\showwordrank}[2]{%
1852     #1\textsuperscript{#2}%
1853 }%
1854 %

```

VII Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than straight onto the vertical list, and when the paragraph ends we slice the paragraph into its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

VII.1 Boxes, counters, `\pstart` and `\pend`

`\raw@text` Here are numbers and flags that are used internally in the course of the paragraph decomposition.

`\ifnumberedpar@` When we first form the paragraph, it goes into a box register, `\raw@text`, instead of onto the current vertical list. The `\ifnumberedpar@` flag will be `true` while a paragraph is being processed in that way. `\num@lines` will store the number of lines in the paragraph when it is complete. When we chop it up into lines, each line in turn goes into the `\one@line` register, and `\par@line` will be the number of that line within the paragraph.

```

1855 \newbox\raw@text
1856 \newif\ifnumberedpar@
1857 \newcount\num@lines
1858 \newbox\one@line
1859 \newcount\par@line
1860 %

```

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

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

`\AtStartEveryPstart`

`\numberpstarttrue`

`\numberpstartfalse`

`\labelpstarttrue`

`\labelpstartfalse`

`\thepstart`

`\ifat@every@pstart@star@`

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

```

1861 \newcommand{\AtStartEveryPstart}[1]{%
1862   \ifstrempy{#1}%
1863     {\gdef\@at@start@every@pstart{}}%
1864     {\gdef\@at@start@every@pstart{#1}}%
1865   }%
1866 \def\@at@start@every@pstart{}%
1867
1868 \newif\ifat@every@pstart@star%
1869 \newcommand{\AtEveryPstart}[1]{%
1870   \ifstrempy{#1}%
1871     {\gdef\@at@every@pstart{}}%
1872     {\gdef\@at@every@pstart{\noindent#1}}%
1873   \global\@at@every@pstart@star@false%
1874 }%
1875 \WithSuffix\newcommand\AtEveryPstart*[1]{%
1876   \ifstrempy{#1}%
1877     {\gdef\@at@every@pstart{}}%
1878     {\gdef\@at@every@pstart{#1}}%
1879   \global\@at@every@pstart@star@true%
1880 }%
1881 \def\@at@every@pstart{}%
1882
1883 \newcounter{pstart}
1884 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
1885 \newif\ifnumberpstart
1886 \numberpstartfalse
1887 \newif\iflabelpstart
1888 \labelpstartfalse
1889 \newcommandx*\pstart[2][1,2,usedefault]{%
1890   \normal@pars%
1891   \ifboolexpr{%
1892     test {\ifstrempy{#1}}%
1893     and test {\ifstrempy{#2}}%
1894   }%
1895   {\@at@every@pstart}%
1896   {%
1897     \ifstrempy{#1}{\noindent#1}%
1898     \ifstrempy{#2}{#2}%
1899   }%
1900 }%
1901 \ifautopar%
1902   \autopar%
1903 \fi%
1904 \ifluatex%
1905   \edef\l@luatexttextdir@L{\the\textdir}%
1906 \fi%
```

```

1907 \@nobraektrue%
1908 \ifnumbering \else%
1909   \led@err@PstartNotNumbered%
1910   \beginnumbering%
1911 \fi%
1912 \ifnumberedpar@%
1913   \led@err@PstartInPstart%
1914   \pend%
1915 \fi%
1916 \list@clear{\inserts@list}%
1917 \global\let\next@insert=\empty%
1918 \begingroup\normal@pars%
1919 \global\advance \l@dnumpstartsL\@ne
1920 \global\setbox\raw@text=\vbox\bgroup%
1921   \if@nobraek%
1922     \if@afterindent\else%
1923       \noindent%
1924       \global\@afterindenttrue%
1925     \fi%
1926   \fi%
1927   \ifautopar\else%
1928   \ifnumberpstart%
1929     \ifinstanza\else%
1930     \ifsidepstartnum\else%
1931       \thepstart%
1932     \fi%
1933   \fi%
1934   \fi%
1935   \fi%
1936 \numberedpar@true%
1937 \iflabelpstart\protected@edef\@currentlabel%
1938   {\p@pstart\thepstart}
1939 \fi%
1940 \l@dzeropenalties%
1941 \@at@start@every@pstart%
1942 \ignorespaces%because not automatically ignored if an optional argument
is used (classical TeX behavior for space after commands)
1943 }
1944 %

```

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

```

1945 \newcommand*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
1946   \led@err@PendNotNumbered%
1947   \fi%
1948   \global\l@dskipversenumberfalse%
1949   \ifnumberedpar@ \else%
1950     \led@err@PendNoPstart%
1951   \fi%
1952 %

```

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.

```

1953 \l@dzeropenalties%
1954 \@at@end@every@pend%
1955 \endgraf\global\num@lines=\prevgraf\egroup%
1956 \global\par@line=0%
1957 %

```

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`.

```

1958 \csnumdef{pstartline}{0}%
1959 \loop\ifvbox\raw@text%
1960   \csnumdef{pstartline}{\pstartline+\@ne}%
1961   \do@line%
1962   \ifbypstart@%
1963     \ifnumequal{\pstartline}{1}{%
1964       \bgroup%
1965       \let\leavevmode\relax%
1966       \setline{1}%
1967       \egroup%
1968       \resetprevline@}{}%
1969     \fi%
1970   \repeat%
1971 %

```

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

```

1972 \flush@notes%
1973 \endgroup%
1974 \ignorespaces%
1975 %

```

Increase `pstart` counter.

```

1976 \ifnumberpstart%
1977   \global\pstartnumtrue%
1978   \fi%
1979   \addtocounter{pstart}{1}%
1980   \ifcontinuousnumberingwithcolumns%
1981     \addtocounter{pstartL}{1}%
1982     \addtocounter{pstartR}{1}%
1983   \fi%
1984 %

```

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


```

1985 \normal@pars%
1986 \ifbool{expr}%
1987   test {\ifstrempy{#1}}%
1988   and test {\ifstrempy{#2}}%
1989 }%
1990 {\at@every@pend}%
1991 {%
1992 \ifstrempy{#1}{\noindent#1}%
1993 \ifstrempy{#2}{\#2}%
1994 }%
1995 %

```

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.

```

1996 \@nobreakfalse%
1997 \ifautopar%
1998   \autopar%
1999 \fi%
2000 }
2001 %

```

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.

```

\AtEveryPend
\at@every@pend
\ifat@every@pend@star%
2002 \newif\ifat@every@pend@star%
2003 \newcommand{\AtEveryPend}[1]{%
2004   \ifstrempy{#1}%
2005     {\gdef\at@every@pend{}}%
2006     {\gdef\at@every@pend{\noindent#1}}%
2007   \global\at@every@pend@star@false%
2008 }%
2009 \WithSuffix\newcommand\AtEveryPend*[1]{%
2010   \ifstrempy{#1}%
2011     {\gdef\at@every@pend{}}%
2012     {\gdef\at@every@pend{#1}}%
2013   \global\at@every@pend@star@true%
2014 }%
2015 \xdef\at@every@pend{}%
2016 %
2017 %
2018 %
2019 %

```

\AtEndEveryPend Here a macro to insert automatically any content at the end of \pend, in numbered lines.

```

2020 \newcommand{\AtEndEveryPend}[1]{%
2021   \ifstrempy{#1}%
2022     {\xdef\at@end@every@pend{}}%

```

```

2023 {\gdef\@at@end@every@pend{#1}}%
2024 }%
2025 \def\@at@end@every@pend{}%
2026 %

```

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

```

2027 \newcommand*\l@dzeroopenalties{%
2028   \brokenpenalty \z@ \clubpenalty \z@
2029   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
2030   \postdisplaypenalty \z@ \widowpenalty \z@}
2031
2032 %

```

\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.

```

2033 \newif\ifautopar
2034 \autoparfalse
2035 \newcommand*\autopar{%
2036   \ifledRcol
2037     \ifnumberingR \else
2038       \led@err@AutoparNotNumbered
2039     \beginnumberingR
2040     \fi
2041   \else
2042     \ifnumbering \else

```

```

2043 \led@err@AutoparNotNumbered
2044 \beginnumbering
2045 \fi
2046 \fi
2047 \autopartrue
2048 \everypar{\setbox0=\lastbox
2049 \endgraf \vskip-\parskip
2050 \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\
fi\fi
2051 \let\par=\pend}%
2052 \ignorespaces}
2053 %

```

\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.

```

2054 \newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}
2055
2056 %

```

\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.

```

2057 \newif\ifautopar@pause
2058 %

```

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

```

2059 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
2060 \newcommand*{\do@line}{%
2061 {\vbadness=10000
2062 \splittopskip=\z@
2063 \do@linehook
2064 \l@demptyd@ta
2065 \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2066 \unvbox\one@line \global\setbox\one@line=\lastbox
2067 \getline@num
2068 \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{\}
2069 \ifnum\@lock>\@ne
2070 \inserthangingsymboltrue
2071 \else
2072 \inserthangingsymbolfalse
2073 \fi

```

```

2074 \check@pb@in@verse
2075 \ifl@dhidenumber%
2076   \global\l@dhidenumberfalse%
2077   \f@x@l@cks%
2078 \else%
2079   \affixline@num%
2080 \fi%
2081 %

```

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

```

2082 \xifinlist{\the\l@dnumstartsL}{\eled@sections@}%
2083   {\print@eledsection}%
2084   {\print@line}%
2085 \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{%}
2086 }%
2087 %

```

VII.2.2 Process for “normal” line

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

```

2088 \def\print@line{
2089 %

```

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

```

2090   \affixpstart@num%
2091 %

```

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

```

2092   \hb@xt@ \linewidth{%
2093 %

```

User hook.

```

2094   \do@insidelinehook%
2095 %

```

Left line number

```

2096   \l@dld@ta%
2097 %

```

Prepare text to be inserted before notes.

```

2098   \if@firstlineofpage%
2099     \set@Xtxtbeforenotes%
2100     \set@txtbeforenotesX%
2101     \global\@firstlineofpagefalse%
2102   \fi%
2103 %

```

Insert footnotes made of manuscripts data and critical footnotes.

```

2104     \ifdefstring{\ms@data@position}{msdata-regular}{%
2105         \insert@msdata%
2106         \add@inserts%
2107         \add@Xgroupbyline%
2108     }{%
2109         \add@inserts%
2110         \add@Xgroupbyline%
2111         \insert@msdata%
2112     }%
2113 %

```

Insert marginal notes.

```

2114     \affixside@note%
2115 %

```

Print left notes.

```

2116     \l@dlsn@te
2117 %

```

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

```

2118     {\ledllfill\hb@xt@ \wd\one@line{\new@line%
2119 %

```

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

```

2120     \ifluatex%
2121         \textdir\l@luatextextdir@L%
2122     \fi%
2123 %

```

Insert, if needed, the hanging symbol.

```

2124     \inserthangingsymbol%
2125 %

```

And so, print the line.

```

2126     \l@dunhbox@line{\one@line}}%
2127 %

```

Right line number

```

2128     \ledrlfill\l@drd@ta%
2129 %

```

Print right notes.

```

2130     \l@drsn@te
2131     }}%
2132 %

```

And reinsert penalties (for page breaking)...

```

2133 \add@penalties%
2134 }
2135 %

```

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.

```

2136 \def\print@eledsection{%
2137   \if@firstlineofpage%
2138     \set@Xtxtbeforenotes%
2139     \set@txtbeforenotesX%
2140     \global\@firstlineofpagefalse%
2141   \fi%
2142   \ifdefstring{\ms@data@position}{msdata-regular}{%
2143     \insert@msdata%
2144     \add@inserts%
2145     \add@Xgroupbyline%
2146   }{%
2147     \add@inserts%
2148     \add@Xgroupbyline%
2149     \insert@msdata%
2150   }%
2151   \affixside@note%
2152   \numdef{\temp@}{\l@dnumpstartsL-1}%
2153   \xifinlist{\temp@}{\eled@sections@@}{\@nbreaktrue}{\@nbreakfalse}%
2154   \@eled@sectioningtrue%
2155   \csuse{eled@sectioning@the\l@dnumpstartsL}%
2156   \@eled@sectioningfalse%
2157   \global\csundef{eled@sectioning@the\l@dnumpstartsL}%
2158   \if@RTL%
2159     \hspace{-3\paperwidth}%
2160     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2161   \else%
2162     \hspace{3\paperwidth}%
2163     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2164   \fi%
2165   \vskip-\baselineskip%
2166 }
2167 %

```

VII.2.4 Hooks

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

\do@insidelinehook

```

2168 \newcommand*{\do@linehook}{%
2169 \newcommand*{\do@insidelinehook}{%
2170 %

```

\do@linehook These high level commands just redefine the low level commands. They have to be used
\do@insidelinehook be user, without `\makeatletter`.

```

2171 \newcommand*{\do@linehook}[1]{\gdef\do@linehook{#1}}%
2172 \newcommand*{\do@insidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
2173
2174 %

```

VII.2.5 Sidenotes and marginal line number initialization

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

```

\l@dcsnotetext
\l@dcsnotetext@l
\l@dcsnotetext@r
2175 \newcommand*{\l@emptyd@ta}{%
2176 \gdef\l@dld@ta}{%
2177 \gdef\l@drd@ta}{%
2178 \gdef\l@dcsnotetext@l{}%
2179 \gdef\l@dcsnotetext@r{}%
2180 \gdef\l@dcsnotetext{}%
2181
2182 %

```

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

```

\l@drsn@te
2183 \newcommand*{\l@dlsn@te}{%
2184 \hb@xt@ \z@{\hss\box\l@dldp@rbox\kern\ledlsnotesep}}
2185 \newcommand*{\l@drsn@te}{%
2186 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
2187
2188 %

```

\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`.

```

2189 \newcommand*{\ledllfill}{\hfil}
2190 \newcommand*{\ledrlfill}{%
2191
2192 %

```

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.

```

2193 \newcommand*{\getline@num}{%
2194   \global\advance\absline@num \@ne%
2195   \do@actions
2196   \do@ballast
2197   \ifnumberline
2198     \ifsublines@
2199       \ifnum\sub@lock<\tw@
2200         \global\advance\subline@num \@ne
2201       \fi
2202     \else
2203       \ifnum\@lock<\tw@
2204         \global\advance\line@num \@ne
2205         \global\subline@num \z@
2206       \fi
2207     \fi
2208   \fi
2209 }
2210 %

```

`\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, T_EX will be given extra encouragement to break the page here (see XI.2 p. 162).

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

```

2211 \newcount\ballast@count
2212 \newcounter{ballast}
2213 \setcounter{ballast}{0}
2214 %

```

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.

```

2215 \newcommand*{\do@ballast}{\global\ballast@count \z@
2216   \begingroup
2217     \advance\absline@num \@ne
2218     \ifnum\next@actionline=\absline@num
2219       \ifnum\next@action>-1001\relax
2220         \global\advance\ballast@count by -\c@ballast
2221       \fi
2222     \fi
2223   \endgroup}
2224 %

```


`\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.

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`.

```
2225 \newcommand*{\do@actions}{%
2226   \global\let\do@actions@next=\relax
2227   \ifnum\absline@num<\next@actionline\else
2228   %
```

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.

```
2229   \ifnum\next@action>-1001
2230     \global\page@num=\next@action
2231     \global\@firstlineofpagetrue%
2232     \ifbypage@
2233       \global\line@num=\z@ \global\subline@num=\z@
2234       \resetprevline@
2235     \fi
2236     \add@msdata@firstlineofpage%
2237   %
```

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`.)

```
2238   \else
2239     \ifnum\next@action<-4999
2240       \@l@tempcnta=-\next@action
2241       \advance\@l@tempcnta by -5001
2242       \ifsublines@
2243         \global\subline@num=\@l@tempcnta
2244       \else
2245         \global\line@num=\@l@tempcnta
2246       \fi
2247   %
```

We rescale the value in `\@l@tempcnta` so that we can use a case statement.

```
2248   \else
2249     \@l@tempcnta=-\next@action
2250     \advance\@l@tempcnta by -1000
2251     \do@actions@fixedcode
2252   \fi
2253 \fi
2254 %
```

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.

```

2255 \ifx\actionlines@list\empty
2256 \gdef\next@actionline{1000000}%
2257 \else
2258 \gl@p\actionlines@list\to\next@actionline
2259 \gl@p\actions@list\to\next@action
2260 \global\let\do@actions@next=\do@actions
2261 \fi
2262 \fi
2263 %

```

Make the recursive call, if necessary.

```

2264 \do@actions@next}
2265
2266 %

```

`\do@actions@fixedcode` This macro handles the fixed codes for `\do@actions`. It is one big case statement.

```

2267 \newcommand*{\do@actions@fixedcode}{%
2268 \ifcase\@l@dttempcnta
2269 \or% % 1001 = starting sublineation
2270 \global\sublines@true
2271 \or% % 1002 = ending sublineation
2272 \global\sublines@false
2273 \or% % 1003 = starting locking number
2274 \global\@lock=\@ne
2275 \or% % 1004 = ending locking number
2276 \ifnum\@lock=\tw@
2277 \global\@lock=\thr@@
2278 \else
2279 \global\@lock=\z@
2280 \fi
2281 \or% % 1005 = starting locking subnumber
2282 \global\sub@lock=\@ne
2283 \or% % 1006 = ending locking subnumber
2284 \ifnum\sub@lock=\tw@
2285 \global\sub@lock=\thr@@
2286 \else
2287 \global\sub@lock=\z@
2288 \fi
2289 \or% % 1007 = skipping numbering
2290 \l@dskipnumbertrue
2291 \or% % 1008 = skipping numbering in stanza
2292 \l@dskipversenumbertrue%
2293 \or% % 1009 = hiding number
2294 \l@dhidenumbertrue
2295 \or% % 1010 = inserting msdata
2296 \add@msdata%

```

```

2297 \else
2298   \led@warn@BadAction
2299 \fi}
2300
2301
2302 %

```

IX Line number printing

`\affixline@num` `\affixline@num` just puts a left line number into `\l@dld@ta` or a right line number into `\l@drd@ta` if required.

To determine whether we need to affix a line number to this line, we compute the following:

$$n = \text{int}((\text{linenum} - \text{firstlinenum}) / \text{linenumincrement})$$

$$m = \text{firstlinenum} + (n \times \text{linenumincrement})$$

(where *int* truncates a real number to an integer). *m* will be equal to *linenum* only if we are to paste a number on here. However, the formula breaks down for the first line to number (and any before that), so we check that case separately: if `\line@num ≤ \firstlinenum`, we compare the two directly instead of making these calculations.

We compute, in the scratch counter `\@l@tempcnta`, the number of the next line that should be printed with a number (*m* in the above discussion), and move the current line number into the counter `\@l@tempcntb` for comparison.

First, the case when we are within a sub-line range.

```

2303 \newcommand*{\affixline@num}{%
2304 %

```

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).

```

2305 \ifledgroupnotesL\else
2306   \ifnumberline
2307     \ifl@dskipnumber
2308       \global\l@dskipnumberfalse
2309     \else
2310       \ifsublines@
2311         \@l@tempcntb=\subline@num
2312         \ifnum\subline@num>\c@firstsublinenum
2313           \@l@tempcnta=\subline@num
2314           \advance\@l@tempcnta by-\c@firstsublinenum
2315           \divide\@l@tempcnta by\c@sublinenumincrement
2316           \multiply\@l@tempcnta by\c@sublinenumincrement
2317           \advance\@l@tempcnta by\c@firstsublinenum
2318         \else
2319           \@l@tempcnta=\c@firstsublinenum
2320         \fi
2321 %

```

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.

```
2322 \ch@cksub@l@ck
2323 %
```

Now the line number case, which works the same way.

```
2324 \else
2325 \@l@tempcntb=\line@num
2326 %
```

Check on the `\linenumberlist` If it is `\empty` use the standard algorithm.

```
2327 \ifx\linenumberlist\empty
2328 \ifnum\line@num>\c@firstlinenum
2329 \@l@tempcnta=\line@num
2330 \advance\@l@tempcnta by-\c@firstlinenum
2331 \divide\@l@tempcnta by\c@linenumincrement
2332 \multiply\@l@tempcnta by\c@linenumincrement
2333 \advance\@l@tempcnta by\c@firstlinenum
2334 \else
2335 \@l@tempcnta=\c@firstlinenum
2336 \fi
2337 \else
2338 %
```

The `\linenumberlist` was not `\empty`, so here is Wayne's numbering mechanism. This takes place in \TeX 's mouth.

```
2339 \@l@tempcnta=\line@num
2340 \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2341 \edef\sc@n@list{\def\noexpand\sc@n@list
2342 ###1,\number\@l@tempcnta,###2|\def\noexpand\rem@inder
2343 {####2}}}%
2344 \sc@n@list\expandafter\sc@n@list\rem@inder|}%
2345 \ifx\rem@inder\empty%
2346 \advance\@l@tempcnta\@ne
2347 \fi
2348 %
```

A locking check for lines, just like the version for sub-line numbers above.

```
2349 \ch@ck@l@ck
2350 \fi
2351 %
```

The following tests are true if we need to print a line number.

```
2352 \ifnum\@l@tempcnta=\@l@tempcntb
2353 \ifl@dskipversenumber\else
2354 %
```

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`

```

2355         \if@twocolumn
2356             \if@firstcolumn
2357                 \gdef\l@dld@ta{\llap{\leftlinenum}}%
2358             \else
2359                 \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2360             \fi
2361         \else
2362             \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}}%
2363                 {\@l@tempcntb=\line@margin@columns}%
2364                 {\@l@tempcntb=\line@margin}%
2365                 \ifnum\@l@tempcntb>\@ne
2366                     \advance\@l@tempcntb \page@num
2367                 \fi
2368                 \ifodd\@l@tempcntb
2369                     \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2370                 \else
2371                     \gdef\l@dld@ta{\llap{\leftlinenum}}%
2372                 \fi
2373             \fi
2374         \fi
2375     \fi
2376 %

```

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

```

2377         \f@x@l@cks
2378     \fi
2379 \fi
2380 \fi
2381 }
2382 %
2383

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck`
`\ch@ck@l@ck`
`\f@x@l@cks`

checks subline locking. If it fails, then we disable the line-number display by setting the counters to arbitrary but unequal values.

```

2384 \newcommand*{\ch@cksub@l@ck}{%
2385   \ifcase\sub@lock
2386     \or
2387       \ifnum\sublock@disp=\@ne
2388         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2389       \fi
2390     \or
2391       \ifnum\sublock@disp=\tw@ \else
2392         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2393       \fi
2394     \or
2395       \ifnum\sublock@disp=\z@
2396         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2397       \fi
2398   \fi}
2399 %

```

Similarly for line numbers.

```

2400 \newcommand*{\ch@ck@l@ck}{%
2401   \ifcase\@lock
2402     \or
2403       \ifnum\lock@disp=\@ne
2404         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2405       \fi
2406     \or
2407       \ifnum\lock@disp=\tw@ \else
2408         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2409       \fi
2410     \or
2411       \ifnum\lock@disp=\z@
2412         \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2413       \fi
2414   \fi}
2415 %

```

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

```

2416 \newcommand*{\f@x@l@cks}{%
2417   \ifcase\@lock
2418     \or
2419       \global\@lock=\tw@
2420     \or \or
2421       \global\@lock=\z@
2422     \fi
2423   \ifcase\sub@lock
2424     \or
2425       \global\sub@lock=\tw@

```

```

2426 \or \or
2427 \global\sub@lock=\z@
2428 \fi}
2429
2430 %

```

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.

```

\leftpstartnum31
\rightpstartnum32 \newif\ifsidepstartnum
\ifsidepstartnum33 \newcommand*{\affixpstart@num}{%
2434 \ifsidepstartnum
2435 \if@twocolumn
2436 \if@firstcolumn
2437 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2438 \else
2439 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2440 \fi
2441 \else
2442 \l@dtempcntb=\line@margin%
2443 \ifnum\l@dtempcntb>\@ne
2444 \advance\l@dtempcntb \page@num
2445 \fi
2446 \ifodd\l@dtempcntb
2447 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2448 \else
2449 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2450 \fi
2451 \fi
2452 \fi
2453
2454 }
2455 %
2456
2457 \newif\ifpstartnum
2458 \pstartnumtrue

```

```

2459 \newcommand*\leftpstartnum){
2460     \ifpstartnum\thepstart
2461     \kern\linenumsep\fi
2462     \global\pstartnumfalse
2463 }
2464 \newcommand*\rightpstartnum){
2465     \ifpstartnum
2466     \kern\linenumsep
2467     \thepstart
2468     \fi
2469     \global\pstartnumfalse
2470 }
2471 %

```

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.

```

2472 \list@create{\inserts@list}
2473 %

```

`\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`.

```

2474 \newcommand*\add@inserts){%
2475     \global\let\add@inserts@next=\relax
2476 %

```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```

2477     \ifx\inserts@list\empty \else
2478 %

```


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.

```

2479 \ifx\next@insert\empty
2480 \ifx\insertlines@list\empty
2481 \global\noteschanged@true
2482 \gdef\next@insert{100000}%
2483 \else
2484 \glp\insertlines@list\to\next@insert
2485 \fi
2486 \fi
2487 %

```

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.

```

2488 \ifnum\next@insert=\absline@num
2489 \glp\inserts@list\to\@insert
2490 \@insert
2491 \global\let\@insert=\undefined
2492 \global\let\next@insert=\empty
2493 \global\let\add@inserts@next=\add@inserts
2494 \fi
2495 \fi
2496 %

```

Make the recursive call, if necessary.

```

2497 \add@inserts@next}
2498
2499 %

```

`\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`.

```

2500 \newcommand{\add@Xgroupbyline}{%
2501 \unless\ifnocritical{%
2502 \def\do##1{%Looping on the series
2503 \def\do####1{%Looping on the ##1@forinserting command
2504 \ifcsdef{##1@forinserting@####1}{%
2505 \X@beforeinsertion{##1}%
2506 \if@ledgroup%
2507 \global\setbox\@nameuse{mp##1footins}=\vbox%
2508 \else%
2509 \insert\csize ##1footins\endcsize%
2510 \fi%
2511 {%
2512 \ifcsdef{Xhsize\csize{series@display##1}@##1}%
2513 {\hsize \csize{Xhsize\csize{series@display##1}@##1}}%

```

```

2514      {%
2515      \if@ledgroup%
2516        \unvbox\@nameuse{mp##1footins}%
2517      \fi%
2518      \X@atbegininsertion{##1}%
2519      \ifcsstring{series@display##1}%
2520        {%
2521          \Xledsetnormalparstuff{##1}%
2522          \rule\z@\splittopskip%
2523        }%
2524      {%
2525        \csuse{##1@forinserting@###1}%
2526        \strut\par%
2527      }%
2528      \global\csundef{##1@forinserting@###1}%
2529      }%
2530      {%
2531      }%
2532      \ifcsdef{##1@forinserting}{%
2533        \dolistcsloop{##1@forinserting}%
2534      }{%
2535        \global\csundef{##1@forinserting}%
2536      }%
2537      \dolistloop{\@series}%
2538    \fi%
2539  }%
2540
2541
2542  %

```

XI.2 Penalties

\add@penalties \add@penalties is the last macro used by \do@line. It adds up the club, widow, and interline penalties, and puts a single penalty of the appropriate size back into the paragraph; these penalties get removed by the \vsplit operation. \displaywidowpenalty and \brokenpenalty are not restored, since we have no easy way to find out where we should insert them.

In this code, \num@lines is the number of lines in the whole paragraph, and \par@line is the line we are working on at the moment. The count \@l@tempcnta is used to calculate and accumulate the penalty; it is initially set to the value of \ballast@count, which has been worked out in \do@ballast above (VIII p. 152). Finally, the penalty is checked to see that it does not go below -10000 .

```

2543 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
2544   \ifnum\num@lines>\@ne
2545     \global\advance\par@line \@ne
2546     \ifnum\par@line=\@ne
2547       \advance\@l@tempcnta \clubpenalty
2548     \fi

```

```

2549 \l@l@tempcntb=\par@line \advance\l@l@tempcntb \@ne
2550 \ifnum\l@l@tempcntb=\num@lines
2551   \advance\l@l@tempcnta \widowpenalty
2552 \fi
2553 \ifnum\par@line<\num@lines
2554   \advance\l@l@tempcnta \interlinepenalty
2555 \fi
2556 \fi
2557 \ifnum\l@l@tempcnta=\z@
2558   \relax
2559 \else
2560   \ifnum\l@l@tempcnta>-10000
2561     \penalty\l@l@tempcnta
2562   \else
2563     \penalty -10000
2564   \fi
2565 \fi}
2566
2567 %

```

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. First the first run, we don't flush the notes, as that means all the notes will be added at the end of numbered section, and so, very far of the expected position

```

2568 \newcommand*{\flush@notes}{%
2569   \iftoggle{notfirstrun@jobname.\extensionchars\the\section@num}{%
2570     \@xloop%
2571     \ifx\inserts@list\empty \else%
2572       \gl@p\inserts@list\to\@insert%
2573       \@insert%
2574       \global\let\@insert=\undefined%
2575     \repeat%
2576   }{}%
2577 }%
2578
2579
2580 %

```

\@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.

```
2581 \def\@xloop#1\repeat{%
2582   \def\body{#1\expandafter\body\fi}%
2583   \body}
2584
2585 %
```

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 `\Xtextbeforenotes` will be inserted.

```
2586 \newcommand{\set@Xtxtbeforenotes}{%
2587   \unless\ifnocritical{%
2588     \def\do##1{%
2589       \nottoggle{Xtxtbeforenotesonlyonce@##1}{%
2590         \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2591       }{}%
2592     }%
2593     \dolistloop{\@series}%
2594   \fi%
2595 }%
2596 %
```

`\set@txtbeforenotesX` The `\set@txtbeforenotesX` does the same for the `\textbeforenotesX`.

```
2597 \newcommand{\set@txtbeforenotesX}{%
2598   \unless\ifnofamiliar{%
2599     \def\do##1{%
2600       \nottoggle{txtbeforenotesonlyonceX@##1}{%
2601         \global\togglefalse{txtbeforenotesX@##1@typeset}%
2602       }{}%
2603     }%
2604     \dolistloop{\@series}%
2605   \fi%
2606 }%
2607 %
```

`\insert@Xtxtbeforenotes` `\insert@Xtxtbeforenotes{⟨series⟩}`, called when inserting a familiar footnote, will insert the text before the note if it is not already inserted. For paragraphed footnotes, it will insert it as a component of the first footnote. For other types of footnotes, it will insert it as a regular footnote.

`\insert@txtbeforenotesX` is the same for familiar footnotes.

```

2608 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2609   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2610     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2611     \ifcsvoid{Xtxtbeforenotes@#1}{-%
2612       \ifcsstring{series@display#1}{paragraph}%
2613       {\noindent\csuse{Xtxtbeforenotes@#1}}%
2614       {\expandafter\insert\csname#1footins\endcsname%
2615         \bgroup%
2616         \noindent%
2617         \ifcsdef{\csuse{series@display#1}@begin@insert}{%
2618           \csuse{\csuse{series@display#1}@begin@insert}{#1}%
2619           }{}%
2620           \strut\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}%
2621         \egroup%
2622       }%
2623     }%
2624   }%
2625 }%
2626 }%
2627
2628
2629 \newcommand{\insert@txtbeforenotesX}[1]{%
2630   \nottoggle{txtbeforenotesX@#1@typeset}{%
2631     \global\toggletrue{txtbeforenotesX@#1@typeset}%
2632     \ifcsvoid{txtbeforenotesX@#1}{-%
2633       \ifcsstring{series@displayX#1}{paragraph}%
2634       {\noindent\csuse{txtbeforenotesX@#1}}%
2635       {\expandafter\insert\csname footins#1\endcsname%
2636         \bgroup%
2637         \noindent%
2638         \ifcsdef{\csuse{series@displayX#1}@begin@insert}{%
2639           \csuse{\csuse{series@displayX#1}@begin@insert}{#1}%
2640           }{}%
2641           \strut\csuse{notefontsizeX@#1}\csuse{txtbeforenotesX@#1}%
2642         \egroup%
2643       }%
2644     }%
2645   }%
2646 }%
2647 }%
2648
2649 %
2650 %

```

XII Critical footnotes

The footnote macros are adapted from those in PLAIN T_EX, but they differ in these respects: the outer-level commands must add other commands to a list macro rather

than doing insertions immediately; there are many separate levels of the footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

XII.1 Fonts

Before getting into the details of formatting the notes, we set up some font macros. It is the notes that present the greatest challenge for our font-handling mechanism, because we need to be able to take fragments of our main text and print them in different forms: it is common to reduce the size, for example, without otherwise changing the fonts used.

`\select@lemmafонт` `\select@lemmafонт` 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.

```

2651 \def\select@lemmafонт#1|#2|#3|#4|#5|#6|#7|{\select@@lemmafонт#7|}
2652 \def\select@@lemmafонт#1/#2/#3/#4|{%
2653   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2654   \selectfont}
2655
2656 %

```

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.

```

2657 \newcommand*{\footnoteoptions@}[3]{%
2658   \def\do##1{%
2659     \ifstrequal{#1}{L}{% In Leftside
2660       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@list%
2661       \global\advance\insert@count \@ne% Increment the left insert
counter.
2662     }%
2663     {%
2664       \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@listR%
2665       \global\advance\insert@countR \@ne% Increment the right insert
counter insert.
2666     }%
2667   }%
2668   \notblank{#2}{\docsvlist{#2}}}% Parsing all options
2669 }
2670 %

```

XII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua¹TeX is used.

```

2671 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
2672   \ifstrequal{#1}{L}{%
2673     \xright@appenditem{\csxdef{footnote@luatexttextdir}{\the\textdir}}{\to\
inserts@list%Know the dir of lemma
2674     \global\advance\insert@count \@ne%
2675     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@list%Know the dir of lemma
2676     \global\advance\insert@count \@ne%
2677   }%
2678   {%
2679     \xright@appenditem{\csxdef{footnote@luatexttextdir}{\the\textdir}}{\to\
inserts@listR%Know the dir of lemma
2680     \global\advance\insert@countR \@ne%
2681     \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}{\to\
inserts@listR%Know the dir of lemma
2682     \global\advance\insert@countR \@ne%
2683   }%
2684 }
2685 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

2686 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2687   \ifstrequal{#1}{L}{%
2688     \if@RTL%
2689       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@list%Know the language used in the lemma
2690       \global\advance\insert@count \@ne%
2691     }%
2692     \else
2693       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@list%Know the language of lemma
2694       \global\advance\insert@count \@ne%
2695     }%
2696     \fi%
2697     \xright@appenditem{\csxdef{footnote@lang}{\expandonce\language}}{\to\
to\inserts@list%Know the language of lemma
2698     \global\advance\insert@count \@ne%
2699   }%
2700   {%
2701     \if@RTL
2702       \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}{\to\
inserts@listR%Know the language of lemma
2703       \global\advance\insert@countR \@ne%
2704     }%
2705     \else
2706       \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}{\to\
inserts@listR%Know the language of lemma

```

```

2704 \global\advance\insert@countR \@ne%
2705 \fi
2706 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language\language}}}{\
to\inserts@listR%Know the language of lemma
2707 \global\advance\insert@countR \@ne%
2708 }%
2709 }
2710 %

```

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`.

```

2711 \newcommand*{\footsplitskips}{%
2712 \interlinepenalty=\interfootnotelinepenalty
2713 \unless\ifl@dprintingpages%
2714 \floatingpenalty=\@MM%
2715 \fi%

```



```

2716 \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2717 \leftskip=\z@skip \rightskip=\z@skip
2718
2719 %

```

\normalfootnoterule \normalfootnoterule is a standard footnote-rule macro, for use by a footstart macro: just the same as the PLAIN T_EX footnote rule.

```

2720 \let\normalfootnoterule=\footnoterule
2721 %

```

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.

```

2722 \newcommand{\Xarrangement}[2][1,usedefault]{%
2723   \def\do##1{%
2724     \csname Xarrangement@#2\endcsname{##1}%
2725   }%
2726   \ifstrempy{#1}%
2727   {%
2728     \dolistloop{\@series}%
2729   }%
2730   {
2731     \docsvlist{#1}%
2732   }%
2733 }%
2734 %

```

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\vsizel
\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.

```

2735 \newcommand*{\Xarrangement@normal}[1]{%
2736   \csgdef{series@display#1}{normal}
2737   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
2738   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
2739   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
2740   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
2741   \expandafter\let\csname #1footnoterule\endcsname=%
2742                                     \normalfootnoterule
2743   \count\csname #1footins\endcsname=1000
2744   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2745   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2746   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2747   %

```

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.

```

2748   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
2749   side only
2750   %

```

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).

```

2750   \ifnoledgroup@else%
2751     \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2752     \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2753     \count\csname mp#1footins\endcsname=1000
2754     \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2755     \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2756     \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2757   \fi
2758 }
2759 %
2760 %

```

`\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.

```

2761 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2762 \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
2763 \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
2764 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
2765 \X@beforeinsertion{#1}%
2766 \insert\csname #1footins\endcsname{%
2767 \X@atbegininsertion{#1}%
2768 \normalvfootnote@inserted{#1}{#2}%
2769 }%
2770 }%
2771 }%
2772 %

```

\normalvfootnote@inserted The `\normalvfootnote@inserted` macro is expanded to the content to be add to a `\insert` for normal critical footnote.

```

2773 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\
normalvfootnote@inserted}[2]{%
2774 \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
2775 \csuse{Xnotefontsize@#1}%
2776 \footsplitskips
2777 \ifl@dpairing\ifl@dpaging\else%
2778 \setXnoteswidthliketwocolumns@{#1}%
2779 \fi\fi%
2780 \setXnotespositionliketwocolumns@{#1}%
2781 \spaceskip=\z@skip \xspaceskip=\z@skip%
2782 \csname #1footfmt\endcsname #2{#1}%
2783 }%
2784 %

```

```

\X@beforeinsertion 2785 \newcommand{\X@beforeinsertion}[1]{%
2786 \if@ledgroup\else%
2787 \insert@Xtxtbeforenotes{#1}%
2788 \fi%
2789 \csuse{Xbeforeinserting@#1}%
2790 }%
2791 %

```

```

\beforeinsertion@X 2792 \newcommand{\beforeinsertion@X}[1]{%
2793 \if@ledgroup\else%
2794 \insert@txtbeforenotesX{#1}%
2795 \fi%
2796 \csuse{beforeinsertingX@#1}%
2797 }%
2798 %

```

```

\X@atbegininsertion99 \newcommand{\X@atbegininsertion}[1]{%
2800   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2801 }%
2802 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote03 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
2804   \iftoggle{Xgroupbyline@#1}{%
2805     \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
2806   }%
2807   {%
2808     \global\setbox\@nameuse{mp#1footins}%
2809     \vbox{%
2810       \unvbox\@nameuse{mp#1footins}%
2811       \mpnormalvfootnote@inserted{#1}{#2}%
2812     }%
2813   }%
2814 }%
2815 %
2816 %

```

```

\mpnormalvfootnote@inserted17 \newcommand{\mpnormalvfootnote@inserted}[2]{%
2818   \noindent\csuse{Xbhooknote@#1}%
2819   \csuse{Xnotefontsize@#1}%
2820   \hsize\columnwidth%
2821   \@parboxrestore%
2822   \color@begingroup%
2823   \csname #1footfmt\endcsname #2{#1}\color@endgroup%
2824 }%
2825 %

```

`\normalfootfmt` `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 100), 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.

```

2826 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
2827   \Xstorelineinfo{#1}{#4}%
2828   \nottoggle{Xgroupbyline@#4}{\Xledsetnormalparstuff{#4}}{}%
2829   \hangindent=\csuse{Xhangindent@#4}%
2830   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2831   \nottoggle{Xgroupbyline@#4}{\rule{z@\splittopskip}}{}%
2832   {\printlinefootnote{#1}{#4}}%
2833   \print@lemma{#1}{#2}{#4}%

```

```

2834 \csuse{Xwrapcontent@#4}{#3}%
2835 \nottoggle{Xgroupbyline@#4}{\strut\par}{}%
2836 }%
2837 %

```

\normalfootstart \normalfootstart is a standard footnote-starting macro, called in the output routine whenever there are footnotes of this series to be printed: it skips a bit and then draws a rule.

Any \footstart macro must put onto the page something that takes up space exactly equal to the \skipXfootins value for the associated series of notes. \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.

```

2838 \newcommand*{\normalfootstart}[1]{%
2839 %

```

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. 223. Here is part of this algorithm, when the block of notes are ready to be printed.

```

2840 \ifdimequal{0pt}{\Xprenotes@}{}%
2841   {%
2842     \iftoggle{Xprenotes@}{%
2843       \togglefalse{Xprenotes@}%
2844       \skip\csname #1footins\endcsname=%
2845       \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
2846     }%
2847   }%
2848 }%
2849 \vskip\skip\csname #1footins\endcsname%
2850 %

```

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. 221 for the general description of the problem.

```

2851 \leftskip0pt \rightskip0pt
2852 \ifl@dpairing\else%
2853   \hsize=\old@hsize%
2854 \fi%

```

```

2855 \setXnoteswidthliketwocolumns@{#1}%
2856 \setXnotespositionliketwocolumns@{#1}%
2857 %

```

And now, print the footnote's rule to finish the footnote's introduction.

```

2858 \print@Xfootnoterule{#1}%
2859 }%
2860 %

```

\normalfootgroup \normalfootgroup is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

2861 \newcommand*{\normalfootgroup}[1]{%
2862 \csuse{Xhookgroup@#1}%
2863 \unvbox\csname #1footins\endcsname%
2864 \hsize=\old@hsize%
2865 }%
2866 %
2867 %

```

\mpnormalfootgroup A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the \Xfootgroup and \Xfootstarts macros.

```

2868 \unless\ifnoledgroup@
2869 \newcommand*{\mpnormalfootgroup}[1]{%
2870 \vskip\skip\@nameuse{mp#1footins}%
2871 \ifl@dpairing\ifparledgroup%
2872 \leavevmode\marks\parledgroup@{begin}%
2873 \marks\parledgroup@series{#1}%
2874 \marks\parledgroup@type{Xfootnote}%
2875 \fi\fi\normalcolor%
2876 \ifparledgroup%
2877 \ifl@dpairing%
2878 \else%
2879 \setXnoteswidthliketwocolumns@{#1}%
2880 \setXnotespositionliketwocolumns@{#1}%
2881 \print@Xfootnoterule{#1}%%
2882 \fi%
2883 \else%
2884 \setXnoteswidthliketwocolumns@{#1}%
2885 \setXnotespositionliketwocolumns@{#1}%
2886 \print@Xfootnoterule{#1}%%
2887 \fi%
2888 \setlength{\parindent}{0pt}%
2889 \csuse{Xhookgroup@#1}%
2890 \unvbox\csname mp#1footins\endcsname}}
2891 \fi
2892 %

```

XII.6.3 Paraphrased footnotes

The paraphrased-footnote option reformats all the footnotes of one series for a page into a single paragraph; this is especially appropriate when the notes are numerous and brief. The code is based on *The TeXbook*, pp. 398–400, with alterations for our environment. This algorithm uses a considerable amount of save-stack space: a \TeX of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

`\Xarrangement@paragraph` The `\Xarrangement@paragraph` macro sets up everything for one series of the footnotes so that they will be paraphrased; it takes the series letter as argument. We include the setting of `\count\footins` to 1000 for the footnote series just in case user is switching to paraphrased 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 paraphrased.

```

2893 \newcommand*\Xarrangement@paragraph}[1]{%
2894   \csgdef{series@display#1}{paragraph}
2895   \expandafter\let\csname #1footstart\endcsname=\parafootstart
2896   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
2897   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
2898   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
2899   \count\csname #1footins\endcsname=1000
2900   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2901   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2902   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2903   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2904   \para@footsetup{#1}
2905   %

```

And the extra setup for minipages.

```

2906 \ifnoledgroup@else
2907   \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
2908   \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
2909   \count\csname mp#1footins\endcsname=1000
2910   \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2911   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2912   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2913 \fi
2914 }
2915 %

```

`\footfudgefiddle` For paraphrased 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.

```

2916 \providecommand{\footfudgefiddle}{64}
2917 %

```

`\para@footsetup` `\footparagraph` calls the `\para@footsetup` macro to calculate a special fudge factor, which is the ratio of the `\baselineskip` to the `\hsize`. We assume that the proper value of `\baselineskip` for the footnotes (normally 9pt) has been set already. The argument of the macro is again the note series letter.

Peter Wilson thinks that `\columnwidth` should be used here for \LaTeX not `\hsize`. Peter Wilson have also included `\footfudgefiddle`.

```

2918 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}
2919 \setXnoteswidthliketwocolumns@{#1}%
2920 \ifcempty{Xwidth@#1}%
2921 {}%
2922 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
2923 \dimen0=\baselineskip
2924 \multiply\dimen0 by 1024
2925 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
2926 \csxdef{#1footfudgefactor}{%
2927 \expandafter\strip@pt\dimen0 }}
2928
2929 %

```

`\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.

```

2930 \newcommand*{\parafootstart}[1]{%
2931 \rightskip=0pt \leftskip=0pt%
2932 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2933 \ifdimequal{0pt}{\Xprenotes@}{}%
2934 {%
2935 \iftoggle{Xprenotes@}{%
2936 \togglefalse{Xprenotes@}%
2937 \skip\csname #1footins\endcsname=%
2938 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
2939 }%
2940 }%
2941 }%
2942 \vskip\skip\csname #1footins\endcsname%
2943 \setXnoteswidthliketwocolumns@{#1}%
2944 \setXnotespositionliketwocolumns@{#1}%
2945 \print@Xfootnoterule{#1}%
2946 \let\bidirectional@everypar\@empty%
2947 \noindent\leavevmode}
2948 %

```


`\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 `\Xunvxh`: `unvbox`, extract the last line, and `unhbox` it.

Doing things this way has an important consequence: as Michael pointed out, you really can't put an explicit line-break into a note built in a `vbox` the way we are doing.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. 173 above). We need to do this, since `\footfudgefactor` is calculated on the assumption that the

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 `\Xunvxh` macro since it is publicly documented.

³²'Line Breaking', p. 610.

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.

```

2949 \newcommand*{\paravfootnote}[2]{%
2950   \csuse{Xbeforeinserting@#1}%
2951   \insert\csname #1footins\endcsname
2952   \bgroup
2953     \csuse{Xnotefontsize@#1}
2954     \footsplitskips
2955     \setbox0=\vbox{\hsize=\maxdimen%
2956       \let\bidir@RTL@everypar\@empty%
2957       \insert@Xtxtbeforenotes{#1}%
2958       \noindent\csuse{Xhooknote@#1}%
2959       \csname #1footfmt\endcsname #2{#1}}%
2960     \setbox0=\hbox{\Xunvvh{0}{#1}}%
2961     \dp0=0pt
2962     \ht0=\csname #1footfudgefactor\endcsname\wd0
2963   %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

2964   \if@RTL\noindent \leavevmode\fi\box0%
2965   \penalty0
2966   \egroup}
2967
2968 %

```

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.

```

2969 \newcommand*{\mpparavfootnote}[2]{%
2970   \global\setbox\@nameuse{mp#1footins}\vbox{%
2971     \unvbox\@nameuse{mp#1footins}%
2972     \csuse{Xnotefontsize@#1}
2973     \footsplitskips
2974     \setbox0=\vbox{\hsize=\maxdimen%
2975       \let\bidir@RTL@everypar\@empty%
2976       \insert@Xtxtbeforenotes{#1}%
2977       \noindent\color@begingroup%

```

```

2978 \csuse{Xhooknote@#1}%
2979 \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
2980 \setbox0=\hbox{\Xunvxh{0}{#1}}%
2981 \dp0=\z@
2982 \ht0=\csname #1footfudgefactor\endcsname\wd0
2983 \box0
2984 \penalty0
2985 }}
2986
2987 %

```

\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`.

```

2988 \newcommand*{\Xunvxh}[2]{%
2989 \setbox0=\vbox{\unvbox#1%
2990 \global\setbox1=\lastbox}%
2991 \unhbox1
2992 \unskip % remove \rightskip,
2993 \unskip % remove \parfillskip,
2994 \unpenalty % remove \penalty of 10000,
2995 \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
2996
2997 %

```

\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).

```

2998 \newcommand*{\parafootfmt}[4]{%
2999 \Xstorelineinfo{#1}{#4}%
3000 \Xinsertparafootsep{#4}%
3001 \ledsetnormalparstuff@common%
3002 \printlinefootnote{#1}{#4}%
3003 \print@lemma{#1}{#2}{#4}%
3004 \csuse{Xwrapcontent@#4}{#3}%
3005 \penalty-10 }
3006 %

```

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.

```

3007 \newcommand*{\parafootgroup}[1]{%
3008   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3009   \unvbox\csname #1footins\endcsname
3010   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3011   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3012   \makehboxofhboxes
3013   \setbox0=\hbox{\unhbox0 \removehboxes}%
3014   \csuse{Xbhookgroup@#1}%
3015   \csuse{Xnotefontsize@#1}%
3016   \unhbox0\par%
3017   \global\hsize=\old@hsize%
3018 }%
3019
3020 %

```

`\mpparafootgroup` The minipage version.

```

3021 \newcommand*{\mpparafootgroup}[1]{%
3022   \setXnoteswidthliketwocolumns@{#1}%
3023   \vskip\skip\@nameuse{mp#1footins}
3024   \ifl@dpairing\ifparledgroup%
3025     \leavevmode\marks\parledgroup@{begin}%
3026     \marks\parledgroup@series{#1}%
3027     \marks\parledgroup@type{Xfootnote}%
3028   \fi\fi\normalcolor
3029   \ifparledgroup%
3030     \ifl@dpairing%
3031     \else%
3032       \setXnoteswidthliketwocolumns@{#1}%
3033       \setXnotespositionliketwocolumns@{#1}%
3034       \print@Xfootnoterule{#1}%
3035     \fi%
3036   \else%
3037     \setXnoteswidthliketwocolumns@{#1}%
3038     \setXnotespositionliketwocolumns@{#1}%
3039     \print@Xfootnoterule{#1}%
3040   \fi%
3041   \unvbox\csname mp#1footins\endcsname
3042   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
3043   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
3044   \makehboxofhboxes
3045   \setbox0=\hbox{\unhbox0 \removehboxes}%
3046   \csuse{Xbhookgroup@#1}%
3047   \csuse{Xnotefontsize@#1}%

```

```

3048 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3049 \unhbox0\par}}
3050
3051 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes\newcommand*\makehboxofhboxes{\setbox0=\hbox{}}%
\removehboxes\loop
3054 \unpenalty
3055 \setbox2=\lastbox
3056 \ifhbox2
3057 \setbox0=\hbox{\box2\unhbox0}%
3058 \repeat}
3059
3060 \newcommand*\removehboxes{\setbox0=\lastbox
3061 \ifhbox0{\removehboxes}\unhbox0 \fi}
3062
3063 %

```

Insertion of the footnotes separator The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num\newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep\iflabeledRcol{%
3066 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
3067 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3068 {\ifcsequal{prevline#1}{lineinfo@}%
3069 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}}%
3070 {\csuse{Xparafootsep@#1}}}%
3071 }%
3072 {\csuse{Xparafootsep@#1}}%
3073 }%
3074 {}%
3075 \global\csname #1prevpage@numR\endcsname=\page@numR%
3076 \else%
3077 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%
3078 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
3079 {\ifcsequal{prevline#1}{lineinfo@}%
3080 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}}%
3081 {\csuse{Xparafootsep@#1}}}%
3082 }%
3083 {\csuse{Xparafootsep@#1}}%
3084 }%
3085 {}%
3086 \global\csname #1prevpage@num\endcsname=\page@num%
3087 \fi%

```

```

3088 }
3089 %

```

XII.6.4 Columnar footnotes

Common tools

`\rigidbalance` We will now define macros for three-column notes and two-column notes. Both sets of macros will use `\rigidbalance`, which splits a box (#1) into a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\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`.

```

3090 \newcount\@k \newdimen\@h
3091 \newcommand*\Xrigidbalance}[3]{%
3092     \hsize=\expandafter\dimexpr\csuse{Xwidth@\@currentseries}\relax%
3093     \rigidbalance{#1}{#2}{#3}%
3094 }%
3095
3096 \newcommand*\rigidbalanceX}[3]{%
3097     \hsize=\expandafter\dimexpr\csuse{widthX@\@currentseries}\relax%
3098     \rigidbalance{#1}{#2}{#3}%
3099 }%
3100
3101 \newcommand*\rigidbalance}[3]{%
3102     \setbox0=\box#1 \@k=#2 \@h=#3%
3103     \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3104       \valign{##\vfil\cr\dosplits}}}%
3105
3106 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splitoff
3107   \global\advance\@k-1\cr\dosplits\fi}
3108
3109 \newcommand*\splitoff{\dimen0=\ht0
3110   \divide\dimen0 by\@k \advance\dimen0 by\@h
3111   \setbox2 \vsplit0 to \dimen0
3112   \unvbox2 }
3113
3114 %

```

Three columns

```

\Xarrangement@threecol 15 \newcommand*{\Xarrangement@threecol}[1]{%
3116 \csgdef{series@display#1}{threecol}
3117 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3118 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3119 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
3120 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3121 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3122 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3123 \threecolfootsetup{#1}
3124 %

```

The additional setup for minipages.

```

3125 \ifnoledgroup@else
3126 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3127 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3128 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3129 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3130 \mpthreecolfootsetup{#1}
3131 \fi
3132 }
3133 %
3134 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XII.6.2 p. 173 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.

```

3135 \newcommand*{\threecolfootsetup}[1]{%
3136 \count\csname #1footins\endcsname 333
3137 \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3138 \multiply\dimen\csname #1footins\endcsname \thr@@}
3139 %

```

`\mpthreecolfootsetup` The setup for minipages.

```

3140 \newcommand*\mpthreecolfootsetup}[1]{%
3141   \count\csname mp#1footins\endcsname 333
3142   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3143
3144 %

```

\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).

```

3145 \notbool{parapparatus@}\newcommand*\newcommand*\threecolvfootnote}[2]{%
3146   \iftoggle{Xgroupbyline@#1}{%
3147     \prepare@Xgroupbyline{#1}{#2}\threecolvfootnote@inserted}%
3148   }%
3149   {%
3150     \X@beforeinsertion{#1}%
3151     \insert\csname #1footins\endcsname{%
3152       \threecolvfootnote@inserted{#1}{#2}%
3153     }%
3154   }%
3155 }%
3156 %

```

```

\threecolvfootnote@inserted 3157 \notbool{parapparatus@}\newcommand*\newcommand*\threecolvfootnote@inserted}[2]{%
3158   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3159   \noindent\csuse{Xhooknote@#1}%
3160   \csuse{Xnotefontsize@#1}%
3161   \footsplitskips%
3162   \csname #1footfmt\endcsname #2{#1}%
3163 }%
3164 %

```

\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.

```

3165 \notbool{parapparatus@}\newcommand*\newcommand*\threecolfootfmt}[4]{%
3166   \Xstorelineinfo{#1}{#4}%
3167   \threecol@begin@insert{#4}%
3168   \hspace{\parindent}%

```



```

3169 \printlinefootnote{#1}{#4}%
3170 \print@lemma{#1}{#2}{#4}%
3171 \csuse{Xwrapcontent@#4}{#3}%
3172 \nottoggle{Xgroupbyline@#4}%
3173   {\strut\par\allowbreak}%
3174   {}%
3175 }%
3176 %

```

\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`.

```

3177 \newcommand{\threecol@begin@insert}[1]{%
3178   \normal@pars%
3179   \nottoggle{Xgroupbyline@#1}%
3180   {\hspace \csuse{Xhsizethreecol@#1}}%
3181   {}%
3182   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3183   \tolerance=5000%
3184   \hangindent=\csuse{Xhangindent@#1}%
3185   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3186   \@tempdima=\parindent%
3187   \csuse{Xcolalign@#1}%
3188   \parindent=\@tempdima%
3189   \strut%
3190 }%
3191 %

```

\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.

```

3192 \newcommand*\threecolfootgroup}[1]{%
3193   \csuse{Xhookgroup@#1}\par%
3194   \splittopskip=\ht\strutbox
3195   \expandafter
3196   \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
3197 %

```

\mpthreecolfootgroup The setup for minipages.

```

3198 \newcommand*\mpthreecolfootgroup}[1]{%

```

```

3199 \vskip\skip\@nameuse{mp#1footins}
3200 \ifl@dpairing\ifparledgroup%
3201   \leavevmode\marks\parledgroup@{begin}%
3202   \marks\parledgroup@series{#1}%
3203   \marks\parledgroup@type{Xfootnote}%
3204 \fi\fi\normalcolor
3205 \ifparledgroup%
3206   \ifl@dpairing%
3207   \else%
3208     \setXnoteswidthliketwocolumns@{#1}%
3209     \setXnotespositionliketwocolumns@{#1}%
3210     \print@Xfootnoterule{#1}%
3211   \fi%
3212 \else%
3213   \setXnoteswidthliketwocolumns@{#1}%
3214   \setXnotespositionliketwocolumns@{#1}%
3215   \print@Xfootnoterule{#1}%
3216 \fi%
3217 \csuse{Xbhookgroup@#1}\par%
3218 \splittopskip=\ht\strutbox
3219 \expandafter
3220 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3221
3222 %

```

Two columns

```

\Xarrangement@twocol 323 \newcommand*\Xarrangement@twocol}[1]{%
3224   \csgdef{series@display#1}{twocol}
3225   \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3226   \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3227   \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3228   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3229   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3230   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3231   \twocolfootsetup{#1}
3232 %

```

The additional setup for minipages.

```

3233 \ifnoledgroup@else
3234   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3235   \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
3236   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3237   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3238   \mptwocolfootsetup{#1}
3239 \fi
3240 }
3241
3242 %

```

`\twocolfootsetup` Here is a series of macros which are very similar to their three-column counterparts. In this case, each note is assumed to contribute only a half a line of text. And the notes are set in columns giving a gap between them of one tenth of the `\hsiz`.

```

\twocolvfootnote@inserted
\twocolfootfmt
\twocolfootgroup
3243 \newcommand*{\twocolfootsetup}[1]{%
3244   \count\csname #1footins\endcsname 500
3245   \csxdef{default@#1footins}{500}%
3246   \multiply\dimen\csname #1footins\endcsname \tw@}
3247   %

3248 \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolvfootnote}[2]{%
3249   \iftoggle{Xgroupbyline@#1}{%
3250     \prepare@Xgroupbyline{#1}{#2}{\twocolvfootnote@inserted}%
3251   }{%
3252     \X@beforeinsertion{#1}%
3253     \insert\csname #1footins\endcsname{%
3254       \twocolvfootnote@inserted{#1}{#2}%
3255     }%
3256   }%
3257 }%
3258 \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolvfootnote@inserted}[2]{%
3259   \hsiz=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3260   \noindent\csuse{Xhooknote@#1}%
3261   \csuse{Xnotefontsize@#1}%
3262   \footsplitskips%
3263   \csname #1footfmt\endcsname #2{#1}%
3264 }%
3265   %

3266 \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolfootfmt}[4]{% 4th
   arg is optional, for backward compatibility
3267   \Xstorelineinfo{#1}{#4}%
3268   \twocol@begin@insert{#4}%
3269   \hspace{\parindent}%
3270   \printlinefootnote{#1}{#4}%
3271   \print@lemma{#1}{#2}{#4}%
3272   \csuse{Xwrapcontent@#4}{#3}%
3273   \nottoggle{Xgroupbyline@#4}%
3274     {\strut\par\allowbreak}%
3275   }%
3276 }%
3277   %

3278 \newcommand{\twocol@begin@insert}[1]{%
3279   \normal@pars%
3280   \hsiz \csuse{Xsizetwocol@#1}%
3281   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3282   \tolerance=5000%
3283   \hangindent=\csuse{Xhangindent@#1}%

```

```

3284 \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3285 \@tempdima=\parindent%
3286 \csuse{Xcolalign@#1}%
3287 \parindent=\@tempdima%
3288 \strut%
3289 }%
3290
3291 \newcommand*{\twocolfootgroup}[1]{%
3292   \csuse{Xbhookgroup@#1}\par%
3293   \splittopskip=\ht\strutbox
3294   \expandafter
3295   \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
3296
3297 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3298 \newcommand*{\mptwocolfootgroup}[1]{%
3299   \count\csname mp#1footins\endcsname 500
3300   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3301 %
3302
3303 \newcommand*{\mptwocolfootgroup}[1]{%
3304   \vskip\skip\@nameuse{mp#1footins}
3305   \ifl@dpairing\ifparledgroup%
3306     \leavevmode\marks\parledgroup@{begin}%
3307     \marks\parledgroup@series{#1}%
3308     \marks\parledgroup@type{Xfootnote}%
3309   \fi\fi\normalcolor
3310   \ifparledgroup%
3311     \ifl@dpairing%
3312     \else%
3313       \setXnoteswidthliketwocolumns@{#1}%
3314       \setXnotespositionliketwocolumns@{#1}%
3315       \print@Xfootnoterule{#1}%
3316     \fi%
3317   \else%
3318     \setXnoteswidthliketwocolumns@{#1}%
3319     \setXnotespositionliketwocolumns@{#1}%
3320     \print@Xfootnoterule{#1}%
3321   \fi%
3322   \csuse{Xbhookgroup@#1}\par%
3323   \splittopskip=\ht\strutbox
3324   \expandafter
3325   \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}
3326 %

```

XII.7 Critical notes presentation

Here, we define some commons macro which are used in order to print a critical notes, that is a note with 1) line number 2) lemma 3) lemma separator 4) text associated to the lemma.

XII.7.1 Font tools

`\endashchar` The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals does not contain an en-dash or square brackets, and its period and comma are in odd locations. To allow use of the standard footnote macros with such fonts, we use the following macros for certain characters.

The `\endashchar` macro is simply an en-dash from the normal font and is immune to changes in the surrounding font. The same goes for the full stop. These two are used in `\printlines`. The right bracket macro is the same again; it crops up in `\normalfootfmt` and the other footnote macros for controlling the format of the footnotes.

Note that these commands are not directly called by `reledmac`, but are enclosed as default value of specific hooks. Consequently, people should not redefine them, but use instead the `\Xlinrangeseparator`, `\Xendlinrangeseparator`, `\Xsublinesep`, `\Xendsublinesep` and `\Xlemmaseparator` macros.

With `polyglossia`, each critical note has a `\footnote@lang` which shows the language of the lemma, and which can be used to switch the bracket from right to left.

```

3327 \def\endashchar{\textnormal{--}}
3328
3329 \newcommand*{\fullstop}{\textnormal{.}}
3330 \def\Xsublinesep@side{\fullstop}
3331
3332 \newcommand*{\rbracket}{\textnormal{%
3333   \csuse{text\csuse{footnote@lang}}{%
3334     \ifluatex%
3335       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]}{\thinspace
3336     }%
3337     \else%
3338       \thinspace}%
3339     \fi}%
3340   }}
3341
3342 %

```

XII.7.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

3343 \newcommand{\printpstart}[0]{%

```

```

3344 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3345 \ifledRcol%
3346 \thepstartR%
3347 \else%
3348 \thepstartL%
3349 \fi%
3350 }{%
3351 \thepstart%
3352 }%
3353 }
3354 %

```

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).

```

3355 %
3356 \newcommand{\print@lemma}[3]{%
3357 \bgroup%
3358 \nottoggle{Xlemmadisablefontselection@#3}%
3359 {\select@lemmafnt#1|}%
3360 }%
3361 \bgroup%
3362 \csuse{Xlemmafnt@#3}%Deprecated
3363 \csuse{Xwraplemma@#3}{#2}%
3364 \egroup%
3365 \egroup%
3366 \iftoggle{nosep@}{%
3367 \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3368 \relax%
3369 }%
3370 {\ifcempty{Xlemmaseparator@#3}%
3371 {%
3372 \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3373 \relax%
3374 }%
3375 {%
3376 \nobreak%
3377 \hskip\csuse{Xbeforelemmaseparator@#3}%
3378 \csuse{Xlemmaseparator@#3}%
3379 \hskip\csuse{Xafterlemmaseparator@#3}%
3380 \relax%
3381 }%
3382 }%
3383 }%
3384 %

```

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.

```

3385 \newcommand{\Xstorelineinfo}[2]{%
3386   \l@dp@rsefootspec#1|{%
3387     \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3388       \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3389         \l@dparsedendline - \l@dparsedendsub}%
3390     }%
3391     {%
3392       \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3393     }%
3394   }%

```

\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`.

```

3395 \newcommand{\printlinefootnote}[2]{%
3396   \iftoggle{nonum@}{%Try if the line number must printed for this specific
3397     not (by default, yes)
3398     \hspace{\csuse{Xinplaceofnumber@#2}}}%
3399   }%
3400   {%
3401     \iftoggle{Xnonumber@#2}{%Try if the line number must printed (by
3402       default, yes)
3403       {%
3404         \hspace{\csuse{Xinplaceofnumber@#2}}}%
3405       }%
3406       {%
3407         \iftoggle{Xnumberonlyfirstinline@#2}{% If for this series the
3408           line number must be printed only in the first time.
3409           {%
3410             \ifcsdef{prevline#2}%
3411             {%Be sure the \prevline exists.
3412             \ifcsequal{prevline#2}{\lineinfo@}{%Try it
3413             {%
3414               \ifcsempy{Xsymlinenum@#2}%
3415             }%
3416             \hspace{\csuse{Xinplaceofnumber@#2}}}%
3417           }%

```

```

3418             {%
3419             \printlinefootnotearea{#1}{#2}%
3420             }%
3421         }%
3422         {%
3423         \printlinefootnotearea{#1}{#2}%
3424         }%
3425     }%
3426     {%
3427     \printlinefootnotearea{#1}{#2}%
3428     }%
3429     \csxdef{prevline#2}{\lineinfo@}%
3430 }%
3431 }%
3432 }%
3433 }%
3434 }
3435 %

```

\printsymlinefootnotearea This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

3436 \newcommand{\printsymlinefootnotearea}[1]{%
3437   \hspace{\csuse{Xbeforesymmlinenumber@#1}}%
3438   \csuse{Xnotenumfont@#1}%
3439   \ifdimequal{\csuse{Xboxsymmlinenumber@#1}}{\z@}%
3440     {\csuse{Xsymmlinenumber@#1}}%
3441     {\hbox to \csuse{Xboxsymmlinenumber@#1}%
3442       {\csuse{Xsymmlinenumber@#1}\hfill}%
3443     }%
3444   \hspace{\csuse{Xaftersymmlinenumber@#1}}%
3445 }%
3446 %

```

\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.)

```

3447 \newcommand{\printlinefootnotearea}[2]{%
3448   \printXbeforenumber{#2}%
3449   \csuse{Xnotenumfont@#2}%
3450   \boxfootnotenumbers{#1}{#2}%
3451   \printXafternumber{#2}%
3452 }%
3453 %

```

\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.


```

3454 \newcommand{\boxfootnotenumbers}[2]{%
3455   \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
3456     \printlinefootnotenumbers{#1}{#2}%
3457   }%
3458   {%
3459     \hbox to \csuse{Xboxlinenum@#2}%
3460     {%
3461       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3462       \printlinefootnotenumbers{#1}{#2}%
3463       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
3464     }%
3465   }%
3466 }%
3467 %

```

\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.

```

3468 \newcommand{\printlinefootnotenumbers}[2]{%
3469   \xdef\@currentseries{#2}%
3470   \ifboolexpr{%
3471     (togl{Xpstart@#2} and bool{numberpstart})%
3472     or togl{Xpstarteverytime@#2}}%
3473   {\printpstart}{}%
3474   \iftoggle{Xstanza@#2}{%
3475     \ifnumberstanza%
3476       \printstanza%
3477       \csuse{Xstanzaseparator@#2}%
3478     \fi%
3479   }{%}%
3480   \iftoggle{Xonlypstart@#2}{%}%
3481   \csuse{Xtxtbeforenumber@#2}%
3482   \printlines#1|\ifledRcol@{\@Rlineflag\fi}|}%
3483 }%
3484 %

```

\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.)

```

3485 \newcommand{\printXbeforenumber}[1]{%
3486   \hspace{\csuse{Xbeforenumber@#1}}%
3487 }%
3488 %

```

\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

```

3489 \newcommand{\printXafternumber}[1]{%
3490   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%

```

```

3491 \hspace{\csuse{Xafternumber@#1}}%
3492 }%
3493 %

```

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. 100: the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma.

edmac’ creator have defined six boolean in order to know which component of line number description we have to print:

- `\ifl@d@pnum` for page numbers;
- `\ifl@d@ssub` for starting sub-line;
- `\ifl@d@elin` for ending line;
- `\ifl@d@esl` for ending sub-line; and
- `\ifl@d@dash` for the dash between the starting and ending groups.

There is no boolean for the line number because it is always printed.

Maïeul Rouquette has added `\ifl@d@Xtwolines` and `\ifl@d@Xmorethantwolines` to print a symbol which stands for “and subsequent” when there are two, three or more lines.

```

\ifl@d@pnum94 \newif\ifl@d@pnum
\ifl@d@ssub95 \newif\ifl@d@ssub
\ifl@d@elin96 \newif\ifl@d@elin
\ifl@d@esl97 \newif\ifl@d@esl
\ifl@d@dash98 \newif\ifl@d@dash
\ifl@d@Xtwolines99 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines3500 \newif\ifl@d@Xmorethantwolines%
3501 %

```

```

\l@dparsefootspec \l@dparsefootspec{<spec>}{<lemma>}{<text>}} parses a footnote specification. <lemma>
\l@dp@rsefootspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@dparsedstartpage ber and lemma font specifier in \l@d@nums style format. The real work is done by
\l@dparsedstartline \l@dp@rsefootspec which defines macros holding the numeric values. In many cases,
\l@dparsedstartsub this last command is called directly. Just a reminder of the arguments:
\l@dparsedendpage \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dparsedendline \printlines start-page | line | subline | end-page | line | subline | fontflag
\l@dparsedendsub \newcommand*{\l@dparsefootspec}[3]{\l@dp@rsefootspec#1|}
3502
3503 \def\l@dp@rsefootspec#1|#2|#3|#4|#5|#6|#7|{%
3504 \gdef\l@dparsedstartpage{#1}%
3505 \gdef\l@dparsedstartline{#2}%

```

```

3506 \gdef\l@dparsedstartsub{#3}%
3507 \gdef\l@dparsedendpage{#4}%
3508 \gdef\l@dparsedendline{#5}%
3509 \gdef\l@dparsedendsub{#6}%
3510 }
3511 %

```

Initialise the several number value macros.

```

3512 \def\l@dparsedstartpage{0}%
3513 \def\l@dparsedstartline{0}%
3514 \def\l@dparsedstartsub{0}%
3515 \def\l@dparsedendpage{0}%
3516 \def\l@dparsedendline{0}%
3517 \def\l@dparsedendsub{0}%
3518
3519 %

```

\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`.

```

3520 \newcommand*{\setprintlines}[6]{%
3521 \l@d@pnumfalse \l@d@dashfalse
3522 %

```

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

```

3523 \ifbypage@
3524 \ifnum#4=#1 \else
3525 \l@d@pnumtrue
3526 \l@d@dashtrue
3527 \fi
3528 \fi
3529 %

```

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.

```

3530 \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
3531 \ifnum#2=#5 \else
3532 \l@d@elintrue
3533 \l@d@dashtrue
3534 \fi
3535 %

```

We print the starting sub-line if it is nonzero.

```

3536 \l@d@ssubfalse
3537 \ifnum#3=0 \else
3538 \l@d@ssubtrue
3539 \fi
3540 %

```

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.

```

3541 \l@d@eslfalse
3542 \ifnum#6=0 \else
3543   \ifnum#6=#3
3544     \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3545   \else
3546     \l@d@esltrue
3547     \l@d@dashtrue
3548   \fi
3549 \fi%
3550 %

```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```

3551 \ifl@d@dash%
3552 \ifbool{expr{togl{fulllines@} or test{\ifcempty{Xtwolines@}
@currentseries}}}%
3553 {}%
3554 {}%
3555 \setistwofollowinglines{#1}{#2}{#4}{#5}%
3556 \ifbool{expr{
3557   (%
3558     togl {Xtwolinesbutnotmore@}\@currentseries}%
3559     and not%
3560     (%
3561       bool {istwofollowinglines@}%
3562     )%
3563   )%
3564   or%
3565   (%
3566     (not test{\ifnumequal{#1}{#4}})%
3567     and togl{Xtwolinesonlyinsamepage@}\@currentseries}%
3568   )%
3569   }%
3570 {}%
3571 {}%
3572 \l@d@dashfalse%
3573 \l@d@Xtwolinestrue%
3574 \l@d@elinfalse%
3575 \l@d@eslfalse%
3576 \ifcempty{Xmorethantwolines@}\@currentseries}%
3577 {}%
3578 {ifistwofollowinglines@}\else%
3579   \l@d@Xmorethantwolinestrue%
3580 \fi%
3581 }%
3582 }%
3583 }%

```

```

3584 \fi%
3585 %

```

End of \setprintlines.

```

3586 }%
3587 %

```

\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.

```

3588 \newif\ifistwofollowinglines@%
3589 \newcommand{\setistwofollowinglines}[4]{%
3590   \ifcsdef{lastlinenumberon@#1}%
3591     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
3592     {\numdef{\tmp}{0}}}%
3593   \istwofollowinglines@false%
3594   \ifnumequal{#4-#2}{1}%
3595     {\istwofollowinglines@true}%
3596     {\ifbypage@%
3597       \ifnumequal{#3-#1}{1}%
3598       {%
3599         \ifnumequal{#2}{\tmp}%
3600         {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
3601         {}%
3602       }%
3603     }%
3604   \fi%
3605 }%
3606 }%
3607 %

```

`\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

```
3608 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3609   \begingroup%
3610   %
```

If we use Lua_T_EX, ensure we use good text's direction.

```
3611   \ifluatex%
3612     \edef\@tmp{\the\textdir}%
3613     \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
    spurious space (bug #397)
3614   \fi%
3615   %
```

Decide which part of line number components we will print.

```
3616   \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3617   %
```

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.

```
3618   \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3619     {\bgroup}%
3620     {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
    hfill}%
3621   \ifl@d@pnum%
3622     \wrap@edcrossref{\@this@crossref@start}{#1}%
3623     \csuse{Xpagelinesep@\@currentseries}%
3624   \fi%
3625   \wrap@edcrossref{\@this@crossref@start}{%
    \linenumrep{#2}%
3626     \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3627   }%
3628   \ifl@d@ssub%
3629     \csuse{Xsublinesep@\@currentseries}%
3630     \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3631   \fi
3632   \egroup%
3633   %
3634   %
```

Then print the dash + end line number, or the range symbol.

```
3635   \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3636     {\bgroup}%
3637     {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
3638   \ifl@d@Xtwolines%
```

```

3639 \ifl@d@Xmorethantwolines%
3640 \csuse{Xmorethantwolines@ \@currentseries}%
3641 \else%
3642 \csuse{Xtwolines@ \@currentseries}%
3643 \fi%
3644 \else%
3645 \ifl@d@dash%
3646 \ifdefined\linangesep%
3647 \linangesep%
3648 \else%
3649 \csuse{Xlinangeseparator@ \@currentseries}%
3650 \fi%
3651 \fi%
3652 \ifl@d@pnum%
3653 \wrap@edcrossref{\@this@crossref@end}{#4}%
3654 \csuse{Xpagelinesep@ \@currentseries}%
3655 \fi%
3656 \ifl@d@elin%
3657 \wrap@edcrossref{\@this@crossref@end}{%
3658 \linenumrep{#5}%
3659 \iftoggle{Xlineflag@ \@currentseries}{#8}{}}%
3660 }%
3661 \fi%
3662 \ifl@d@esl%
3663 \ifl@d@elin%
3664 \csuse{Xsublinesep@ \@currentseries}%
3665 \fi%
3666 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3667 \fi%
3668 \fi%
3669 \ifdimequal{\csuse{Xboxendlinenum@ \@currentseries}}{0pt}%
3670 {}%
3671 {\hfill}%Prevent underfull hbox
3672 \egroup%
3673 \endgroup%
3674 }%
3675 %

```

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`.

```

3676 \newcommand{\prepare@Xgroupbyline}[3]{%
3677 \iftoggle{Xgroupbylineseparetwolines@#1}{%
3678 \l@dparsfootspec#2%
3679 \ifcsdef{#1@forinserting@ \l@dparsedendpage- \l@dparsedendline- \l@dparsedendsub}%
3680 {%
3681 \csgappto%

```

```

3682     {#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\
l@dparsedendsub}%
3683     {%
3684         \ifcempty{Xsymlinenum@#1}%
3685         {\csuse{Xparafootsep@#1}}%
3686         {}%
3687         #3{#1}{#2}%
3688         \hskip\csuse{Xafternote@#1}\relax%
3689     }%
3690 }%
3691 {%
3692     \csdef%
3693     {#1@forinserting@\l@dparsedendpage-\l@dparsedendline-\
l@dparsedendsub}%
3694     {%
3695         #3{#1}{#2}%
3696         \hskip\csuse{Xafternote@#1}\relax%
3697     }%
3698 }%
3699 \listcsxadd{#1@forinserting}{\l@dparsedendpage-\l@dparsedendline-\
l@dparsedendsub}%
3700 }{%
3701     \ifcsdef{#1@forinserting@all}{%
3702         \csgappto%
3703         {#1@forinserting@all}%
3704         {%
3705             \ifcempty{Xsymlinenum@#1}%
3706             {\csuse{Xparafootsep@#1}}%
3707             {}%
3708             #3{#1}{#2}%
3709             \hskip\csuse{Xafternote@#1}\relax%
3710         }%
3711     }%
3712     {%
3713         \csdef%
3714         {#1@forinserting@all}%
3715         {%
3716             #3{#1}{#2}%
3717             \hskip\csuse{Xafternote@#1}\relax%
3718         }%
3719     }%
3720     \listcsgadd{#1@forinserting}{all}%
3721 }%
3722 }%
3723 %

```


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`.

```
3724 \providecommand*\multiplefootnotemarker}{3sp}
3725 \providecommand*\multfootsep{\textsuperscript{\normalfont,}}
3726
3727 %
```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the `memoir` class.

```
3728 \providecommand*\m@mmf@prepare}{%
3729 \kern-\multiplefootnotemarker
3730 \kern\multiplefootnotemarker\relax}
3731 %
```

`\m@mmf@check` This may have been defined in the `memoir` class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```
3732 \providecommand*\m@mmf@check}{%
3733 \ifdim\lastkern=\multiplefootnotemarker\relax
3734 \edef\@x@sf{\the\spacefactor}%
3735 \unkern
3736 \multfootsep
3737 \spacefactor\@x@sf\relax
3738 \fi}
3739
3740 %
```

We have to modify `\@footnotetext` and `\@footnotemark`. However, if `memoir` is used the modifications have already been made.

```
3741 \@ifclassloaded{memoir}{\}%
3742 %
```

`\@footnotetext` Add `\m@mmf@prepare` at the end of `\@footnotetext`.

```
3743 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}
3744 %
```

`\@footnotemark` Modify `\@footnotemark` to cater for adjacent `\footnotes`.

```
3745 \patchcmd{\@footnotemark}
3746   {\nobreak}
3747   {\m@mmf@check
3748    \nobreak
3749   }
3750   {}{}
3751 \patchcmd{\@footnotemark}
3752   {\@makefnmark}
3753   {\@makefnmark
3754    \m@mmf@prepare
3755   }
3756   {}{}
3757 %
3758 %
```

Finished the modifications for the non-memoir case.

```
3759 }
3760
3761 %
```

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.

```
3762 \pretocmd{\@footnotetext}{%
3763   \ifnumberedpar@
3764     \edtext{}{\l@dbfnote{#1}}}%
3765   \else
3766   }{}{}
3767 \apptocmd{\@footnotetext}{\fi}{}{}%
3768 %
```

`\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`

```

3769 \patchcmd%
3770   {\footnote}%
3771   {\stepcounter\@mpfn}%
3772   {%
3773   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3774   l@dprintingcolumns}}{%
3775     \global\advance\footnote@reading by \@ne%
3776     \get@thisfootnote%
3777     \get@fnmark{\thisc@footnote}%
3778     \ifcsdef{footnotereading\the\footnote@reading=typeset}%
3779     {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
typeset}}}%
3780     {\setcounter{\@mpfn}{\footnote@reading}}%
3781   }{%
3782     \stepcounter\@mpfn%
3783   }%
3784 }%
3785 {}
3786 {}
3787
3788 \newcommand{\get@thisfootnote}{%
3789   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3790     \protected@xdef\thisc@footnote{\the\footnote@reading}%
3791   }{%
3792     \protected@xdef\thisc@footnote{\the\c@footnote}%
3793   }%
3794 }%
3795
3796 \newcommand{\l@dbfnote}[1]{%
3797   \get@thisfootnote%
3798   \gdef\@tag{#1\relax}%
3799   \ifledRcol%
3800     \xright@appenditem{%
3801       \ifdefined\Hy@footnote@currentHref%
3802         \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3803       \fi%
3804       \noexpand\l@dbfnote{\expandonce\@tag}{\thisc@footnote}%
3805     }%
3806     \to\inserts@listR
3807     \global\advance\insert@countR \@ne%
3808   \else%
3809     \xright@appenditem{%
3810       \ifdefined\Hy@footnote@currentHref%
3811         \noexpand\def\noexpand\Hy@footnote@currentHref{\
Hy@footnote@currentHref}%
3812       \fi%
3813       \noexpand\l@dbfnote{\expandonce\@tag}{\thisc@footnote}%

```

```

3814     }%
3815         \to\inserts@list
3816     \global\advance\insert@count \@ne%
3817     \fi
3818     \ignorespaces%
3819 }%
3820
3821 \newcommand{\get@fnmark}[1]{%
3822     \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3823         l@dprintingcolumns}}}%
3824     {
3825         \stepcounter{footnote@typeset}%
3826         \setcounter{footnote}{\c@footnote@typeset}%
3827         \immediate\write\@mainaux{%
3828             \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
3829         }%
3830     }%
3831     {
3832         \setcounter{footnote}{#1}%
3833         \def\@thefnmark{\thefootnote}%
3834     }%
3835 }%
3836
3837 \newcommand{\vl@dbfnote}[2]{%
3838     \get@fnmark{#2}%
3839     \@footnotetext{#1}%
3840 }%
3841 %

```

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`

```

3842 \newcommand*\prebodyfootmark{%
3843     \leavevmode
3844     \ifhmode
3845         \edef\@x@sf{\the\spacefactor}%
3846         \m@mmf@check
3847         \nobreak
3848     \fi}
3849 \newcommand*\postbodyfootmark{%
3850     \m@mmf@prepare
3851     \ifhmode\spacefactor\@x@sf\fi\relax}
3852
3853 %

```

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.

```

3854 \newcommand{\arrangementX}[2][1,usedefault]{%
3855   \def\do##1{%
3856     \csname arrangementX@#2\endcsname{##1}%
3857   }%
3858   \ifstrempy{#1}%
3859     {%
3860       \dolistloop{\@series}%
3861     }%
3862     {
3863       \docsvlist{#1}%
3864     }%
3865   }%
3866   %

```

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.

```

3867 \newcommand*\normal@footnotemarkX[1]{%
3868   \prebodyfootmark
3869   \wrapped@bodyfootmarkX{#1}%
3870   \postbodyfootmark}
3871
3872 %

```

\normalbodyfootmarkX The `\normalbodyfootmarkX{series}` *really* typesets the in-text marker. The style is the normal superscript.

```

3873 \newcommand*\normalbodyfootmarkX[1]{%
3874   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}%
3875   %

```

\normalvfootnoteX `\normalvfootnoteX{series}{text}` does the `\insert` for the *series* and calls the series' `\footfmt...` to format the *text*.

```

3876 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnoteX}[2]{%
3877   \beforeinsertion@X{#1}%
3878   \insert\@nameuse{footins#1}\bgroup
3879     \reset@font%
3880     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3881     \noindent\csuse{bhooknoteX@#1}%
3882     \csuse{notefontsizeX@#1}%
3883     \footssplit skips

```

```

3884 \ifl@dpairing\ifl@dpaging\else%
3885   \setnoteswidthliketwocolumnsX@{#1}%
3886   \fi\fi%
3887   \setnotesXpositionliketwocolumns@{#1}%
3888   \spaceskip=\z@skip \xspaceskip=\z@skip
3889   \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
3890
3891 %

```

`\mpnormalvfootnoteX` The minipage version.

```

3892 \newcommand*{\mpnormalvfootnoteX}[2]{%
3893   \get@thisfootnoteX{#1}%
3894   \get@fnmarkX{#1}{\thisc@footnote}%
3895   \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3896   \global\setbox\@nameuse{mpfootins#1}\vbox{%
3897     \unvbox\@nameuse{mpfootins#1}
3898     \noindent\csuse{bhooknoteX@#1}%
3899     \csuse{notefontsizeX@#1}%
3900     \hsize\columnwidth
3901     \@parboxrestore
3902     \color@begingroup
3903     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
3904
3905 %

```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

3906 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmtX}[2]{%
3907   \ifluatex%
3908     \texdir\footnote@luatextextdir%
3909     \pardir\footnote@luatexpardir%
3910     \par%
3911   \fi%
3912   \protected@edef\@currentlabel{%
3913     \@nameuse{@thefnmark#1}%
3914   }%
3915   \ledsetnormalparstuffX{#1}%
3916   \hangindent=\csuse{hangindentX@#1}%
3917   \everypar{\hangindent=\csuse{hangindentX@#1}}%
3918   \rule{\z@}{\splittopskip}%
3919   {\csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}}%
3920   \csuse{wrapcontentX@#1}{#2}%
3921   \strut\par}}
3922
3923 %

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

3924 \newcommand*{\normalfootfootmarkX}[1]{%
3925   \textsuperscript{\@nameuse{@thefnmark#1}}}
3926
3927 %

```

\normalfootstartX `\normalfootstartX{⟨series⟩}` is the `⟨series⟩` footnote starting macro used in the output routine.

```

3928 \newcommand*{\normalfootstartX}[1]{%
3929   \ifdimequal{Opt}{\prenotesX@}{}%
3930   {%
3931     \iftoggle{prenotesX@}{%
3932       \togglefalse{prenotesX@}%
3933       \skip\csname footins#1\endcsname=%
3934       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3935     }%
3936   }%
3937 }%
3938 \vskip\skip\csname footins#1\endcsname%
3939 \leftskip=\z@
3940 \rightskip=\z@
3941 \ifl@dpairing\else%
3942   \hsize=\old@hsize%
3943 \fi%
3944 \setnoteswidthliketwocolumnsX@{#1}%
3945 \setnotesXpositionliketwocolumns@{#1}%
3946 \print@footnoteXrule{#1}%
3947 }%
3948
3949 %

```

\normalfootnoteruleX The rule drawn before the footnote series group.

```

3950 \let\normalfootnoteruleX=\footnoterule
3951
3952 %

```

\normalfootgroupX `\normalfootgroupX{⟨series⟩}` sends the contents of the `⟨series⟩` insert box to the output page without alteration.

```

3953 \newcommand*{\normalfootgroupX}[1]{%
3954   \csuse{bhookgroupX@#1}%
3955   \unvbox\@nameuse{footins#1}%
3956   \hsize=\old@hsize%
3957 }%
3958
3959 %

```

\mpnormalfootgroupX The minipage version.

```

3960 \newcommand*{\mpnormalfootgroupX}[1]{%
3961   \vskip\skip\@nameuse{mpfootins#1}
3962   \ifl@dpairing\ifparledgroup%
3963     \leavevmode\marks\parledgroup@{begin}%
3964     \marks\parledgroup@series{#1}%
3965     \marks\parledgroup@type{footnoteX}%
3966   \fi\fi\normalcolor
3967   \ifparledgroup%
3968     \ifl@dpairing%
3969     \else%
3970       \setnoteswidthliketwocolumnsX@{#1}%
3971       \setnotesXpositionliketwocolumns@{#1}%
3972       \print@footnoteXrule{#1}%
3973     \fi%
3974   \else%
3975     \setnoteswidthliketwocolumnsX@{#1}%
3976     \setnotesXpositionliketwocolumns@{#1}%
3977     \print@footnoteXrule{#1}%
3978   \fi%
3979   \csuse{bhookgroupX@#1}%
3980   \unvbox\@nameuse{mpfootins#1}}
3981
3982 %

```

\normalbfnoteX₈₃

```

3984 \newcommand{\normalbfnoteX}[2]{%
3985   \get@thisfootnoteX{#1}%
3986   \ifledRcol%
3987     \ifluatex
3988       \footnotelang@lua[R]%
3989     \fi
3990     \@ifundefined{xpg@main@language}%if polyglossia
3991     {}%
3992     {\footnotelang@poly[R]}%
3993   \xright@appenditem{%
3994     \noexpand\led@set@index@fornote{#1}%
3995     \noexpand\prepare@edindex@fornote{\led@nums}%
3996     \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3997     \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
3998     \noexpand\led@reinit@index@fornote%
3999     \unexpanded{\advance\@edindex@fornote@m@ne}%
4000   }%
4001   \to\inserts@listR
4002   \global\advance\insert@countR \@ne%
4003 \else%
4004   \ifluatex
4005     \footnotelang@lua%
4006   \fi

```



```

4007 \ifundefined{xpg@main@language}%if polyglossia
4008 {}%
4009 {\footnotelang@poly}%
4010 \xright@appenditem{%
4011 \noexpand\led@set@index@fornote{#1}%
4012 \noexpand\prepare@edindex@fornote{\led@nums}%
4013 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
4014 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
4015 \noexpand\led@reinit@index@fornote%
4016 \unexpanded{\advance\@edindex@fornote@\m@ne}%
4017 }%
4018 \to\inserts@list
4019 \global\advance\insert@count \@ne%
4020 \fi
4021 \ignorespaces}
4022 %
4023 %

```

\get@thisfootnoteX The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

4024 \newcommand{\get@thisfootnoteX}[1]{%
4025 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4026 \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
endcsname}%
4027 }{%
4028 \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}%
4029 }%
4030 }%
4031 %

```

\vbfnoteX This command calls the correct footnote-inserting commands.

```

4032 \newcommand{\vbfnoteX}[3]{%
4033 \get@fnmarkX{#1}{#3}%
4034 \@nameuse{regvfootnote#1}{#1}{#2}%
4035 }%
4036 %
4037 %

```

\get@fnmarkX This command gets the correct footnote number when typesetting parallel texts.

```

4038 \newcommand{\get@fnmarkX}[2]{%
4039 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
4040 {%
4041 \stepcounter{footnote#1@typeset}%
4042 \setcounter{footnote#1}{\value{footnote#1@typeset}}}%

```

```

4043 \namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4044 \immediate\write\@mainaux{%
4045 \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
@typeset\endcsname}%
4046 }%
4047 }%
4048 {%
4049 \setcounter{footnote#1}{#2}%
4050 \namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
4051 }%
4052 }
4053 %
4054 %

```

```

\newcommand{\vnumfootnoteX}[2]{%
4056 \ifnumberedpar@
4057 \edtext{}{\normalbfnoteX{#1}{#2}}%
4058 \else
4059 \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4060 \get@thisfootnoteX{#1}%
4061 \get@fnmarkX{#1}{\expandonce\thisc@footnote}%
4062 \@nameuse{regvfootnote#1}{#1}{#2}%
4063 \fi}
4064
4065 %

```

arrangementX@normal `\arrangementX@normal{<series>}` initialises the settings for the <series> footnotes. This should always be called for each series.

```

4066 \newcommand*{\arrangementX@normal}[1]{%
4067 \csgdef{series@displayX#1}{normal}
4068 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
4069 \namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
4070 \namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
4071 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
4072 \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
4073 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
4074 \namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
4075 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
4076 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4077 \count\csname footins#1\endcsname=1000
4078 \csxdef{default@footins#1}{1000}%Use to have note only for one side
4079 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4080 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4081 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4082 %

```

Additions for minipages.

```

4083 \ifnoledgroup@else%
4084 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4085 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
4086 \count\csname mpfootins#1\endcsname=1000
4087 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}%
4088 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4089 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4090 \fi
4091 }
4092
4093 %

```

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 4094 \newcommand*\arrangementX@twocol}[1]{%
4095 \csgdef{series@displayX#1}{twocol}
4096 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4097 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4098 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4099 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4100 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4101 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4102 \twocolfootsetupX{#1}
4103 \ifnoledgroup@else%
4104 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4105 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4106 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4107 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4108 \mptwocolfootsetupX{#1}
4109 \fi%
4110 }
4111
4112 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
4113 \newcommand*\twocolfootsetupX}[1]{%
4114 \count\csname footins#1\endcsname 500
4115 \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
4116 \multiply\dimen\csname footins#1\endcsname by \tw@}
4117 \newcommand*\mptwocolfootsetupX}[1]{%
4118 \count\csname mpfootins#1\endcsname 500
4119 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
4120
4121 %

```

`\twocolvfootnoteX` `\twocolvfootnoteX{<series>}`

```

4122 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
4123   \beforeinsertion@X{#1}%
4124   \insert\csname footins#1\endcsname\bgroup%
4125   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4126   \noindent\csuse{bhooknoteX@#1}%
4127   \csuse{notefontsizeX@#1}%
4128   \footssplitskips%
4129   \spaceskip=\z@skip \xspaceskip=\z@skip%
4130   \@nameuse{footfmt#1}{#1}{#2}\egroup}
4131
4132 %

```

`\twocolfootfmtX` `\twocolfootfmtX{<series>}`

```

4133 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4134   \protected@edef\@currentlabel{%
4135     \@nameuse{@thefnmark#1}%
4136   }%
4137   \normal@pars%
4138   \hangindent=\csuse{hangindentX@#1}%
4139   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4140   \hsize \csuse{hsizetwocolX@#1}%
4141   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4142   \tolerance=5000\relax%
4143   \par%
4144   \@tempdima=\parindent%
4145   \csuse{colalignX@#1}%
4146   \parindent=\@tempdima%
4147   {\hspace{\parindent}%
4148    \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4149    \csuse{wrapcontentX@#1}{#2}%
4150    \strut\par}%
4151   \allowbreak%
4152 }%
4153
4154 %

```

`\twocolfootgroupX` `\twocolfootgroupX{<series>}`
`\mptwocolfootgroupX`

```

4155 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
4156   notefontsizeX@#1}
4157   \splittopskip=\ht\strutbox
4158   \expandafter
4159   \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4160 \newcommand*{\mptwocolfootgroupX}[1]{%
4161   \vskip\skip\@nameuse{mpfootins#1}
4162   \ifl@dpairing\ifparledgroup%

```

```

4163 \leavevmode\marks\parledgroup@{begin}%
4164 \marks\parledgroup@series{#1}%
4165 \marks\parledgroup@type{footnoteX}%
4166 \fi\fi\normalcolor
4167 \ifparledgroup%
4168 \ifl@dpairing%
4169 \else%
4170 \setnoteswidthliketwocolumnsX@{#1}%
4171 \setnotesXpositionliketwocolumns@{#1}%
4172 \print@footnoteXrule{#1}%
4173 \fi%
4174 \else%
4175 \setnoteswidthliketwocolumnsX@{#1}%
4176 \setnotesXpositionliketwocolumns@{#1}%
4177 \print@footnoteXrule{#1}%
4178 \fi%
4179 \csuse{bhookgroupX@#1}%
4180 \splittopskip=\ht\strutbox
4181 \expandafter
4182 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4183
4184 %

```

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 85 \newcommand*{\arrangementX@threecol}[1]{%
4186 \csgdef{series@displayX#1}{threecol}
4187 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4188 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4189 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4190 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4191 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4192 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4193 \threecolfootsetupX{#1}
4194 \ifnoledgroup@ \else%
4195 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4196 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4197 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4198 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4199 \mpthreecolfootsetupX{#1}
4200 \fi%
4201 }
4202
4203 %

```

```

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX

```

```

4204 \newcommand*{\threecolfootsetupX}[1]{%
4205   \count\csname footins#1\endcsname 333
4206   \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4207   \multiply\dimen\csname footins#1\endcsname by \thr@@}
4208 \newcommand*{\mpthreecolfootsetupX}[1]{%
4209   \count\csname mpfootins#1\endcsname 333
4210   \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
4211
4212 %

```

`\threecolvfootnoteX` `\threecolvfootnoteX{<series>}{<text>}`

```

4213 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{%
%
4214   \beforeinsertion@X{#1}%
4215   \insert\csname footins#1\endcsname\bgroup%
4216     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4217     \noindent\csuse{bhooknoteX@#1}%
4218     \csuse{notefontsizeX@#1}%
4219     \footsplitskips%
4220     \@nameuse{footfmt#1}{#1}{#2}\egroup}
4221
4222 %

```

`\threecolfootfmtX` `\threecolfootfmtX{<series>}`

```

4223 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4224   \protected@edef\@currentlabel{%
4225     \@nameuse{@thefnmark#1}%
4226   }%
4227   \hangindent=\csuse{hangindentX@#1}%
4228   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4229   \normal@pars%
4230   \hsize \csuse{hsizethreecolX@#1}%
4231   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4232   \tolerance=5000\relax%
4233   \@tempdima=\parindent%
4234   \csuse{colalignX@#1}%
4235   \parindent=\@tempdima%
4236   {\hspace{\parindent}%
4237     \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4238     \csuse{wrapcontentX@#1}{#2}%
4239     \strut\par}\allowbreak}
4240
4241 %

```

`\threecolfootgroupX` `\threecolfootgroupX{<series>}`
`\mpthreecolfootgroupX`

```

4242 \newcommand*{\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4243 \splittopskip=\ht\strutbox
4244 \expandafter
4245 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4246
4247 \newcommand*{\mpthreecolfootgroupX}[1]{\%
4248 \vskip\skip\@nameuse{mpfootins#1}
4249 \ifl@dpairing\ifparledgroup
4250 \leavevmode\marks\parledgroup@{begin}%
4251 \marks\parledgroup@series{#1}%
4252 \marks\parledgroup@type{footnoteX}%
4253 \fi\fi\normalcolor
4254 \ifparledgroup%
4255 \ifl@dpairing%
4256 \else%
4257 \setnoteswidthliketwocolumnsX@{#1}%
4258 \setnotesXpositionliketwocolumns@{#1}%
4259 \print@footnoteXrule{#1}%
4260 \fi%
4261 \else%
4262 \setnoteswidthliketwocolumnsX@{#1}%
4263 \setnotesXpositionliketwocolumns@{#1}%
4264 \print@footnoteXrule{#1}%
4265 \fi%
4266 \csuse{bhookgroupX@#1}%
4267 \splittopskip=\ht\strutbox
4268 \expandafter
4269 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4270
4271 \%

```

XIII.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@threecol` `\footparagraphX{<series>}`

```

4272 \newcommand*{\arrangementX@paragraph}[1]{\%
4273 \csgdef{series@displayX#1}{paragraph}%
4274 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4275 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4276 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4277 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4278 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4279 \count\csname footins#1\endcsname=1000
4280 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
4281 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4282 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%

```

```

4283 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4284 \para@footsetupX{#1}
4285 \ifnoledgroup\else
4286   \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4287   \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4288   \count\csname mpfootins#1\endcsname=1000
4289   \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4290   \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4291   \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4292 \fi
4293 }
4294
4295 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

4296 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4297   \setnoteswidthliketwocolumnsX@{#1}%
4298   \ifcsempy{widthX@#1}%
4299     {}%
4300     {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4301   \dimen0=\baselineskip
4302   \multiply\dimen0 by 1024
4303   \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
4304   %
4305   \expandafter
4306   \xdef\csname footfudgefactor#1\endcsname{%
4307     \expandafter\strip@pt\dimen0 }}
4308 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

4309 \newcommand*{\parafootstartX}[1]{%
4310   \ifdimequal{Opt}{\prenotesX@}{}%
4311   {%
4312     \iftoggle{prenotesX@}{%
4313       \togglefalse{prenotesX@}%
4314       \skip\csname footins#1\endcsname=%
4315       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4316     }%
4317   }%
4318 }%
4319 \leftskip=\z@
4320 \rightskip=\z@
4321 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4322 \vskip\skip@nameuse{footins#1}%
4323 \setnoteswidthliketwocolumnsX@{#1}%
4324 \setnotesXpositionliketwocolumns@{#1}%

```



```

4325 \print@footnoteXrule{#1}%
4326 }
4327
4328 %

```

`\para@vfootnoteX` `\para@vfootnoteX{<series>}{<text>}`
`\mppara@vfootnoteX`

```

4329 \newcommand*{\para@vfootnoteX}[2]{%
4330 \csuse{beforeinsertingX@#1}%
4331 \insert\csname footins#1\endcsname%
4332 \bgroup
4333 \csuse{notefontsizeX@#1}
4334 \footsplitskips
4335 \setbox0=\vbox{\hsize=\maxdimen%
4336 \let\bidir@RTL@everypar\@empty%
4337 \insert@txtbeforenotesX{#1}%
4338 \noindent\csuse{bhooknoteX@#1}%
4339 \@nameuse{footfmt#1}{#1}{#2}}%
4340 \setbox0=\hbox{\unvvhX{0}{#1}}%
4341 \dp0=\z@
4342 \ht0=\csname footfudgefactor#1\endcsname\wd0
4343 \box0
4344 \penalty0
4345 \egroup}
4346 \newcommand*{\mppara@vfootnoteX}[2]{%
4347 \get@thisfootnoteX{#1}%
4348 \get@fnmarkX{#1}{\thisc@footnote}%
4349 \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4350 \global\setbox\@nameuse{mpfootins#1}\vbox{%
4351 \unvvhX\@nameuse{mpfootins#1}
4352 \csuse{notefontsizeX@#1}
4353 \footsplitskips
4354 \setbox0=\vbox{\hsize=\maxdimen%
4355 \let\bidir@RTL@everypar\@empty%
4356 \noindent\color@begingroup%
4357 \csuse{bhooknoteX@#1}%
4358 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
4359 \setbox0=\hbox{\unvvhX{0}{#1}}%
4360 \dp0=\z@
4361 \ht0=\csname footfudgefactor#1\endcsname\wd0
4362 \box0
4363 \penalty0}}
4364
4365 %

```

`\unvvhX`₆₆ `\newcommand*{\unvvhX}[2]{% 2th is optional for retro-compatibility`
`\setbox0=\vbox{\unvvhX#1%`
`\global\setbox1=\lastbox}%`
`\unhbox1`

```

4370 \unskip          % remove \rightskip,
4371 \unskip          % remove \parfillskip,
4372 \unpenalty       % remove \penalty of 10000,
4373 \hskip\csuse{afternoteX@#2}%
4374 \relax}% but add the glue to go between the notes
4375
4376 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

4377 \newcommand*{\parafootfmtX}[2]{%
4378   \protected@edef\@currentlabel{%
4379     \@nameuse{@thefnmark#1}%
4380   }%
4381   \insertparafootsepX{#1}%
4382   \ledsetnormalparstuff@common%
4383   {\csuse{notenumfontX@#1}%
4384    \csuse{notenumfontX@#1}%
4385    \wrapped@footfootmarkX{#1}%
4386    \strut%
4387    \csuse{wrapcontentX@#1}{#2}%
4388    \penalty-10}}
4389
4390 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`

`\mppara@footgroupX`

```

4391 \newcommand*{\para@footgroupX}[1]{%
4392   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4393   \unvbox\csname footins#1\endcsname
4394   \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4395   \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4396   \makehboxofhboxes
4397   \setbox0=\hbox{\unhbox0 \removehboxes}%
4398   \csuse{bhookgroupX@#1}
4399   \csuse{notefontsizeX@#1}
4400   \unhbox0\par}
4401
4402 \newcommand*{\mppara@footgroupX}[1]{%
4403   \setnoteswidthliketwocolumnsX@{#1}%
4404   \vskip\skip\@nameuse{mpfootins#1}
4405   \ifl@dpairing\ifparledgroup
4406     \leavevmode%
4407     \leavevmode\marks\parledgroup@{begin}%
4408     \marks\parledgroup@series{#1}%
4409     \marks\parledgroup@type{footnoteX}%
4410     \fi\fi\normalcolor
4411   \ifparledgroup%
4412     \ifl@dpairing%
4413     \else%

```

```

4414 \setnoteswidthliketwocolumnsX@{#1}%
4415 \setnotesXpositionliketwocolumns@{#1}%
4416 \print@footnoteXrule{#1}%
4417 \fi%
4418 \else%
4419 \setnoteswidthliketwocolumnsX@{#1}%
4420 \setnotesXpositionliketwocolumns@{#1}%
4421 \print@footnoteXrule{#1}%
4422 \fi%
4423 \unvbox\csname mpfootins#1\endcsname
4424 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4425 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4426 \makehboxofhboxes
4427 \setbox0=\hbox{\unhbox0 \removehboxes}%
4428 \csuse{bhookgroupX@#1}%
4429 \csuse{notefontsizeX@#1}%
4430 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4431 \unhbox0\par}}
4432 %
4433 %

```

Insertion of the footnotes separator The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX34 \newcommand{\insertparafootsepX}[1]{%
4435 \ifledRcol@%
4436 \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
4437 {\csuse{Xparafootsep@#1}}%
4438 {}%
4439 \global\csname prevpage#1@numR\endcsname=\page@numR%
4440 \else%
4441 \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
4442 {\csuse{Xparafootsep@#1}}%
4443 {}%
4444 \global\csname prevpage#1@num\endcsname=\page@num%
4445 \fi%
4446 }
4447 %

```

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.

```

4448 \newcommand{\wrapped@footfootmarkX}[1]{%
4449 \ifdefined\hypertarget%
4450 \hyperlink%
4451 {@bodyfootmark#1@\this@footnoteX@reading}%

```

```

4452     {\@nameuse{footfootmark#1}}}%
4453     \Hy@raisedlink{%
4454       \hypertarget%
4455         {@footnotemark#1@\this@footnoteX@reading}%
4456       {}%
4457     }%
4458   \else%
4459     \@nameuse{footfootmark#1}%
4460   \fi%
4461 }%
4462 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package’s commands, if needed.

```

4463 \newcommand{\wrapped@bodyfootmarkX}[1]{%
4464   \ifdefined\hypertarget%
4465     \hyperlink%
4466       {@footnotemark#1@\expandafter\the\csname footnote#1@reading\
endcsname}%
4467     {\@nameuse{bodyfootmark#1}}}%
4468   \Hy@raisedlink{%
4469     \hypertarget%
4470       {@bodyfootmark#1@\expandafter\the\csname footnote#1@reading\
endcsname}%
4471     {}%
4472   }%
4473   \else%
4474     \@nameuse{bodyfootmark#1}%
4475   \fi%
4476 }%
4477 %

```

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@common78 \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff79 \ifluatex%
\ledsetnormalparstuffX

```

```

4480 \ifdefstring{\footnote@luatextextdir}{TLT}{}%
4481 {\textdir\footnote@luatextextdir}%
4482 \pardir\footnote@luatexpardir%
4483 \fi%
4484 \csuse{\csuse{footnote@dir}}%
4485 \normal@pars%
4486 \parfillskip \z@ \@plus 1fil}%
4487
4488 \newcommand*{\Xledsetnormalparstuff}[1]{%
4489 \ledsetnormalparstuff@common%
4490 \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
4491 }%
4492
4493 \newcommand*{\ledsetnormalparstuffX}[1]{%
4494 \ledsetnormalparstuff@common%
4495 \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}%
4496 }%
4497 %

```

XV Footnotes' width for two columns

We define here some commands which make sense only with `reledpar`, but must be called when defining notes parameters. These commands change the width of block notes to allow them to have the same size than two parallel columns.

`\old@hsize` These two commands are called at the beginning of critical or familiar notes groups. They set, if the option is enabled, the `\hsize`. They are also called at the on the setup for paragraphed notes.

`\noteswidthliketwocolumns@`
`\oteswidthliketwocolumnsX@`

```

4498
4499 \newdimen\old@hsize%
4500 \AtBeginDocument{\old@hsize=\hsize}%
4501
4502 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
4503 \global\let\hsize@fornote=\hsize%
4504 \global\old@hsize=\hsize%
4505 \let\old@columnwidth=\columnwidth%
4506 \iftoggle{Xnoteswidthliketwocolumns@#1}%
4507 {%
4508 \csuse{setwidthliketwocolumns@\columns@position}%
4509 \global\let\hsize@fornote=\hsize%
4510 }%
4511 {}%
4512 \let\hsize=\hsize@fornote%
4513 \let\columnwidth=\old@columnwidth%
4514 }%
4515
4516 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4517 \global\let\hsize@fornote=\hsize%

```

```

4518 \global\old@hsize=\hsize%
4519 \let\old@columnwidth=\columnwidth%
4520 \iftoggle{noteswidthliketwocolumnsX@#1}{%
4521   {%
4522     \csuse{setwidthliketwocolumns@\columns@position}%
4523     \global\let\hsize@fornote=\hsize%
4524   }%
4525   {%
4526     \let\hsize=\hsize@fornote%
4527     \let\columnwidth=\old@columnwidth%
4528   }%
4529 }%
4530 %

```

`\setXnotespositionliketwocolumns@`
`\setnotesXpositionliketwocolumns@`

These two commands set the position of the critical / familiar footnotes, depending on the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They call commands which are defined only in `reledpar`, because this feature has no sens without `reledpar`.

```

4531 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
4532   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4533     \csuse{setnotespositionliketwocolumns@\columns@position}%
4534   }{%
4535   }%
4536 }%
4537 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
4538   \iftoggle{noteswidthliketwocolumnsX@#1}{%
4539     \csuse{setnotespositionliketwocolumns@\columns@position}%
4540   }{%
4541   }%
4542 }%
4543 %

```

XVI Footnotes' order

`\fnpos` The `\fnpos` and `\mpfnpos` simply place their arguments in `\@fnpos` and `\@mpfnpos`,
`\mpfnpos` which will be used later in the output routine.

```

4544 \def\@fnpos{familiar-critical}
4545 \def\@mpfnpos{critical-familiar}
4546 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
4547 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
4548 %

```

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@Xfootnoterule49 \newcommand{\print@Xfootnoterule}[1]{%
\print@footnoteXrule50 \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
4551 \nointerlineskip%
4552 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
4553 \nointerlineskip%
4554 \vskip\csuse{Xafterrule@#1}%
4555 }%
4556
4557 \newcommand{\print@footnoteXrule}[1]{%
4558 \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
4559 \nointerlineskip%
4560 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
4561 \nointerlineskip%
4562 \vskip\csuse{afterruleX@#1}%
4563 }%
4564
4565 %

```

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.

```

4566 \newtoggle{Xprenotes@}%
4567 \toggletrue{Xprenotes@}%
4568 \newcommand{\Xprenotes@}{Opt}%
4569 \newcommand*{\Xprenotes}[1]{\renewcommand{\Xprenotes@}{#1}}%
4570 \newcommand{\preXnotes}[1]{\led@warning@preXnotes@deprecated\Xprenotes{#1}}
    %For compatibility
4571 %

```

The same, but for familiar footnotes.

```

\Xprenotes 72 \newtoggle{prenotesX@}
\Xprenotes@ 73 \toggletrue{prenotesX@}
4574 \newcommand{\prenotesX@}{Opt}
4575 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
4576 %

```

XVIII.0.3 Internal commands

```

firstXseries@ 77 \gdef\firstXseries@{}
prepare@Xprenotes 78 \newcommand{\prepare@Xprenotes}[1]{%
4579   \ifdimequal{Opt}{\Xprenotes@}%
4580   }%
4581   {%
4582     \IfStrEq{\firstXseries@}{}{%
4583       \global\skip\csuse{#1footins}=\Xprenotes@%
4584       \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4585       \gdef\firstXseries@{#1}%
4586     }%
4587     {%
4588       \ifseriesbefore{#1}{\firstXseries@}%
4589       {%
4590         \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
4591         \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4592         \gdef\firstXseries@{#1}%
4593       }%
4594     }%

```



```

4595 }%
4596 }%
4597 }
4598 %

```

The same thing is required for familiar notes and `\prenotesX`.

```

firstseriesX@ \gdef\firstseriesX@{}
prepare@prenotesX@ \newcommand{\prepare@prenotesX}[1]{%
4601 \ifdimequal{0pt}{\prenotesX@}%
4602 {}%
4603 {%
4604 \IfStrEq{\firstseriesX@}{-}{%
4605 \global\skip\csuse{footins#1}=\prenotesX@%
4606 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4607 \gdef\firstseriesX@{#1}%
4608 }%
4609 {%
4610 \ifseriesbefore{#1}{\firstseriesX@}%
4611 {%
4612 \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
4613 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4614 \gdef\firstXseries@{#1}%
4615 }%
4616 {}%
4617 }%
4618 }%
4619 }
4620 %

```

XIX Endnotes

First, check the `noend` option.

```

4621 \ifbool{noend@}{-}{%Used instead of \ifnoend@ to prevent expansion problem
4622 %

```

XIX.1 Internal commands

`\l@dend@open` and `\l@dend@close` are the macros that are used to open and close the endnote file. Note that all our writing to this file is `\immediate`: all page and line numbers for the endnotes are generated by the same mechanism we use for the footnotes, so that there is no need to defer any writing to catch information from the output routine. The argument of these two command is the series letter.

```

4623 \newcommand{\l@dend@open}[1]{%
4624   \global\booltrue{l@dend@#1}%
4625   \expandafter\immediate%
4626   \expandafter\openout%
4627   \csname l@d@#1end\endcsname%
4628   =\l@auxdir\jobname.#1end\relax%
4629 }%
4630 \newcommand{\l@dend@close}[1]{%
4631   \global\boolfalse{l@dend@#1}%
4632   \expandafter\immediate%
4633   \expandafter\closeout\csname l@d@#1end\endcsname%
4634 }%
4635 %
4636 %

```

\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.

```

4637 \newcommand{\l@dend@stuff}{%
4638   \def\do##1{%
4639     \ifbool{l@dend@##1}{}%
4640     {\l@dend@open{##1}}%
4641     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
4642   }%
4643   \dolistloop{@series}%
4644 }%
4645 %
4646 %

```

\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.

```

4647 \global\newbool{parapparatus@}{\long}\def\endprint#1#2#3#4#5#6#7{
4648   \csuse{Xendbhooknote@#4}%
4649   \csuse{Xendnotefontsize@#4}%
4650   \hangindent=\csuse{Xendhangindent@#4}%
4651   \ifXendinsertsep%
4652     \hskip\csuse{Xendafternote@#4}\relax%
4653     \csuse{Xendsep@#4}%
4654   \else%
4655     \iftoggle{Xendparagraph@#4}%
4656       {\global\Xendinsertsep@true}%
4657     {}%
4658   \fi%
4659   \xdef\@currentseries{#4}%
4660   \def\do##1{%
4661     \setkeys[mac]{truefootnoteoption}{##1}%
4662   }%
4663   \notblank{#5}{\docsvlist{#5}}{}%
4664   \IfStrEq{#6}{R}{\ledRcol@true}{}%
4665   \def\@this@crossref@start{#7:start}%
4666   \def\@this@crossref@end{#7:end}%
4667   \printlineendnote{#1}{#4}%
4668   \IfStrEq{#6}{R}{\ledRcol@false}{}%
4669   \undef\@this@crossref@start%
4670   \undef\@this@crossref@end%
4671   \nottoggle{Xendlemmadisablefontselection@#4}%
4672   {\select@lemmafont#1}%
4673   {}%
4674   \bgroup%
4675     \csuse{Xendlemmafont@#4}%
4676     \csuse{Xendwraplemma@#4}{#2}%
4677   \egroup%
4678   \ifbool{expr}%
4679     togl {nosep@}%
4680     or test{\ifcempty{Xendlemmaseparator@#4}}%
4681   }%
4682   {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
4683   {\nobreak%
4684     \hskip\csuse{Xendbeforelemmaseparator@#4}%
4685     \csuse{Xendlemmaseparator@#4}%
4686     \hskip\csuse{Xendafterlemmaseparator@#4}%
4687     \relax%
4688   }%
4689   \csuse{Xendwrapcontent@#4}{#3}%
4690   \nottoggle{Xendparagraph@#4}{\par}{}%
4691   \def\do##1{%
4692     \setkeys[mac]{falsefootnoteoption}{##1}%
4693   }%
4694   \notblank{#5}{\docsvlist{#5}}{}%
4695 }%
4696

```

```

4697 \let\l@d@section=\@gobble
4698
4699 %

```

\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.

```

4700 \newcommand{\printlineendnote}[2]{%
4701   \l@dp@rsefootspec#1|{%
4702     \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
4703       \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
4704         \l@dparsedstartsub - \l@dparsedendpage - \l@dparsedendline - \
4705         \l@dparsedendsub}%
4706       }%
4707       {%
4708         \xdef\lineinfo@{\l@dparsedstartpage - \l@dparsedstartline - \
4709         \l@dparsedstartsub}%
4710       }%
4711       \ifboolexpr{%
4712         togl {nonum@}%
4713         or togl {Xendnonumber@#2}}%
4714       {%
4715         \iftoggle{Xendnumberonlyfirstinline@#2}%
4716         {\ifcsequal{prevendline#2}%
4717           {\ifcsequal{prevendline#2}{\lineinfo@}%
4718             {%
4719               \csuse{Xendbhookinplaceofnumber@#2}%
4720               \ifcsequal{Xendsymnum@#2}%
4721                 {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4722                 {\printlineendnotearea{#2}}}%
4723               \csuse{Xendahookinplaceofnumber@2}%
4724             }%
4725             {\printlineendnotearea{#1}{#2}}}%
4726           }%
4727           {\printlineendnotearea{#1}{#2}}}%We keep every time line
4728         \csxdef{prevendline#2}{\lineinfo@}%
4729       }%
4730     }%
4731   }%

```

```

\printsymlineendnotearea 4732 \newcommand{\printsymlineendnotearea}[1]{%
4733   \hspace{\csuse{Xendbeforenum@#1}}%
4734   \csuse{Xendnotenumfont@#1}%
4735   \ifdimequal{\csuse{Xendboxsymnum@#1}}{\z@}%
4736   {\csuse{Xendsymnum@#1}}%

```

```

4737     {\hbox to \csuse{Xendboxsymlinenum@#1}%
4738      {\csuse{Xendsymlinenum@#1}\hfill}%
4739     }%
4740     \hspace{\csuse{Xendaftersymlinenum@#1}}%
4741 }%
4742 %

```

\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.)

```

4743 \newcommand{\printlineendnotearea}[2]{%
4744   \csuse{Xendbhooklinenumber@#2}%
4745   \hspace{\csuse{Xendbeforenumber@#2}}%
4746   \bgroup%
4747   \csuse{Xendnotenumfont@#2}%
4748   \ifdimequal{\csuse{Xendboxlinenum@#2}}{0pt}%
4749     {\printendlines#1||\ifledRcol@\@Rlineflag\fi}%
4750     {\leavevmode%
4751      \hbox to \csuse{Xendboxlinenum@#2}%
4752      {%
4753       \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4754       \printendlines#1||\ifledRcol@\@Rlineflag\fi%
4755       \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
4756      }%
4757   \egroup%
4758   \hspace{\csuse{Xendafternumber@#2}}%
4759   \csuse{Xendahooklinenumber@#2}%
4760 }%
4761 %

```

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.

```

\toendnotes62 \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes*63 \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4764   \def\do##1{%
4765     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4766     {\unexpanded{#2}\@percentchar}%
4767   }%
4768   \ifstrempy{#1}%
4769     {\dolistloop{\@series}}%

```

```

4770     {\docsvlist{#1}}%
4771   }{\led@err@toendnotes@outsidenumbering}%
4772 }%
4773 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
4774   \ifbool{expr{bool{numbering} or bool{numberingR}}}{%
4775     \def\do##1{%
4776       \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4777       {#2\@percentchar}%
4778     }%
4779     \ifstrempy{#1}%
4780     {\dolistloop{\@series}}%
4781     {\docsvlist{#1}}%
4782   }{\led@err@toendnotes@outsidenumbering}%
4783 }%
4784 %

```

XIX.2.2 Printing endnotes

\doendnotes \doendnotes is the command you use to print one series of endnotes; it takes one argument: the series letter of the note series you want to print. **\Xendinsertsep@** is set to true at the first note of the series, and to false at the last one.

```

4785 \newif\ifXendinsertsep@%
4786 \newcommand*{\doendnotes}[1]{%
4787   \l@dend@close{#1}%
4788   \begingroup
4789     \csxdef{prevpagenum@#1}{}%
4790     \csxdef{prevpagerange@#1}{}%
4791     \makeatletter
4792     \expandafter\let\csname #1end\endcsname=\endprint
4793     \input\l@auxdir\jobname.#1end%
4794     \global\Xendinsertsep@false%
4795   \endgroup}
4796 %

```

\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.

```

4797 \newcommand*{\doendnotesbysection}[1]{%
4798   \l@dend@close{#1}%
4799   \csxdef{prevpagenum@#1}{}%
4800   \csxdef{prevpagerange@#1}{}%
4801   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
4802   \begingroup%
4803     \makeatletter%
4804     \def\l@d@section##1{%

```

```

4805 \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
4806 {\cslet{#1end}{\endprint}}%
4807 {\cslet{#1end}{\@gobbleseven}}%
4808 }%
4809 \input\l@auxdir\jobname.#1end%
4810 \global\Xendinsertsep@false%
4811 \endgroup%
4812 }%
4813 %

```

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.

```

4814 }%
4815 %

```

`\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.

```

4816 \newcommand*{\setprintendlines}[6]{%
4817 \l@d@pnumfalse \l@d@dashfalse
4818 \ifnum#4=#1 \else
4819 \l@d@pnumtrue
4820 \l@d@dashtrue
4821 \fi
4822 %

```

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.

```

4823 \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
4824 \ifnum#2=#5 \else
4825 \l@d@elintrue
4826 \l@d@dashtrue
4827 \fi
4828 %

```

We print the starting sub-line if it is nonzero.

```

4829 \l@d@ssubfalse
4830 \ifnum#3=0 \else
4831 \l@d@ssubtrue
4832 \fi
4833 %

```

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.

```

4834 \l@d@eslfalse
4835 \ifnum#6=0 \else
4836   \ifnum#6=#3
4837     \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4838   \else
4839     \l@d@esltrue
4840     \l@d@dashtrue
4841   \fi
4842 \fi%
4843 %

4844 \ifl@d@dash%
4845   \ifbool{expr{togl{fulllines@} or test{\ifcempty{Xendtwolines@}
@currentseries}}}%
4846   {%
4847   {%
4848     \setistwofollowinglines{#1}{#2}{#4}{#5}%
4849     \ifbool{expr{
4850       (%
4851         togl {Xendtwolinesbutnotmore@}\@currentseries}%
4852         and not%
4853         (%
4854           bool {istwofollowinglines@}%
4855         )%
4856       )%
4857       or%
4858       (%
4859         (not test{\ifnumequal{#1}{#4}})%
4860         and togl{Xendtwolinesonlyinsamepage@}\@currentseries}%
4861       )%
4862     }%
4863   }%
4864   {%
4865     \l@d@dashfalse%
4866     \l@d@Xtwolinesttrue%
4867     \l@d@elinfalse%
4868     \l@d@eslfalse%
4869     \ifcempty{Xendmoreethantwolines@}\@currentseries}%
4870     {%
4871       \ifistwofollowinglines@ \else%
4872         \l@d@Xmoreethantwolinesttrue%
4873       \fi%
4874     }%
4875   }%
4876 }%
4877 \fi%
4878 %

```


End of `\setprintendlines`.

```
4879 }%
4880 %
```

`\printendlines` Now we are ready to print it all.

```
4881 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4882   \begingroup
4883   \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
4884   %
```

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.

```
4885   \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
4886     {\bgroup}%
4887     {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}}%
4888   %
```

Then, print the starting page number-

```
4889   \ifboolexpr{%
4890     (%
4891       test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
4892       and not%
4893       (togl{Xendpagenumberonlyfirstifsingle@\@currentseries} and bool{
1@d@pnum})}%
4894     )%
4895     or%
4896     (%
4897       test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
4898     )%
4899   }%
4900   {%
4901     \ifcsempy{Xendsympagenum@\@currentseries}%
4902       {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
4903       {\csuse{Xendsympagenum@\@currentseries}}%
4904     }%
4905     {%
4906       \wrap@edcrossref{\@this@crossref@start}{\printnpnum{#1}}%
4907     }%
4908   %
```

Then, determine what must be printed before the start line.

```
4909   \ifl@d@dash%
4910     \ifl@d@pnum%
```

```

4911 \csuse{Xendlineprefixsingle@\@currentseries}%
4912 \else%
4913 \ifcempty{Xendlineprefixmore@\@currentseries}%
4914 {\csuse{Xendlineprefixsingle@\@currentseries}}%
4915 {\csuse{Xendlineprefixmore@\@currentseries}}}%
4916 \fi%
4917 \else%
4918 \csuse{Xendlineprefixsingle@\@currentseries}%
4919 \fi%
4920 %

```

The print the starting line, followed, if needed, by the side flag and the starting sub line number.

```

4921 \wrap@edcrossref{\@this@crossref@start}{}%
4922 \ifledRcol@%
4923 \linenumrepR{#2}%
4924 \else%
4925 \linenumrep{#2}%
4926 \fi%
4927 }%
4928 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
4929 \ifl@d@ssub%
4930 \csuse{Xendsublinesep@\@currentseries}%
4931 \wrap@edcrossref{\@this@crossref@start}{}%
4932 \ifledRcol@%
4933 \sublinenumrepR{#3}%
4934 \else%
4935 \sublinenumrep{#3}%
4936 \fi%
4937 }%
4938 \fi%
4939 %

```

Close the box.

```

4940 \egroup%
4941 %

```

Open the box for the ending line number.

```

4942 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
4943 {\bgroup}%
4944 {\hbox to \csuse{Xendboxendlinenum@\@currentseries}\bgroup}%
4945 %

```

Print the dash + the ending line number, or the line number range symbol.

```

4946 \ifl@d@xtwolines%
4947 \ifl@d@xmorethantwolines%
4948 \csuse{Xendmorethantwolines@\@currentseries}%
4949 \else%
4950 \csuse{Xendt看olines@\@currentseries}%

```

```

4951 \fi%
4952 \else%
4953 \ifl@d@dash%
4954 \ifdefined\linangesep@%
4955 \linangesep@%
4956 \else%
4957 \csuse{Xendlinangeseparator@}\@currentseries}%
4958 \fi%
4959 \fi%
4960 %

```

Print the ending page number.

```

4961 \ifl@d@pnum%
4962 \ifcsstring{prevpagerange@\@currentseries}{#1-#4}%
4963 {%
4964 \ifcsempy{Xendsympagenum@\@currentseries}%
4965 {\hspace{\csuse{Xendingplaceofpagenumber@\@currentseries}}}%
4966 {\csuse{Xendsympagenum@\@currentseries}}}%
4967 }%
4968 {%
4969 \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
4970 }%
4971 \fi%
4972 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag and the subline number.

```

4973 \ifl@d@elin%
4974 \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
4975 \wrap@edcrossref{\@this@crossref@end}{%
4976 \ifledRcol@%
4977 \linenumrepR{#5}%
4978 \else%
4979 \linenumrep{#5}%
4980 \fi%
4981 }%
4982 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{%
4983 \fi%
4984 \ifl@d@esl%
4985 \ifl@d@elin%
4986 \csuse{Xendsublinesep@\@currentseries}%
4987 \fi%
4988 \wrap@edcrossref{\@this@crossref@end}{%
4989 \ifledRcol@%
4990 \sublinenumrepR{#6}%
4991 \else%
4992 \sublinenumrep{#6}%
4993 \fi%
4994 }%
4995 \fi%

```

```
4996 \fi%
4997 %
```

Close the ending line box.

```
4998 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
4999 {}%
5000 {\hfill}%Prevent underfull hbox
5001 \egroup%
5002 %
```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```
5003 \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
5004 {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%
5005 {\csxdef{prevpagerange@\@currentseries}{#1-#4}}%
5006 {\csxdef{prevpagenum@\@currentseries}{#4}}%
5007 }%
5008 {}%
5009 %
```

Now, the end of \printendlines macro.

```
5010 \endgroup%
5011 }%
5012
5013 %
```

\printnpnum A macro to print a page number in an endnote. Should not be override anymore

```
5014 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@\@currentseries}
5015 }#1\csuse{Xendafterpagenumber@\@currentseries}}
5016 %
```

XX **Generate series of notes**

In this section, X means the name of the series (A, B etc.)

\series \series\series creates one more new series. It is a public command, which just loops on the private command \newseries@.

```
5017 \newcommand{\newseries}[1]{%
5018 \def\do##1{\newseries@{##1}}%
5019 \docsvlist{#1}
5020 }
5021 %
```

\@series The \series@ macro is an etoolbox list, which contains the name of all series.

```

5022 \newcommand{\@series}{}
5023 %

```

The command `\newseries@series` creates a new series of the footnote.

```

\newseries@24 \newcommand{\newseries@}[1]{
5025 %

```

XX.1 Test if series is still existing

```

5026 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
5027 {%
5028 %

```

XX.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```

5029 \ifdefined\newseries@par%
5030 \newseries@par{#1}%
5031 \fi%
5032 %

```

XX.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```

5033 \unless\ifnocritical@
5034 %

```

XX.3.1 Options

```

5035 \newtoggle{Xlineflag@#1}
5036 \newtoggle{Xparindent@#1}
5037 \newtoggle{Xlemmadisablefontselection@#1}
5038 \csgdef{Xwrapcontent@#1}{}%
5039 \csgdef{Xbeforeinserting@#1}{}%
5040 \csgdef{Xhangindent@#1}{Opt}%
5041 \csgdef{Xragged@#1}{}%
5042 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
5043 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
5044 \csgdef{Xcolalign@#1}{\raggedright}%
5045 \csgdef{Xnotenumfont@#1}{\normalfont}%
5046 \csgdef{Xnotefontsize@#1}{\footnotesize}%
5047 \csgdef{Xhooknote@#1}{}%

```

```

5048 \csgdef{Xbhookgroup@#1}{}%
5049
5050 \csgdef{Xboxlinenum@#1}{Opt}%
5051 \csgdef{Xboxlinenumalign@#1}{L}%
5052
5053 \csgdef{Xboxstartlinenum@#1}{Opt}%
5054 \csgdef{Xboxendlinenum@#1}{Opt}%
5055
5056 \csgdef{Xboxsymlinenum@#1}{Opt}%
5057 \newtoggle{Xnumberonlyfirstinline@#1}%
5058 \newtoggle{Xgroupbyline@#1}%
5059 \newtoggle{Xgroupbylineseparatetwolines@#1}%
5060 \newtoggle{Xnumberonlyfirstintwolines@#1}%
5061 \csgdef{Xtwolines@#1}{}%
5062 \csgdef{Xmorethantwolines@#1}{}%
5063 \csgdef{Xsublinesep@#1}{\fullstop}%
5064 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
5065 \newtoggle{Xtwolinesbutnotmore@#1}%
5066 \newtoggle{Xtwolinesonlyinsamepage@#1}%
5067 \newtoggle{Xonlypstart@#1}%
5068 \newtoggle{Xpstarteverytime@#1}%
5069 \newtoggle{Xpstart@#1}%
5070 \newtoggle{Xstanza@#1}%
5071 \csgdef{Xstanzaseparator@#1}{}%
5072 \csgdef{Xsymlinenum@#1}{}%
5073 \newtoggle{Xnonumber@#1}%
5074 \csgdef{Xbeforenumber@#1}{Opt}%
5075 \csgdef{Xtxtbeforenumber@#1}{}%
5076 \csgdef{Xafternumber@#1}{0.5em}%
5077 \newtoggle{Xnonbreakableafternumber@#1}%
5078 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
5079 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
5080 \csgdef{Xinplaceofnumber@#1}{1em}%
5081 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
5082 \csgdef{Xbeforelemmaseparator@#1}{0em}%
5083 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
5084 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
5085 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}
5086 \csgdef{Xafterrule@#1}{Opt}
5087
5088 \csgdef{Xtxtbeforenotes@#1}{%
5089 \newtoggle{Xtxtbeforesnotes@#1@typeset}}%Not directly used by user,
but internal
5090 \newtoggle{Xtxtbeforenotesonlyonce@#1}%
5091
5092 \csgdef{Xmaxhnotes@#1}{0.8\vsizer}
5093 \newtoggle{Xnoteswidthliketwocolumns@#1}%
5094 \csgdef{Xparafootsep@#1}{}%
5095 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}

```

```

5096 \csgdef{Xlinrangeseparator@#1}{\endashchar}%
5097
5098 \csgdef{Xlemmafont@#1}{}%
5099 \csgdef{Xwraplemma@#1}{%
5100 \csgdef{Xwidth@#1}{\hsize}%
5101 %

```

XX.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

5102 \expandafter\newinsert\csname #1footins\endcsname%
5103 \unless\ifnoledgroup%
5104 \expandafter\newinsert\csname mp#1footins\endcsname%
5105 \fi%
5106 %

```

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.

```

5107 \global\newbool{parapparatus@}{\expandafter\newcommand\expandafter
5108 *}{\expandafter\newcommand}\csname #1footnote\endcsname[2][]{%
5109 \if@edtext@secondarg%
5110 \ifledRcol%
5111 \ifcsstring{Xonlyside@#1}{L}{\
5112 led@error@note@called@onrightside{#1footnote}}}%
5113 \else%
5114 \ifcsstring{Xonlyside@#1}{R}{\
5115 led@error@note@called@onleftside{#1footnote}}}%
5116 \fi%
5117 \beginngroup%
5118 \newcommand{\content}{##2}%
5119 \ifnumberedpar%
5120 \ifledRcol%
5121 \ifluatex%
5122 \footnotelang@lua[R]%
5123 \fi%
5124 \ifundefined{xpg@main@language}%if polyglossia
5125 {}%
5126 {\footnotelang@poly[R]}%
5127 \footnoteoptions@{R}{##1}{true}%
5128 \xright@appenditem{%
5129 \ifbool{indtl@innote}%
5130 {\unexpanded{\let\index\nindex}}%
5131 {}%
5132 \ifbool{indtl@notenumber}%
5133 {\unexpanded{\let\index\nindex}}%There is no note
5134 ...number so
5135 {}%

```

```

5132         \noexpand\Xnote@true%
5133         \noexpand\prepare@Xprenotes{#1}%
5134         \noexpand\prepare@edindex@fornote{\l@d@nums}%
5135         \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.
5136         \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
5137         \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5138         \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5139         \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
5140         \noexpand\csuse{v#1footnote}{#1}%
5141         {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
5142         \noexpand\Xnote@false%
5143         \unexpanded{\advance\@edindex@fornote@m@one}%
5144         \unexpanded{\undef\@this@crossref@start}%
5145         \unexpanded{\undef\@this@crossref@end}%
5146         \ifbool{indtl@innote}%
5147             {\unexpanded{\let\index\orig@@index}}%
5148             {}%
5149         \ifbool{indtl@notenumber}%
5150             {\unexpanded{\let\index\orig@@index}}%
5151             {}%
5152         }\to\inserts@listR
5153         \footnoteoptions@{R}{#1}{false}%
5154         \global\advance\insert@countR \@ne%
5155     \else%
5156         \ifluatex%
5157             \footnotelang@lua%
5158             \fi%
5159         \@ifundefined{xpg@main@language}%if polyglossia
5160             {}%
5161             {\footnotelang@poly}%
5162         \footnoteoptions@{L}{#1}{true}%
5163         \xright@appenditem{%
5164             \ifbool{indtl@innote}%
5165                 {\unexpanded{\let\index\nindex}}%
5166                 {}%
5167             \ifbool{indtl@notenumber}%
5168                 {\unexpanded{\let\index\nindex}}%There is no note
...number so
5169             {}%
5170         \noexpand\Xnote@true%
5171         \noexpand\prepare@Xprenotes{#1}%
5172         \noexpand\prepare@edindex@fornote{\l@d@nums}%

```



```

5173 \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.
5174 \ifl@dpairing%
5175 \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
5176 \fi%
5177 \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5178 \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5179 \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
5180 \noexpand\csuse{v#1footnote}%
5181 {#1}%
5182 {{\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
5183 \unexpanded{\undef\@this@crossref@start}%
5184 \unexpanded{\undef\@this@crossref@end}%
5185 \noexpand\Xnote@false%
5186 \unexpanded{\advance\@edindex@fornote@m@ne}%
5187 \ifbool{indtl@innote}%
5188 {\unexpanded{\let\index\orig@index}}%
5189 {}%
5190 \ifbool{indtl@notenumber}%
5191 {\unexpanded{\let\index\orig@index}}%
5192 {}%
5193 }\to\inserts@list
5194 \global\advance\insert@count \@ne%
5195 \footnoteoptions@{L}{##1}{false}%
5196 \fi
5197 \else
5198 \csuse{v#1footnote}{#1}{0|0|0|0|0|0|0|0}{##1}%
5199 \fi%
5200 \endgroup%
5201 \else%
5202 \led@err@FootnoteNotInSecondArgEdtext{#1}%
5203 \fi%
5204 \ignorespaces%
5205 }
5206 %

```

Create counter used to determine on which page the previous note was called.

```

5207 \expandafter\newcount\csname #1prevpage@num\endcsname%
5208 \expandafter\newcount\csname #1prevpage@numR\endcsname%
5209 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

5210 \global\csletcs{#1@@footnote}{#1footnote}
5211 %

```

XX.3.4 Set standard display

```
5212 \Xarrangement@normal{#1}%
5213 %
```

End of for critical footnotes.

```
5214 \fi
5215 %
```

XX.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```
5216 \unless\ifnofamiliar@
5217 %
```

XX.4.1 Options

```
5218 \newtoggle{parindentX@#1}
5219 \csgdef{wrapcontentX@#1}{}%
5220 \csgdef{hangindentX@#1}{Opt}%
5221 \csgdef{beforeinsertingX@#1}{}%
5222 \csgdef{raggedX@#1}{}%
5223 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
5224 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
5225 \csgdef{colalignX@#1}{\raggedright}%
5226 \csgdef{notenumfontX@#1}{\normalfont}%
5227 \csgdef{notefontsizeX@#1}{\footnotesize}%
5228 \csgdef{bhooknoteX@#1}{}%
5229 \csgdef{bhookgroupX@#1}{}%
5230 \csgdef{afterruleX@#1}{Opt}
5231 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
5232 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
5233 \newtoggle{noteswidthliketwocolumnsX@#1}%
5234 \csgdef{parafootsepX@#1}{}%
5235 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
5236 \csgdef{widthX@#1}{\hsizex}%
5237 \csgdef{txtbeforenotesX@#1}{}%
5238 \newtoggle{txtbeforenotesX@#1@typeset}%Not directly used by user,
but internal
5239 \newtoggle{txtbeforenotesonlyonceX@#1}%
5240 % End of for familiar footnotes.
5241 % \subsubsection{Create inserts, needed to add notes in foot}
5242 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
5243 % \begin{macrocode}
5244 \expandafter\newinsert\csname footins#1\endcsname%
5245 \unless\ifnoledgroup@%
5246 \expandafter\newinsert\csname mpfootins#1\endcsname%
5247 \fi%
```

5248 %

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.

```
5249
5250 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
5251     \begingroup%
5252     \prepare@prenotesX{#1}%
5253     \newcommand{\content}{##1}%
5254 %
```

If we use the \csquotes package, we reset quotation level.

```
5255     \ifdefined\csq@qllevel%
5256     \csq@qllevel=0\relax%
5257 \fi%
5258 %
```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read reledpar's handbook about that (V.1.2 p. 50).

```
5259 \global\expandafter\advance\csname footnote#1@reading\
5260 endcsname by \@ne%
5261 \ifboolexpr{bool{1@dpairing} or bool{1@dprintingpages} or
5262 bool{1@dprintingcolumns}}{%
5263     \ifcsdef{footnote#1reading\the\csname footnote#1@reading\
5264 endcsname=typeset}%
5265     {\setcounter{footnote#1}{\csuse{footnote#1reading\the\
5266 csname footnote#1@reading\endcsname=typeset}}}%
5267     {\setcounter{footnote#1}{\the\csname footnote#1@reading
5268 \endcsname}}}%
5269 }{%
5270     \stepcounter{footnote#1}%
5271 }%
5272 %
```

We also have to check consistency with \onlysideX setting.

```
5268 \ifledRcol%
5269 \ifcsstring{onlysideX@#1}{L}{\
5270 led@error@note@called@onrightside{footnote#1}}{%
5271 \else%
5272 \ifcsstring{onlysideX@#1}{R}{\
5273 led@error@note@called@onleftside{footnote#1}}{%
5274 \fi%
5275 %
```

And now, the feature not depending of whether we are preparing parallel typesetting

```
5274 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5275 \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
```

```

5276         {\csuse{@footnotemark#1}}%
5277         {}%
5278         \ifluatex%
5279         \xdef\footnote@luatextextdir{\the\textdir}%
5280         \xdef\footnote@luatexpardir{\the\pardir}%
5281         \fi%
5282         \if@ledgroup%
5283         \led@set@index@fornote{#1}%
5284         \fi%
5285         \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
5286         \ifbool{indtl@innote}%
5287         {\let\index\orig@@index}%
5288         {}%
5289         \ifbool{indtl@notenumber}%
5290         {\let\index\orig@@index}%
5291         {}%
5292         \endgroup%
5293     }
5294 %

```

Then define the counters. The \TeX 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).

```

5295     \newcounter{footnote#1}
5296     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
5297     \expandafter\newcount\csname footnote#1@reading\endcsname%
5298 %

```

Create counter used to determine on which page the previous note was called.

```

5299     \expandafter\newcount\csname prevpage#1@num\endcsname%
5300     \expandafter\newcount\csname prevpage#1@numR\endcsname%
5301 %

```

Add `\let\footnoteX\@gobble` to `\no@expands`.

```

5302     \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
5303 %

```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to classical `\footnotemark` and `\footnotetext`.

```

5304     \expandafter\newcommand\csname footnote#1mark\endcsname{%
5305         \begingroup%
5306         \prepare@prenotesX{#1}%
5307         \stepcounter{footnote#1}%
5308         \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5309         \csuse{@footnotemark#1}%
5310         \m@mmf@prepare%

```

```

5311 \endgroup%
5312 }%
5313 \expandafter\newcommand\csname footnote#1text\endcsname[1]{%
5314 \begingroup%
5315 \csuse{vfootnote#1}{#1}{\expandonce{##1}}%
5316 \endgroup%
5317 }%
5318 %

```

Do not forget to initialize the series.

```

5319 \arrangementX@normal{#1}%
5320 \fi
5321 %

```

XX.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

5322 \unless\ifnoend@
5323 %

```

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`

```

5324 \expandafter\newwrite\csname l@d@#1end\endcsname%
5325 \expandafter\newif\csname ifl@dend@#1\endcsname%
5326 %

```

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.

```

5327
5328 \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,
usedefault]{%
5329 \bgroup%
5330 \newlinechar='40%
5331 \global\@noneed@Footnotetrue%
5332 \newcommand{\content}{##2}%
5333 \stepcounter{labidx}%
5334 \expandafter\immediate\expandafter\write\csname l@d@#1end\
endcsname{%

```

```

5335 \unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\
expandafter{\sw@inthisedtext}}\@percentchar\space%Explicit space, to add a
linebreak in the output file
5336 \expandafter\string\csname #1end\endcsname%
5337 {\ifnumberedpar@l@d@nums\fi}%
5338 {\ifnumberedpar@expandonce\@tag\fi}%
5339 {\expandonce\content}%
5340 {\#1}%
5341 {\unexpanded{##1}}%
5342 {\ifledRcol R\else L\fi}%
5343 {\theedtext}%
5344 \@percentchar%
5345 }%
5346 \egroup%
5347 \ignorespaces%
5348 }%
5349 %

```

XX.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

5350 \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
5351 \toendnotes[#1]{##1}%
5352 }%
5353
5354 \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
5355 endcsname*[1]{%
5356 \toendnotes*[#1]{##1}%
5357 }%
5358
5359 %

```

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.

```

5360
5361 \global\cslet{#1end}{\@gobbleseven}
5362 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

5363 \global\expandafter\newcount\csname #1end@bysection\endcsname%
5364 %

```

XX.5.5 The options

```

5365 \csgdef{Xendwraplemma@#1}{%
5366 \csgdef{Xendwrapcontent@#1}{}%
5367 \csgdef{Xendtwolines@#1}{}%
5368 \csgdef{Xendmorethantwolines@#1}{}%
5369 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
5370 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
5371 \newtoggle{Xendlemmadisablefontselection@#1}{}%
5372 \csgdef{Xendnotenumfont@#1}{\normalfont}%
5373 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
5374 \csgdef{Xendbhooknote@#1}{}%
5375
5376 \csgdef{Xendsublinesep@#1}{\fullstop}%
5377
5378 \csgdef{Xendbeforenumber@#1}{Opt}
5379 \csgdef{Xendafternumber@#1}{0.5em}
5380
5381 \csgdef{Xendboxlinenum@#1}{Opt}%
5382 \csgdef{Xendboxlinenumalign@#1}{L}%
5383
5384 \csgdef{Xendboxstartlinenum@#1}{Opt}%
5385 \csgdef{Xendboxendlinenum@#1}{Opt}%
5386
5387 \csgdef{Xendlemmaseparator@#1}{}%
5388 \csgdef{Xendbeforelemmaseparator@#1}{0em}%
5389 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
5390 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
5391
5392 \newtoggle{Xendparagraph@#1}%
5393 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
5394 \csgdef{Xendsep@#1}{}%
5395
5396 \csgdef{Xendinplaceofnumber@#1}{Opt}%
5397 \newtoggle{Xendnonumber@#1}%
5398
5399 \csgdef{Xendhangindent@#1}{Opt}%
5400 \newtoggle{Xendnumberonlyfirstinline@#1}%
5401 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
5402
5403 \csgdef{Xendbeforesymlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
5404 \csgdef{Xendaftersymlinenum@#1}{\csuse{Xendafternumber@#1}}%
5405 \csgdef{Xendsymlinenum@#1}{}%
5406 \csgdef{Xendboxsymlinenum@#1}{Opt}%
5407
5408 \csgdef{Xendbhooklinenum@#1}{}%
5409 \csgdef{Xendehooklinenum@#1}{}%
5410 \csgdef{Xendbhookinplaceofnumber@#1}{}%
5411 \csgdef{Xendehookinplaceofnumber@#1}{}%
5412

```

```

5413 \csgdef{Xendlinrangeseparator@#1}{\endashchar}%
5414
5415 \csgdef{Xendbeforepagenumber@#1}{p.}%
5416 \csgdef{Xendafterpagenumber@#1}{) }%
5417 \csgdef{Xendlineprefixsingle@#1}{}%
5418 \csgdef{Xendlineprefixmore@#1}{}%
5419
5420 \newtoggle{Xendlineflag@#1}
5421
5422 \csgdef{Xendlemmafont@#1}{}%
5423
5424 \newtoggle{Xendpagenumberonlyfirst@#1}%
5425 \newtoggle{Xendpagenumberonlyfirstifsingle@#1}%
5426 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
5427 \csgdef{Xendsympagenum@#1}{}%
5428 \csgdef{Xendinplaceofpagenumber@#1}{0pt}%
5429
5430 %

```

End of endnotes declaration

```

5431 \fi%
5432 %

```

Dump series in \@series

```

5433 \listxadd{\@series}{#1}
5434 }
5435 }% End of \newseries
5436 %

```

XX.6 Init standards series (A,B,C,D,E)

```

5437 \expandafter\newseries\expandafter{\default@series}
5438 %

```

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.

```

5439 \newcommand{\seriesatbegin}[1]{%
5440   \StrDel{\@series}{#1}[\@series]%
5441   \edef\@new{%
5442     \listead{\@new}{#1}%
5443     \listead{\@new}{\@series}%
5444     \xdef\@series{\@new}%
5445   }
5446 %

```


`\seriesatend` And `\seriesatend` moves the series to the end of the list.

```

5447 \newcommand{\seriesatend}[1]{%
5448   \StrDel{\@series}{#1}[\@series]%
5449   \edef\@new{}%
5450   \listadd{\@new}{\@series}%
5451   \listadd{\@new}{#1}%
5452   \xdef\@series{\@new}%
5453 }
5454 %

```

XXI.2 Test series order

`\ifseriesbefore` `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands `<true>` if `<seriesA>` is printed before `<seriesB>`, expands `<false>` otherwise.

```

5455 \newcommand{\ifseriesbefore}[4]{%
5456   \StrPosition{\@series}{#1}[\@first]%
5457   \StrPosition{\@series}{#2}[\@second]%
5458   \ifnumgreater{\@second}{\@first}{#3}{#4}%
5459 }
5460 %

```

XXI.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries61 \newcommand{\@getfirstseries}{%
5462   \ifdefempty{\@series}%
5463   {\xdef\@firstseries{}}%
5464   {\StrChar{\@series}{1}[\@firstseries]}%
5465 }%
5466 %

```

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`.

```

5467 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
5468   \def\do##1{%
5469     \global\settoggle{#2@##1}{#3}%
5470     \ifstrequal{#4}{critical}{
5471       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
5472     }{}
5473     \ifstrequal{#4}{familiar}{
5474       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
5475     }{}
5476   }%
5477   \ifstreempty{#1}{%
5478     \dolistloop{\@series}%
5479     \ifstreempty{#5}{}{%
5480       \docsvlist{#5}%
5481     }
5482   }%
5483   {%
5484     \docsvlist{#1}%
5485   }%
5486 }
5487 %

```

`\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`.

```

5488 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
5489   \def\do##1{
5490     \csgdef{#2@##1}{#3}
5491     \ifstrequal{#4}{critical}{%
5492       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
5493     }{
5494       \ifstrequal{#4}{familiar}{%
5495         \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
5496       }{
5497     }%
5498     \ifstreempty{#1}{%
5499       \dolistloop{\@series}%
5500       \ifstreempty{#5}{%
5501         \docsvlist{#5}
5502       }
5503     }%
5504     {%
5505       \docsvlist{#1}%
5506     }%
5507   }%
5508   %

```

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`.

```

5509 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
5510   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5511     \setcommand@series{##1}{#1}{##2}[] [#2]%
5512   }%
5513   \ifstreempty{#2}{%
5514     \def\do##1{%

```

```

5515 \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
[1]{%
5516 \csuse{#1}[##1]{###1}%
5517 }%
5518 }%
5519 \docsvlist{#2}%
5520 }%
5521 }
5522 %

```

\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`.

```

5523 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
5524 \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
5525 \settoggle@series{##1}{#1}{##2}[][#2]%
5526 }%
5527 \ifstrepty{#2}{-}{%
5528 \def\do##1{%
5529 \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
5530 \csuse{#1}[##1]%
5531 }%
5532 }%
5533 \docsvlist{#2}%
5534 }%
5535 }
5536 %

```

\newhooktoggle@series@reload `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement, depending of type os notes

```

5537 \newcommand{\newhooktoggle@series@reload}[2]{%
5538 \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
true},usedefault]{%
5539 \settoggle@series{##1}{#1}{##2}[#2]%
5540 }%
5541 }%
5542 %

```

\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.

```

5543 \newcommand{\newhookcommand@series@reload}[2]{%
5544 \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5545 \setcommand@series{##1}{#1}{##2}[#2]%
5546 }%

```

```

5547 }
5548 %

```

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.

```

5549 \unless\ifnocritical@
5550 \newhookcommand@series{Xwrapcontent}%
5551 \newhookcommand@series{Xbeforeinserting}%
5552 \newhookcommand@series{Xlemmafont}%
5553 \newhookcommand@series{Xwraplemma}%
5554 \newhooktoggle@series{Xparindent}
5555 \newhookcommand@series{Xhangindent}
5556 \newhookcommand@series{Xragged}
5557 \newhookcommand@series{Xhsizetwocol}
5558 \newhookcommand@series{Xhsizethreecol}
5559 \newhookcommand@series{Xcolalign}%
5560 \newhookcommand@series{Xnotenumfont}
5561 \newhookcommand@series{Xbhooknote}
5562 \newhookcommand@series@reload{Xbhookgroup}{critical}
5563 \newhookcommand@series{Xboxsymlinenum}%
5564 \newhookcommand@series{Xsymlinenum}
5565 \newhookcommand@series{Xbeforenumber}
5566 \newhookcommand@series{Xtxtbeforenumber}
5567 \newhookcommand@series{Xafternumber}
5568 \newhookcommand@series{Xbeforesymlinenum}
5569 \newhookcommand@series{Xaftersymlinenum}
5570 \newhookcommand@series{Xinplaceofnumber}
5571 \newhookcommand@series{Xlemmaseparator}
5572 \newhookcommand@series{Xbeforelemmaseparator}
5573 \newhookcommand@series{Xafterlemmaseparator}
5574 \newhookcommand@series{Xinplaceoflemmaseparator}
5575 \newhookcommand@series{Xtxtbeforenotes}
5576 \newhooktoggle@series{Xtxtbeforenotesonlyonce}%
5577 \newhookcommand@series@reload{Xafterrule}{critical}
5578 \newhooktoggle@series{Xnumberonlyfirstinline}
5579 \newhooktoggle@series{Xnumberonlyfirstintwolines}
5580 \newhooktoggle@series{Xgroupbyline}%
5581 \newhooktoggle@series{Xgroupbylineseparatetwolines}%
5582 \newhooktoggle@series{Xnonumber}
5583 \newhooktoggle@series{Xpstart}
5584 \newhooktoggle@series{Xpstarteverytime}%
5585
5586 \newhooktoggle@series{Xstanza}%
5587 \newhookcommand@series{Xstanzaseparator}%
5588
5589 \newhooktoggle@series{Xonlypstart}

```

```

5590 \newhooktoggle@series{Xnonbreakableafternumber}
5591 \newhooktoggle@series{Xlemmadisablefontselection}
5592 \newhookcommand@series@reload{Xmaxhnotes}{critical}
5593 \newhookcommand@series@reload{Xbeforenotes}{critical}
5594 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
5595 \newhookcommand@series@reload{Xnotefontsize}{critical}
5596
5597 \newhookcommand@series{Xboxlinenum}%
5598 \newhookcommand@series{Xboxlinenumalign}%
5599
5600 \newhookcommand@series{Xboxstartlinenum}%
5601 \newhookcommand@series{Xboxendlinenum}%
5602
5603 \newhookcommand@series{Xafternote}%
5604 \newhookcommand@series{Xparafootsep}
5605
5606 \newhookcommand@series@reload{Xwidth}{critical}%
5607
5608 \ifundef{\Xhsize}%
5609 {
5610   \newcommandx{\Xhsize}[2][1,usedefault]{%
5611     \led@warning@Xhsize@deprecated%
5612     \Xwidth[#1]{#2}%
5613   }%
5614 }%
5615 {}%
5616 \fi
5617 \newhooktoggle@series{Xlineflag}[appref,SEref]
5618 \newhookcommand@series{Xtwolines}[appref,SEref]
5619 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
5620 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
5621 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
5622 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
5623 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
5624 \newhookcommand@series{Xlinerrangeseparator}[appref,SEref]
5625 %

```

XXI.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

5626 \unless\ifnofamiliar@
5627   \newhookcommand@series{wrapcontentX}%
5628   \newhookcommand@series{beforeinsertingX}%
5629   \newhooktoggle@series{parindentX}
5630   \newhookcommand@series{hangindentX}
5631   \newhookcommand@series{raggedX}
5632   \newhookcommand@series{hsizetwocolX}
5633   \newhookcommand@series{hsizethreecolX}

```

```

5634 \newhookcommand@series{colalignX}%
5635 \newhookcommand@series{notenumfontX}
5636 \newhookcommand@series{bhooknoteX}
5637 \newhookcommand@series@reload{bhookgroupX}{familiar}
5638 \newhookcommand@series@reload{beforenotesX}{familiar}
5639 \newhookcommand@series@reload{maxhnotesX}{familiar}
5640 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
5641 \newhookcommand@series@reload{afterruleX}{familiar}
5642 \newhookcommand@series@reload{notefontsizeX}{familiar}
5643 \newhookcommand@series{afternoteX}
5644 \newhookcommand@series{parafootsepX}
5645 \newhookcommand@series{txtbeforenotesX}%
5646 \newhooktoggle@series{txtbeforenotesonlyonceX}%
5647 \newhookcommand@series@reload{widthX}{familiar}%
5648 \ifundef{\hsizeX}%
5649   {%
5650     \newcommandx{\hsizeX}[2][1,usedefault]{%
5651       \led@warning@hsizeX@deprecated%
5652       \widthX[#1]{#2}%
5653     }%
5654   }%
5655   {}%
5656 \fi
5657 %

```

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.

```

5658 \unless\ifnoend@
5659   \newhookcommand@series{Xendwraplemma}
5660   \newhookcommand@series{Xendwrapcontent}
5661   \newhookcommand@series{Xendnotenumfont}
5662   \newhookcommand@series{Xendlemmafont}%
5663   \newhookcommand@series{Xendbhooknote}
5664
5665   \newhookcommand@series{Xendboxlinenum}%
5666   \newhookcommand@series{Xendboxlinenumalign}%
5667
5668   \newhookcommand@series{Xendboxstartlinenum}%
5669   \newhookcommand@series{Xendboxendlinenum}%
5670
5671   \newhookcommand@series{Xendnotefontsize}
5672   \newhooktoggle@series{Xendlemmadisablefontselection}
5673   \newhookcommand@series{Xendlemmaseparator}
5674   \newhookcommand@series{Xendbeforelemmaseparator}
5675   \newhookcommand@series{Xendafterlemmaseparator}
5676   \newhookcommand@series{Xendinplaceoflemmaseparator}
5677

```

```

5678 \newhookcommand@series{Xendbeforenumber}%
5679 \newhookcommand@series{Xendafternumber}%
5680
5681 \newhooktoggle@series{Xendparagraph}
5682 \newhookcommand@series{Xendafternote}
5683 \newhookcommand@series{Xendsep}
5684
5685 \newhookcommand@series{Xendinplaceofnumber}%
5686 \newhooktoggle@series{Xendnonumber}%
5687
5688 \newhooktoggle@series{Xendnumberonlyfirstinline}%
5689 \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
5690
5691 \newhookcommand@series{Xendsymmlinenumber}%
5692 \newhookcommand@series{Xendbeforesymmlinenumber}%
5693 \newhookcommand@series{Xendaftersymmlinenumber}%
5694 \newhookcommand@series{Xendboxsymmlinenumber}%
5695
5696 \newhookcommand@series{Xendbhooklinenumber}%
5697 \newhookcommand@series{Xendahooklinenumber}%
5698 \newhookcommand@series{Xendbhookinplaceofnumber}%
5699 \newhookcommand@series{Xendahookinplaceofnumber}%
5700
5701 \newhookcommand@series{Xendhangindent}%
5702
5703 \newhooktoggle@series{Xendpagenumberonlyfirst}%
5704 \newhooktoggle@series{Xendpagenumberonlyfirstifsinglenumber}%
5705 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
5706 \newhookcommand@series{Xendsympagenumber}%
5707 \newhookcommand@series{Xendinplaceofpagenumber}%
5708
5709 \fi
5710 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
5711 \newhookcommand@series{Xendt看两lines}[apprefwithpage,SErefwithpage]
5712 \newhookcommand@series{Xendmoreethant看两lines}[apprefwithpage,SErefwithpage]
5713 \newhooktoggle@series{Xendt看两linesbutnotmore}[apprefwithpage,SErefwithpage]
5714 \newhooktoggle@series{Xendt看两linesonlyinsamepage}[apprefwithpage,
SErefwithpage]
5715 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage
]
5716 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
SErefonlypage]
5717 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
5718 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
5719 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
5720 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
5721
5722 %

```


XXI.4 Hooks for a particular footnote

\newhooktoggle@specific `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

5723 \newcommand{\newhooktoggle@specific}[1]{%
5724   \newtoggle{#1}%
5725   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settoggle{#1}{true}}%
When enabling footnote option
5726   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settoggle{#1}{false}
5727   }}
5727 }
5728 %

```

\newhookarg@specific `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

5729 \newcommand{\newhookarg@specific}[1]{%
5730   \define@key[mac]{truefootnoteoption}{#1}{\global\def\linrangesep@{##1}}%
When enabling footnote option
5731   \define@key[mac]{falsefootnoteoption}{#1}{\global\undef\linrangesep@}%
When
5732 }
5733 %

```

And now, we define some hooks specific to a note.

```

5734 \newhooktoggle@specific{fulllines}%
5735 \newhooktoggle@specific{nonum}
5736 \newhooktoggle@specific{nosep}
5737 \newhookarg@specific{linrangesep}
5738 %

```

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.

```

5739 %

```

\nomk@ `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```

5740 \newtoggle{nomk}%
5741 %

```

XXI.5 Alias

\Xnolemmaseparator `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```

5742 \newcommandx*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]}
5743 %

```

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

`\doextrafeet` `\doextrafeet` is the code extending `\@makecol` to cater for the extra `reledmac` feet. We have two categories of extra footnotes. By default, we order the footnote inserts so that the regular footnotes of \TeX are first, then familiar familiar footnotes and finally the critical footnotes.

```

5744 \newcommand*{\l@ddoxtrafeet}{%
5745   \IfStrEq{familiar-critical}{\@fnpos}
5746     {\do@feetX\do@Xfeet}%
5747   {%
5748     \IfStrEq{critical-familiar}{\@fnpos}%
5749     {\do@Xfeet\do@feetX}%
5750     {%
5751       \setbox\@outputbox \vbox{%
5752         \unvbox\@outputbox%
5753         \do@feet@custom@order{\@fnpos}%
5754       }%
5755     }%
5756   }%
5757 }%
5758 %
5759 %

```

`\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.

```

5760 \newcommand{\do@feet@custom@order}[2]{%
5761   \def\do##1{%
5762     \edef\@notesseries{\@firstoftwo##1}%
5763     \edef\@notetype{\@secondoftwo##1}%
5764     \ifdefstring{\@notetype}{critical}%
5765       {\csuse{#1append@Xnotes}{\@notesseries}}%
5766     {\ifdefstring{\@notetype}{familiar}%
5767       {\csuse{#1append@notesX}{\@notesseries}}%
5768     }%
5769   }%
5770 }%
5771 \expandafter\docsvlist\expandafter{#2}%
5772 }%

```

```
5773 %
```

\do@Xfeet \do@Xfeet is the code extending \@makecol to cater to the extra critical feet.

```
5774 \newcommand*\do@Xfeet}{%
5775   \setbox\@outputbox \vbox{%
5776     \unvbox\@outputbox
5777     \op@Xfeet}}
5778 %
```

\@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

```
5779 \newcommand{\append@Xnotes}[1]{%
5780   \ifvoid\csuse{#1footins}\else%
5781     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
5782     \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
5783     \print@Xnotes{#1}%
5784   \fi%
5785 }%
5786 %
```

The normal way to add one series, \print@Xnotes, is replaced by reledpar when using \Pages.

```
5787 \newcommand\print@Xnotes[1]{%
5788   \xdef\@currentseries{#1}%
5789   \csuse{#1footstart}{#1}%
5790   \csuse{#1footgroup}{#1}%
5791 }%
5792 %
```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```
5793 \newcommand*\@opXfeet}{%
5794   \unless\ifnocritical@%
5795     \gdef\firstXseries@{}%
5796     \def\do##1{%
5797       \append@Xnotes{##1}%
5798     }%
5799     \dolistloop{\@series}%
5800   \fi%
5801 }%
5802 %
```

\l@ddodoreinextrafeet \l@ddodoreinextrafeet is the code for catering for the extra footnotes within \@reinserts. We use the same category and ordering as in \l@ddoxtrafeet.

```
5803 \newcommand*\l@ddodoreinextrafeet}{%
5804   \IfStrEq{familiar-critical}{\@fnpos}
```

```

5805 {\@doreinfeetX\X@doreinfeet}%
5806 {%
5807 \IfStrEq{critical-familiar}{\@fnpos}%
5808 {\X@doreinfeet\@doreinfeetX}%
5809 {\@doreinfeetX\X@doreinfeet}%
5810 }%
5811 }
5812
5813 %

```

\X@doreinfeet \X@doreinfeet is the code for catering for the extra critical footnotes within \@reinserts.

```

5814 \newcommand*{\X@doreinfeet}{%
5815 \unless\ifnocritical@%
5816 \def\do##1{%
5817 \ifvoid\csuse{##1footins}\else%
5818 \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
5819 \fi}%
5820 \dolistloop{\@series}
5821 \fi%
5822 }
5823
5824 %

```

\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.

```

5825 \newcommand{\append@notesX}[1]{%
5826 \ifvoid\csuse{footins#1}\else%
5827 \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
5828 \global\advance\skip\csuse{footins#1} by\csuse{afterruleX@#1}%
5829 \print@notesX{#1}%
5830 \fi%
5831 }%
5832 %

```

The normal way to print one series of notes. \print@Xnotes is replaced by reledpar when using \Pages.

```

5833 \newcommand\print@notesX[1]{%
5834 \xdef\@currentseries{#1}%
5835 \csuse{footstart#1}{#1}%
5836 \csuse{footgroup#1}{#1}%
5837 }%
5838 %

```

We print all the series of notes by looping on them. We check before printing them that they are not voided.

```

5839 \newcommand*{\do@feetX}{%

```

```

5840 \unless\ifnofamiliar@%
5841 \gdef\firstseriesX@{}%
5842 \setbox\@outputbox \vbox{%
5843 \unvbox\@outputbox%
5844 \def\do##1{%
5845 \append@notesX{##1}%
5846 }%
5847 \dolistloop{\@series}}%
5848 \fi%
5849 }%
5850
5851 \newcommand{\@doreinfeetX}{%
5852 \unless\ifnofamiliar@%
5853 \def\do##1{%
5854 \ifvoid\csuse{footins##1}\else
5855 \insert%
5856 \csuse{footins##1}
5857 {\unvbox\csuse{footins##1}}%
5858 \fi%
5859 }%
5860 \dolistloop{\@series}%
5861 \fi%
5862 }%
5863
5864 %

```

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.)

```

5865 \@ifclassloaded{memoir}{%
5866 %
5867
5868 memoir is loaded so we use memoir's built in hooks.
5869
5870 \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
5871 \g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinextrafeet}%
5872 }{%
5873 %

```

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`.

```

5871 \ifbool{expr}%
5872   test{\@ifpackageloaded{fancyhdr}}%
5873   and test {\@ifdef{\latex@makecol}}%
5874 }{%
5875   \patchcmd%
5876     {\latex@makecol}%
5877     {\xdef\@freelist{\@freelist\@midlist}}%
5878     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5879     {}%
5880     {\led@error@fail@patch@@@makecol}%
5881 }{%
5882   \patchcmd%
5883     {\@makecol}%
5884     {\xdef\@freelist{\@freelist\@midlist}}%
5885     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5886     {}%
5887     {\led@error@fail@patch@@@makecol}%
5888 }%
5889
5890 \patchcmd%
5891   {\@reinserts}%
5892   {\ifvbox}%
5893   {\l@ddodoreinextrafeet\ifvbox}%
5894   {}%
5895   {\led@error@fail@patch@@@reinserts}%
5896 }
5897
5898 %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

5899 \newif\if@led@nofoot
5900
5901 %

```

```

5902 \@ifclassloaded{memoir}{%
5903 %

```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet \g@addto@macro{\@mem@extranofeet}{%%
5905   \def\do#1{%
5906     \unless\ifnocritical@%
5907       \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
5908     \fi%
5909     \unless\ifnofamiliar@%
5910       \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
5911     \fi%

```

```

5912 }
5913 \dolistloop{\@series}%
5914 }%
5915 }-%
5916 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot17 \newcommand*{\@led@testifnofoot}{%
\@doclearpage18 \@led@nofoottrue%
5919 \ifvoid\footins\else%
5920 \@led@nofootfalse%
5921 \fi%
5922 \def\do##1{%
5923 \unless\ifnocritical@%
5924 \ifvoid\csuse{##1footins}\else%
5925 \@led@nofootfalse%
5926 \fi%
5927 \fi%
5928 \unless\ifnofamiliar@%
5929 \ifvoid\csuse{footins##1}\else%
5930 \@led@nofootfalse%
5931 \fi%
5932 \fi%
5933 }%
5934 \dolistloop{\@series}%
5935 }-%
5936
5937 \pretocmd%
5938 {\@doclearpage}%
5939 {\@led@testifnofoot}%
5940 {}%
5941 {\led@error@fail@patch@@doclearpage}%
5942
5943 \patchcmd%
5944 {\@doclearpage}%
5945 {\ifvoid\footins}%
5946 {\if@led@nofoot}%
5947 {}%
5948 {\led@error@fail@patch@@doclearpage}%
5949
5950 }
5951
5952 %

```

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.

```
5953 \list@create{\labelref@list}
5954 %
```

\zz@@@ A convenience macro to zero three labeling counters in one go.

```
5955 \newcommand*{\zz@@@}{000|000|000}% Set three counters to zero in one go
5956
5957 %
```

\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.

```
5958 \newcommand*{\edlabel}[1]{%
5959   \ifl@dpairing\ifautopar%
5960     \strut%
5961   \fi\fi%
5962   \@bsphack%
5963   \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
5964     \ifXnote@%
5965       \protected@write\@auxout{%
5966         {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstartR|{#1}}}%
5967       \ifdef{\hypertarget}%
5968         {\Hy@raisedlink{\hypertarget{#1}{}}}%
5969     }
```

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.


```

5969     {}%
5970 \else%
5971   \write\linenum@outR{\string\@lab}%
5972   \ifx\labelref@listR\empty%
5973     \xdef\label@refs{\zz@@@}%
5974   \else%
5975     \gl@p\labelref@listR\to\label@refs%
5976   \fi%
5977   \ifvmode%
5978     \advancelabel@refs%
5979   \fi%
5980 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

5981   \protected@write\@auxout{}%
5982     {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
5983 |{#1}}%
5984   \ifdef{\hypertarget}%
5985     {\Hy@raisedlink{\hypertarget{#1}{}}}%
5986     {}%
5987   \fi%
5988   \ifXnote@%
5989     \ifl@dpairing%Pstart or PstartL?
5990     \protected@write\@auxout{}%
5991       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
5992 l@dparsedstartsub||\the\c@pstartL|{#1}}%
5993     \ifdef{\hypertarget}%
5994       {\Hy@raisedlink{\hypertarget{#1}{}}}%
5995       {}%
5996     \else%
5997       \protected@write\@auxout{}%
5998       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|\
5999 l@dparsedstartsub||\the\c@pstart|{#1}}%
6000     \ifdef{\hypertarget}%
6001       {\Hy@raisedlink{\hypertarget{#1}{}}}%
6002       {}%
6003     \fi%
6004   \else%
6005     \write\linenum@out{\string\@lab}%
6006     \ifx\labelref@list\empty%
6007       \xdef\label@refs{\zz@@@}%
6008     \else%
6009       \gl@p\labelref@list\to\label@refs%
6010     \fi%
6011     \ifvmode%
6012       \advancelabel@refs%
6013     \fi%
6014     \ifl@dpairing%Pstart or PstartL?

```

```

6013     \protected@write\@auxout{}%
6014     {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}}%
6015     \ifdef\hypertarget{%
6016     {\Hy@raisedlink{\hypertarget{#1}{}}}%
6017     }%
6018     \else%
6019     \protected@write\@auxout{}%
6020     {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}}%
6021     \ifdef\hypertarget{%
6022     {\Hy@raisedlink{\hypertarget{#1}{}}}%
6023     }%
6024     \fi%
6025     \fi%
6026   }%
6027 \@@esphack}%
6028
6029 %

```

`\advancelabel@refs` In cases where `\edlabel` is the first element in a paragraph, we have a problem with line counts, because line counts change only at the first horizontal box of the paragraph. Hence, we need to test `\edlabel` if it occurs at the start of a paragraph. To do so, we use `\ifvmode`. If the test is true, we must advance by one unit the amount of text we write into the .aux file. We do so using `\advancelabel@refs` command.

`\labelrefsparseline`
`\labelrefsparsesubline`

```

6030 \newcounter{line}%
6031 \newcounter{subline}%
6032 \newcounter{absline}%
6033 \newcommand{\advancelabel@refs}{%
6034   \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
6035   \stepcounter{line}%
6036   \setcounter{absline}{\expandafter\labelrefsparseline\label@refs}%
6037   \stepcounter{absline}%
6038   \ifsublines@%
6039     \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
6040   %
6041     \stepcounter{subline}{1}%
6042   \def\label@refs{\theline|\thesubline|\theabsline}%
6043   \else%
6044     \def\label@refs{\theline|0|\theabsline}%
6045   \fi%
6046 }
6047 \def\labelrefsparseline#1|#2|#3{#1}%
6048 \def\labelrefsparsesubline#1|#2|#3{#2}%
6049 \def\labelrefsparselineabs#1|#2|#3{#3}%
6050 %

```

`\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.

```

6050 \newcommand*{\l@dmake@labels}{}
6051 \def\l@dmake@labels#1|#2|#3|#4|#5|#6{%
6052   \expandafter\ifx\csname the@label\csuse{XR@prefix}#6\endcsname%
6053   \relax%
6054   \else%
6055     \led@warn@DuplicateLabel{\csuse{XR@prefix}#6}%
6056   \fi
6057   \global\providetoggle{label@#6@ledRcol}%False is the default value of
this toggle, which tell us if a label is linked to a right or a left side
6058   \expandafter\gdef\csname the@label\csuse{XR@prefix}#6\endcsname
{#1|#2|#3|#4|#5|\relax}%
6059   \ignorespaces}
6060
6061 %

```

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.

```

6062 \AtBeginDocument{%
6063   \def\l@dmake@labels#1|#2|#3|#4|#5|#6{%
6064   }
6065
6066 %

```

\@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`.

```

6067
6068 \newcommand*{\@lab}{%
6069   \ifledRcol
6070     \xright@appenditem{\linenumr@p{\line@numR}}|%
        \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|\the\
6071     absline@numR}%
6072     \to\labelref@listR
6073   \else
6074     \xright@appenditem{\linenumr@p{\line@num}}|%

```

```

6075 \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|\the\absline@num}
%
6076 \to\labelref@list
6077 \fi}
6078 %

```

\apptlabel \apptlabel, if called in \edtext will insert automatically both a start and an end label for the current edtext lines.

```

6079 \newcommand*{\apptlabel}[1]{%
6080 \if@edtext@secondarg@%
6081 %

```

Label should not be already defined.

```

6082 \ifcsundef{the@label#1}{%
6083 \csdef{the@label#1}{\apptlabel}%
6084 }%
6085 {%
6086 \led@warn@DuplicateLabel{#1 (\apptlabel)}%
6087 }%
6088 %

```

Parse the \edtext line numbers.

```

6089 \expandafter\l@dp@rsefootsspec\l@d@nums|%
6090 %

```

Use the L^AT_EX standard hack for label.

```

6091 \@bsphack%
6092 %

```

And now, write the data in the auxiliary file.

```

6093 \ifledRcol%
6094 \protected@write\@auxout{%
6095 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\
6096 \l@dparsedstartline|\l@dparsedstartsub||\the\c@pstartR|{#1:start}}%
6097 \ifdef{\hypertarget}%
6098 {\Hy@raisedlink{\hypertarget{#1:start}}}%
6099 {\}}%
6100 \protected@write\@auxout{%
6101 {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline
6102 ||\l@dparsedendsub|\the\c@pstartR|{#1:end}}%
6103 \else%
6104 \ifl@dpairing%pstart or pstartL?
6105 \protected@write\@auxout{%
6106 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\
6107 \l@dparsedstartline|\l@dparsedstartsub||\the\c@pstartL|{#1:start}}%
6108 \ifdef{\hypertarget}%
6109 {\Hy@raisedlink{\hypertarget{#1:start}}}%
6110 {\}}%

```

```

6108         \protected@write\@auxout{}\%
6109         {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub||\the\c@pstartL|{\#1:end}}}%
6110     \else%
6111         \protected@write\@auxout{}\%
6112         {\string\l@dmake@labels\space\l@dparsedstartpage|\
l@dparsedstartline|\l@dparsedstartsub||\the\c@pstart|{\#1:start}}}%
6113         \ifdef{\hypertarget}%
6114             {\Hy@raisedlink{\hypertarget{\#1:start}}{}}}%
6115             {}%
6116         \protected@write\@auxout{}\%
6117         {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub||\the\c@pstart|{\#1:end}}}%
6118     \fi%
6119 \fi%
6120 %

```

Use the \TeX standard hack for label.

```

6121     \@esphack%
6122 %

```

Warning if `\applabel` is called outside of `\edtext`.

```

6123 \else%
6124     \led@warn@AppLabelOutSecondArgEdtext{\#1}%
6125 \fi%
6126 %

```

End of `\applabel`

```

6127 }%
6128 %

```

`\edlabelS` `\edlabelS` and `\edlabelE` are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
6129 \newcommand{\edlabelS}[1]{%
6130     \edlabel{\#1:start}%
6131 }
6132 \newcommand{\edlabelE}[1]{%
6133     \edlabel{\#1:end}%
6134 }
6135 \newcommand{\edlabelSE}[1]{%
6136     \edlabelS{\#1}%
6137     \edlabelE{\#1}%
6138 }
6139 %

```

`\wrap@edcrossref` `\wrap@edcrossref` is called around all `reledmac` crossref commands, except those which start with `x`. It adds the hyperlink.

```

6140 \newrobustcmd{\wrap@edcrossref}[2]{%
6141   \ifdef{\hyperlink}%
6142     {\hyperlink{#1}{#2}}%
6143     {#2}%
6144 }
6145 %

```

\edpageref If the specified label exists, \edpageref gives its page number.

\xpageref For this reference command, as for the other two, a special version with prefix x is provided for use in places where the command is to be scanned as a number, as in \linenum. These special versions have two limitations: they do not print error messages if the reference is unknown, and they can't appear as the first label or reference command in the file; you must ensure that a \edlabel or a normal reference command appears first, or these x-commands will always return zeros.

ℒ_{TEX} already defines a \pageref, so changing the name to \edpageref.

```

6146 \newcommand*\edpageref[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
6147 \newcommand*\xpageref[1]{\l@dgetref@num{1}{#1}}
6148
6149 %

```

\edlineref If the specified label exists, \lineref gives its line number.

\xlineref

```

6150 \newcommand*\edlineref[1]{%
6151   \l@dref@undefined{#1}%
6152   \wrap@edcrossref{#1}{%
6153     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
        label has not yet been parsed on the .aux file
6154     \iftoggle{label@#1@ledRcol}%
6155       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
6156       {\linenumrep{\l@dgetref@num{2}{#1}}}%
6157     \xflagref{#1}%
6158   }%
6159 }%
6160 \newcommand*\xlineref[1]{\l@dgetref@num{2}{#1}}%
6161
6162 %

```

\sublineref If the specified label exists, \sublineref gives its sub-line number.

\xsublineref

```

6163 \newcommand*\sublineref[1]{%
6164   \l@dref@undefined{#1}%
6165   \wrap@edcrossref{#1}{%
6166     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
        label has not yet been parsed on the .aux file
6167     \iftoggle{label@#1@ledRcol}%
6168       {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
6169       {\sublinenumrep{\l@dgetref@num{3}{#1}}}%

```

```

6170 }%
6171 }%
6172 \newcommand*\xsublineref}[1]{\l@getref@num{3}{#1}}
6173
6174 %

```

\xabslineref If the specified label exists, \xabslineref gives its absolute line number. That is used usually only by some reledmac internal macros.

```

6175 \newcommand*\xabslineref}[1]{\l@getref@num{4}{#1}}%
6176 %

```

\pstartref If the specified label exists, \pstartref gives its pstart number.

```

6177 \newcommand*\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@getref@num{5}{#1}}}%
6178 \newcommand*\xpstartref}[1]{\l@getref@num{5}{#1}}%
6179
6180 %

```

\xflagref \xflagref finds the side flag of any ref defined with \edlabel.

```

6181 \newcommand*\xflagref}[1]{\l@getref@num{6}{#1}}%
6182 %

```

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.

```

6183 \newcommand*\l@dref@undefined}[1]{%
6184   \expandafter\ifx\csname the@label#1\endcsname\relax
6185     \led@warn@RefUndefined{#1}%
6186   \fi}
6187
6188 %

```

\l@dgetref@num Next, \l@dgetref@num fetches the number we want. It has two arguments: the first is simply a digit, specifying whether to fetch a page (1), line (2), sub-line (3), (4) pstart number or (5) side flag. (This switching is done by calling \l@dlabel@parse.) The second argument is the label-macro, which because of the \@lab macro above is defined to be a string of the type 123|456|789.

```

6189 \newcommand*\l@dgetref@num}[2]{%
6190   \expandafter
6191   \ifx\csname the@label#2\endcsname \relax

```

```

6192     000%
6193     \else
6194         \expandafter\expandafter\expandafter
6195         \l@dlabel@parse\csname the@label#2\endcsname|#1%
6196     \fi}
6197
6198 %

```

\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.)

```

6199 \newcommand*\l@dlabel@parse{-}{}
6200 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7{%
6201     \ifcase #7%
6202         \or #1%
6203         \or #2%
6204         \or #3%
6205         \or #4%
6206         \or #5%
6207         \or #6%
6208     \fi}
6209 %

```

\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.

```

6210 \newcommand*\xxref}[2]{%
6211     {%
6212         \expandafter\ifx\csname the@label#1\endcsname \relax%
6213             \expandafter\let\csname the@@label#1\endcsname\zz@@@%
6214         \else%
6215             \expandafter\def\csname the@@label#1\endcsname{\l@dgetref@num
6216             {1}{#1}|\l@dgetref@num{2}{#1}|\l@dgetref@num{3}{#1}}%
6217             \fi%
6218             \expandafter\ifx\csname the@label#2\endcsname \relax%
6219                 \expandafter\let\csname the@@label#2\endcsname\zz@@@%
6220             \else%

```



```

6220 \expandafter\def\csname the@@label#2\endcsname{\l@dgetref@num
{1}{#2}|\l@dgetref@num{2}{#2}|\l@dgetref@num{3}{#2}}%
6221 \fi%
6222 \letcs{\@tempa}{the@@label#1}%
6223 \letcs{\@tempb}{the@@label#2}%
6224 \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@start{#1}}%
6225 \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@end{#2}}%
6226 \linenum{\@tempa|}%
6227 \@tempb}}}%
6228
6229 %

```

\appref \appref, \Seref, \apprefwithpage, \Serefwithpage and \SEonlypage print cross-ref to some start / end lines defined by specific commands. It prints the lines as they should be printed in the apparatus (critical notes for not suffixed versions, endnotes for suffixed versions).

\Serefwithpage Here we define hooks similar to some those related to critical footnotes or endnotes.

So, first declare the default value of the hooks for the pseudo-series. Also declare the internal toggle which are switch by reledmac.

```

6230 \def\Xtwolines@appref{}%
6231 \def\Xtwolines@Seref{}%
6232
6233 \def\Xmorethantwolines@appref{}%
6234 \def\Xmorethantwolines@Seref{}%
6235
6236 \def\Xlinerangeseparator@appref{\endashchar}%
6237 \def\Xlinerangeseparator@Seref{\endashchar}%
6238
6239 \def\Xsublinesep@appref{\fullstop}%
6240 \def\Xsublinesep@Seref{\fullstop}%
6241
6242 \def\Xpagelinesep@appref{\fullstop}%
6243 \def\Xpagelinesep@Seref{\fullstop}%
6244
6245
6246 \newtoggle{Xtwolinesbutnotmore@appref}%
6247 \newtoggle{Xtwolinesbutnotmore@Seref}%
6248
6249 \newtoggle{Xtwolinesonlyinsamepage@appref}%
6250
6251 \newtoggle{Xtwolinesonlyinsamepage@Seref}%
6252
6253 \newtoggle{Xlineflag@appref}%
6254 \toggletrue{Xlineflag@appref}%Here exception
6255 \newtoggle{Xlineflag@Seref}%
6256 \toggletrue{Xlineflag@Seref}%Here exception
6257
6258 \def\Xendtwolines@apprefwithpage{}%

```

```

6259 \def\Xendtwolines@SErefwithpage{}%
6260
6261 \def\Xendmoreethantwolines@apprefwithpage{}%
6262 \def\Xendmoreethantwolines@SErefwithpage{}%
6263
6264 \def\Xendlinerangeseparator@apprefwithpage{\endashchar}
6265 \def\Xendlinerangeseparator@SErefwithpage{\endashchar}
6266 \def\Xendlinerangeseparator@SErefonlypage{\endashchar}
6267
6268 \def\Xendbeforepagenumber@apprefwithpage{p.}%
6269 \def\Xendbeforepagenumber@SErefwithpage{p.}%
6270 \def\Xendbeforepagenumber@SEonlypage{p.}%
6271
6272 \def\Xendafterpagenumber@apprefwithpage{ }%
6273 \def\Xendafterpagenumber@SErefwithpage{ }%
6274
6275
6276 \def\Xendlineprefixsingle@apprefwithpage{}%
6277 \def\Xendlineprefixsingle@SErefwithpage{}%
6278
6279 \def\Xendlineprefixmore@apprefwithpage{}%
6280 \def\Xendlineprefixmore@SErefwithpage{}%
6281
6282 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
6283 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
6284
6285 \def\Xendsublinesep@apprefwithpage{\fullstop}%
6286 \def\Xendsublinesep@SErefwithpage{\fullstop}%
6287
6288 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
6289 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
6290
6291 \newtoggle{Xendlineflag@apprefwithpage}
6292 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
6293 \newtoggle{Xendlineflag@SErefwithpage}
6294 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
6295
6296 %

```

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.

```

6297
6298 \gdef\Xboxstartlinenum@appref{Opt}
6299 \gdef\Xboxstartlinenum@SEref{Opt}
6300
6301 \gdef\Xboxendlinenum@appref{Opt}
6302 \gdef\Xboxendlinenum@SEref{Opt}
6303
6304 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}

```

```

6305 \gdef\Xendboxstartlinenum@SErefwithpage{Opt}
6306
6307 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
6308 \gdef\Xendboxendlinenum@SErefwithpage{Opt}
6309
6310 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
6311 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
6312
6313 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
6314 \newtoggle{Xendpagenumberonlyfirstifsingle@SErefwithpage}
6315
6316 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
6317 \newtoggle{Xendpagenumberonlyfirstintwo@SErefwithpage}
6318
6319 \gdef\Xendsympagenum@apprefwithpage{}
6320 \gdef\Xendsympagenum@SErefwithpage{}
6321
6322 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
6323 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
6324
6325 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

6326 \newcommand\@apprefprefixsingle{}%
6327 \newcommand\@SErefprefixsingle{}%
6328
6329 \newcommand\@apprefprefixmore{}%
6330 \newcommand\@SErefprefixmore{}%
6331
6332 \newcommand{\setapprefprefixsingle}[1]{%
6333   \gdef\@apprefprefixsingle{#1}%
6334 }
6335 \newcommand{\setSErefprefixsingle}[1]{%
6336   \gdef\@SErefprefixsingle{#1}%
6337 }
6338
6339 \newcommand{\setapprefprefixmore}[1]{%
6340   \gdef\@apprefprefixmore{#1}%
6341 }
6342 \newcommand{\setSErefprefixmore}[1]{%
6343   \gdef\@SErefprefixmore{#1}%
6344 }
6345
6346 %

```

And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.

```

6347 \newcommand{\setSErefonlypageprefixsingle}[1]{%
6348   \gdef\SErefonlypage@prefixsingle{#1}%
6349 }%

```

```

6350 \newcommand{\setSErefonlypageprefixmore}[1]{%
6351   \gdef\SErefonlypage@prefixmore{#1}%
6352 }%
6353 %

```

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.

```

6354
6355 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
6356 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
6357
6358 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
6359 {#1}{#2}{appref}}
6360 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
6361 {#1}{#2}{SEref}}
6362
6363 \newcommand{\reformatted@}[3]{%
6364   \def\do##1{%
6365     \setkeys[mac]{truefootnoteoption}{##1}%
6366   }%
6367   \notblank{#1}{\docsvlist{#1}}{%}%
6368   \xdef\@currentseries{#3}%
6369   \ifcempty{@#3prefixmore}%
6370     {\@apprefprefixsingle}%
6371     {%
6372       \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}{%
6373         {\csuse{@#3prefixsingle}}%
6374         {\csuse{@#3prefixmore}}%
6375       }%
6376     \ifboolexpr{%
6377       test{\ifcsundef{the@label#2:start}}%
6378       or test{\ifcsundef{the@label#2:end}}%
6379     }%
6380     {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6381     {%
6382       \def\@this@crossref@start{#2:start}%
6383       \def\@this@crossref@end{#2:end}%
6384       \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
6385 start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
6386 xflagref{#2:start}}|
6387       \undef\@this@crossref@end%
6388       \undef\@this@crossref@start%
6389     }%
6390   \def\do##1{%
6391     \setkeys[mac]{falsefootnoteoption}{##1}%

```

```

6390 }%
6391 \notblank{#1}{\docsvlist{#1}}{}%
6392 }%
6393
6394 \newcommand{\reformattedwithpage@}[3]{%
6395   \def\do##1{%
6396     \setkeys[mac]{truefootnoteoption}{##1}%
6397   }%
6398   \notblank{#1}{\docsvlist{#1}}{}%
6399   \xdef\@currentseries{#3withpage}%
6400   \ifboolexpr{%
6401     test{\ifcsundef{the@label#2:start}}}%
6402     or test{\ifcsundef{the@label#2:end}}}%
6403   }%
6404   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6405   {%
6406     \def\@this@crossref@start{#2:start}%
6407     \def\@this@crossref@end{#2:end}%
6408     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|}%
6409     \undef\@this@crossref@end%
6410     \undef\@this@crossref@start%
6411   }%
6412   \def\do##1{%
6413     \setkeys[mac]{falsefootnoteoption}{##1}%
6414   }%
6415   \notblank{#1}{\docsvlist{#1}}{}%
6416 }%
6417
6418 \newcommand{\reformattedonlypage@}[3]{%
6419   \def\do##1{%
6420     \setkeys[mac]{truefootnoteoption}{##1}%
6421   }%
6422   \notblank{#1}{\docsvlist{#1}}{}%
6423   \xdef\@currentseries{#3onlypage}%
6424   \ifboolexpr{%
6425     test{\ifcsundef{the@label#2:start}}}%
6426     or test{\ifcsundef{the@label#2:end}}}%
6427   }%
6428   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6429   {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}}%
6430   {%
6431     \ifcsvoid{#3onlypage@prefixsingle}%
6432     {}%
6433     {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
}}}%
6434   \printnpnum{%
6435     \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6436   }%

```

```

6437 }%
6438 {%
6439 \ifcsvoid{#3onlypage@prefixmore}%
6440 {}%
6441 {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
6442 \ifdefined\linangesep@%
6443 \printnpnum{%
6444 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6445 \linangesep@%
6446 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
6447 }%
6448 \else%
6449 \printnpnum{%
6450 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6451 \csuse{Xendlinangeseparator@}\currentseries%
6452 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
6453 }%
6454 \fi%
6455 }%
6456 }%
6457 \def\do##1{%
6458 \setkeys[mac]{falsefootnoteoption}{##1}%
6459 }%
6460 \notblank{#1}{\docsvlist{#1}}{%}%
6461 }%
6462 %

```

`\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`.

```

6463 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\
6464 endcsname{#2}}
6465 %

```

(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. 133 and V.9 p. 100), 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.

```
6466 \pretocmd{\XR@test}%
6467   {\XR@test@mac+++#1#2#3#4+++}%
6468   {}%
6469   {}%
6470   %
```

\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.

```
6471 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
6472 %
```

\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.

```
6473 \long\def\xR@test@mac@test#1#2...{%The triple dots (NOT \ldots) are because
6474   of the line 22 of xr.sty v5.02 1994/05/28
6475   \ifx#1\l@dmake@labels%
6476     \l@dmake@labels#2%
6477   \else
6478     \ifx#1\l@dmake@labelsR%
6479       \l@dmake@labelsR #2%
6480     \fi%
6481   \fi%
6482   }%
6483   %
```

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.

```
6483 \pretocmd{\@xympar}%
6484   {\ifnumberedpar@
6485     \led@warn@NoMarginpars
6486     \@esphack
6487   \else}%
6488   {}%
6489   {}%
6490
6491 \apptocmd{\@xympar}%
6492   {\fi}%
```

```

6493 {}
6494 {}
6495
6496 %

```

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

```

6497 \newcount\sidenote@margin
6498 \newcommand*{\sidenotemargin}[1]{\%
6499   \l@dgetsidenote@margin{#1}%
6500   \ifnum\@l@dttempcntb>\m@ne
6501     \ifledRcol
6502       \global\sidenote@marginR=\@l@dttempcntb
6503     \else
6504       \global\sidenote@margin=\@l@dttempcntb
6505     \fi
6506   \fi}}
6507 \newcommand*{\l@dgetsidenote@margin}[1]{\%
6508   \def\@tempa{#1}\def\@tempb{left}%
6509   \ifx\@tempa\@tempb
6510     \@l@dttempcntb \z@
6511   \else
6512     \def\@tempb{right}%
6513     \ifx\@tempa\@tempb
6514       \@l@dttempcntb \@ne
6515     \else
6516       \def\@tempb{outer}%
6517       \ifx\@tempa\@tempb
6518         \@l@dttempcntb \tw@
6519       \else
6520         \def\@tempb{inner}%
6521         \ifx\@tempa\@tempb
6522           \@l@dttempcntb \thr@@
6523         \else
6524           \led@warn@BadSidenotemargin
6525           \@l@dttempcntb \m@ne
6526         \fi

```



```

6527 \fi
6528 \fi
6529 \fi}
6530 \sidenotemargin{right}
6531
6532 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
6533 \newbox\l@dlp@rbox
6534 \newbox\l@drp@rbox
6535
6536 %

```

`\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
6537 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
6538 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
6539 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup
6540 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
6541 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
6542 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
6543
6544 %

```

`\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
6545 \newcommand*\ledleftnote[1]{\edtext{}\l@dlsnote{#1}}
6546 \newcommand*\ledrightnote[1]{\edtext{}\l@drsnote{#1}}
6547 \newcommand*\ledsidenote[1]{\edtext{}\l@dcnote{#1}}%
6548 \newcommand*\ledinnernote[1]{\edtext{}\l@disnote{#1}}%
6549 \newcommand*\ledouternote[1]{\edtext{}\l@dosnote{#1}}%
6550 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent
`\l@drsnote` of the critical footnotes code.

```

\l@dcnote
6551 \newif\ifrighnoteup
\l@desnote
6552 \righnoteuptrue
\l@disnote
6553
6554 \newcommand*\l@dlsnote[1]{%
6555 \begingroup%
6556 \newcommand{\content}{#1}%
6557 \ifnumberedpar@
6558 \ifledRcol%
6559 \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%

```

```

6560             \to\inserts@listR
6561     \global\advance\insert@countR \@ne%
6562 \else%
6563     \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}}%
6564             \to\inserts@list
6565     \global\advance\insert@count \@ne%
6566 \fi
6567 \fi%
6568 \ignorespaces%
6569 \endgroup%
6570 }%
6571
6572 \newcommand*{\l@drsnote}[1]{%
6573     \begingroup%
6574     \newcommand{\content}{#1}%
6575     \ifnumberedpar@
6576     \ifledRcol%
6577         \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
6578                 \to\inserts@listR
6579         \global\advance\insert@countR \@ne%
6580     \else%
6581         \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
6582                 \to\inserts@list
6583         \global\advance\insert@count \@ne%
6584     \fi
6585 \fi\ignorespaces%
6586 \endgroup%
6587 }%
6588
6589 \newcommand*{\l@dcnote}[1]{%
6590     \begingroup%
6591     \newcommand{\content}{#1}%
6592     \ifnumberedpar@
6593     \ifledRcol%
6594         \xright@appenditem{\noexpand\vl@dcnote{\expandonce\content}}}%
6595                 \to\inserts@listR
6596         \global\advance\insert@countR \@ne%
6597     \else%
6598         \xright@appenditem{\noexpand\vl@dcnote{\expandonce\content}}}%
6599                 \to\inserts@list
6600         \global\advance\insert@count \@ne%
6601     \fi
6602 \fi\ignorespaces%
6603 \endgroup%
6604 }%
6605
6606 \newcommand*{\l@disnote}[1]{%
6607     \begingroup%
6608     \newcommand{\content}{#1}%
6609     \ifnumberedpar@%

```

```

6610 \ifledRcol%
6611 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
6612 \to\inserts@listR%
6613 \global\advance\insert@countR \@ne%
6614 \else%
6615 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
6616 \to\inserts@list%
6617 \global\advance\insert@count \@ne%
6618 \fi%
6619 \fi\ignorespaces%
6620 \endgroup%
6621 }%
6622
6623 \newcommand*{\l@dosnote}[1]{%
6624 \begingroup%
6625 \newcommand{\content}{#1}%
6626 \ifnumberedpar%
6627 \ifledRcol%
6628 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
6629 \to\inserts@listR%
6630 \global\advance\insert@countR \@ne%
6631 \else%
6632 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
6633 \to\inserts@list%
6634 \global\advance\insert@count \@ne%
6635 \fi%
6636 \fi\ignorespaces%
6637 \endgroup%
6638 }%
6639
6640 %

```

\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.

```

6641 \newcommand*{\vl@dlsnote}[1]{%
6642   \ifledRcol{%
6643     \@l@dttempcntb=\sidenote@marginR%
6644     \ifnum\@l@dttempcntb>\@ne%
6645       \advance\@l@dttempcntb by\page@numR%
6646     \fi%
6647   \else%
6648     \@l@dttempcntb=\sidenote@margin%
6649     \ifnum\@l@dttempcntb>\@ne%
6650       \advance\@l@dttempcntb by\page@num%
6651     \fi%
6652   \fi%
6653   \ifodd\@l@dttempcntb%
6654     \listgadd{\l@dcsnotetext@l}{#1}%
6655   \else%
6656     \listgadd{\l@dcsnotetext}{#1}%
6657   \fi
6658 }
6659 \newcommand*{\vl@drsnote}[1]{%
6660   \ifledRcol{%
6661     \@l@dttempcntb=\sidenote@marginR%
6662     \ifnum\@l@dttempcntb>\@ne%
6663       \advance\@l@dttempcntb by\page@numR%
6664     \fi%
6665   \else%
6666     \@l@dttempcntb=\sidenote@margin%
6667     \ifnum\@l@dttempcntb>\@ne%
6668       \advance\@l@dttempcntb by\page@num%
6669     \fi%
6670   \fi%
6671   \ifodd\@l@dttempcntb%
6672     \listgadd{\l@dcsnotetext}{#1}%
6673   \else%
6674     \listgadd{\l@dcsnotetext@r}{#1}%
6675   \fi%
6676 }
6677 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
6678
6679 \newcommand{\vl@disnote}[1]{%
6680   \ifledRcol{%
6681     \@tempcnta=\page@numR%
6682   \else%
6683     \@tempcnta=\page@num%
6684   \fi%
6685   \ifodd\@tempcnta% ODD => right page => inner side = left side
6686     \vl@dlsnote{#1}%
6687   \else%
```

```

6688 \vl@drsnote{#1}%
6689 \fi%
6690 }%
6691
6692 \newcommand{\vl@dosnote}[1]{%
6693 \ifledRcol%
6694 \@tempcnta=\page@numR%
6695 \else%
6696 \@tempcnta=\page@num%
6697 \fi%
6698 \ifodd\@tempcnta% ODD => right page => outer side = right side
6699 \vl@drsnote{#1}%
6700 \else%
6701 \vl@dlsnote{#1}%
6702 \fi%
6703 }%
6704
6705 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box. `\setl@drpr@box` And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

6706 \newcommand*\setl@dlp@rbox}[1]{%
6707 \begingroup%
6708 \parindent\z@\hsize=\ledlsnotewidth%
6709 \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
6710 \global\setbox\l@dlp@rbox%
6711 \ifleftnoteup%
6712 =\vbox to\z@{\ledlsnotefontsetup\vss #1}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
6713 \else%
6714 =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}%
6715 \fi%
6716 \endgroup%
6717 }
6718
6719 \newcommand*\setl@drp@rbox}[1]{%
6720 \begingroup%
6721 \parindent\z@\hsize=\ledrsnotewidth%
6722 \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
6723 \global\setbox\l@drp@rbox%
6724 \ifrighnoteup%
6725 =\vbox to\z@{\ledrsnotefontsetup\vss#1}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
6726 \else%

```

```

6727     =\vbox to0.7\baselineskip{{\ledrsnotefontsetup\strut#1\vss}}}%
6728     \fi%
6729     \endgroup%
6730 }%
6731 \newif\ifleftnoteup
6732 \leftnoteuptrue
6733 %

```

\@sidenotesep This macro is used to separate sidenotes of the same line.

```

6734 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
6735 \newcommand{\@sidenotesep}{, }
6736 %

```

\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.

```

6737 \newcommand*{\affixside@note}{%
6738     \prepare@edindex@fornote{\the\page@num|\the\line@num|\the\subline@num|\the\page@num|\the\line@num|\the\subline@num|}%
6739     \def\sidenotecontent@{}%
6740     \numgdef\itemcount@{0}%
6741     \def\do##1{%
6742         \ifnumequal{\itemcount@}{0}%
6743             {%
6744                 \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
6745             {\appto\sidenotecontent@{\@sidenotesep ##1}}%
6746             }%
6747         \numgdef\itemcount@{\itemcount@+\@ne}%
6748     }%
6749     \dolistloop{\l@dcstotetext}%
6750     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
6751 %

```

And we do the same for left and right notes (not movable).

```

6752 \gdef\@templ@d{%
6753 \gdef\@templ@n{\l@dcstotetext\l@dcstotetext@l\l@dcstotetext@r}%
6754 \ifx\@templ@d\@templ@n \else%
6755     \if@twocolumn%
6756         \if@firstcolumn%
6757             \setl@dlp@rbox{##1}{\sidenotecontent@}%
6758         \else%
6759             \setl@drp@rbox{\sidenotecontent@}%
6760         \fi%
6761     \else%

```

```

6762 \l@dttempcntb=\sidenote@margin%
6763 \ifnum\l@dttempcntb>\@ne%
6764 \advance\l@dttempcntb by\page@num%
6765 \fi%
6766 \ifodd\l@dttempcntb%
6767 \setl@drp@rbox{\sidenotecontent@}%
6768 \gdef\sidenotecontent@{}%
6769 \numgdef{\itemcount@}{0}%
6770 \dolistloop{\l@dcstetext@l}%
6771 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
6772 \setl@dlp@rbox{\sidenotecontent@}%
6773 \else%
6774 \setl@dlp@rbox{\sidenotecontent@}%
6775 \gdef\sidenotecontent@{}%
6776 \numgdef{\itemcount@}{0}%
6777 \dolistloop{\l@dcstetext@r}%
6778 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
6779 \setl@drp@rbox{\sidenotecontent@}%
6780 \fi%
6781 \fi%
6782 \fi%
6783 \advance\@edindex@fornote@m@ne%
6784 }
6785 %

```

XXV Minipages and such

We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the `\@iiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfeetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage`.
`\l@dfeetendmini` They can be extended to handle other things if necessary.

```

6786 \ifnoledgroup@\else%
6787 \newcommand*{\l@dfeetbeginmini}{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
6788 \newcommand*{\l@dfeetendmini}{%
6789 \IfStrEq{critical-familiar}{\@mpfnpos}%
6790 {\l@dedendmini\l@dfamendmini}%
6791 {%
6792 \IfStrEq{familiar-critical}{\@mpfnpos}%
6793 {\l@dfamendmini\l@dedendmini}%
6794 {\do@feet@custom@order{mp@}{\@mpfnpos}}%
6795 }%
6796 }%
6797 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

`\l@dedendmini`

`\mp@append@Xnotes`

```

6798 \newcommand*{\l@dedbeginmini}{%
6799   \unless\ifnocritical@%
6800   \def\do##1{%
6801     \csletcs{v##1footnote}{mpv##1footnote}%
6802   }%
6803   \dolistloop{\@series}%
6804   \fi%
6805 }
6806 \newcommand*{\l@dedendmini}{%
6807   \unless\ifnocritical@%
6808   \ifl@dpairing%
6809     \ifledRcol%
6810       \flush@notesR%
6811     \else%
6812       \flush@notes%
6813     \fi%
6814   \fi
6815   \def\do##1{%
6816     \mp@append@Xnotes{##1}%
6817   }%
6818   \dolistloop{\@series}%
6819   \fi%
6820 }%
6821 \newcommand{\mp@append@Xnotes}[1]{%
6822   \ifvoid\csuse{mp#1footins}\else%
6823     \ifl@dpairing%
6824       \ifparledgroup%
6825         \ifledRcol%
6826           \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR\skip
\@nameuse{mp#1footins}}%
6827         \else%
6828           \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL\
skip\@nameuse{mp#1footins}}%
6829         \fi%
6830       \fi%
6831     \fi%
6832     \ifcsstring{series@display#1}{paragraph}{%
6833       \setbox\@nameuse{mp#1footins}=\vbox{%
6834         \csuse{Xnotefontsize@#1}%
6835         \ifcsdef{Xhsize}\csuse{series@display#1}@#1{%
6836           \hsize\csuse{Xhsize}\csuse{series@display#1}@#1}%
6837         }{%
6838           \noindent\csuse{Xtxtbeforenotes@#1}%
6839           \unvbox\@nameuse{mp#1footins}%
6840           \@parboxrestore%
6841         }%
6842       }%
6843       \csuse{mp#1footgroup}{#1}%

```



```

6844 \fi%
6845 }%
6846 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
6847 \newcommand*{\l@dfambeginmini}{%
6848 \unless\ifnofamiliar%
6849 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
6850 \dolistloop{\@series}%
6851 \fi%
6852 }%
6853
6854 \newcommand*{\l@dfamendmini}{%
6855 \unless\ifnofamiliar%
6856 \def\do##1{%
6857 \mp@append@notesX{##1}%
6858 }%
6859 \dolistloop{\@series}%
6860 \fi%
6861 }%
6862 \newcommand{\mp@append@notesX}[1]{%
6863 \ifvoid\csuse{mpfootins#1}\else%
6864 \ifcsstring{series@displayX#1}{paragraph}{%
6865 \setbox\@nameuse{mpfootins#1}=\vbox{%
6866 \csuse{notefontsizeX#1}%
6867 \ifcsdef{hsize\csuse{series@display#1}X#1}{%
6868 \hsize\csuse{hsize\csuse{series@display#1}X#1}%
6869 }{}%
6870 \noindent\csuse{txtbeforenotesX#1}%
6871 \unvbox\@nameuse{mpfootins#1}%
6872 \@parboxrestore%
6873 }%
6874 }%
6875 \csuse{mpfootgroup#1}{#1}%
6876 \fi%
6877 }%
6878 %

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

6879 \patchcmd%
6880 {\@iiiminipage}%
6881 {\let\@footnotetext\@mpfootnotetext}%
6882 {\let\@footnotetext\@mpfootnotetext\l@dfetbeginmini}%
6883 {}%
6884 {\led@error@fail@patch@iiiminipage}%
6885 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

6886 \patchcmd%
6887   {\endminipage}%
6888   {\footnoterule}%
6889   {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
6890   {}%
6891   {\led@error@fail@patch@endminipage}
6892
6893 \patchcmd%
6894   {\endminipage}%
6895   {\@minipagefalse}%
6896   {\l@dfeetendmini\@minipagefalse}%
6897   {}%
6898   {\led@error@fail@patch@endminipage}
6899
6900 %

```

`\l@dunboxmpfoot` `\@ldunboxmpfoot` insert normal footnotes for ledgroup.
`\l@advance@parledgroup@beforenormalnotes`

```

6901 \newcommand*{\l@dunboxmpfoot}{%
6902   \vskip\skip\@mpfootins
6903   \normalcolor
6904   \footnoterule
6905   \l@advance@parledgroup@beforenormalnotes
6906   \unvbox\@mpfootins%
6907 }
6908 %

```

When using parallel ledgroup, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

6909 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
6910   \ifparledgroup
6911     \ifl@dpairing
6912       \ifledRcol
6913         \dimgdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR\
skip\@mpfootins}
6914       \else
6915         \dimgdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL\
skip\@mpfootins}
6916       \fi
6917     \fi
6918   \fi
6919 }
6920 %

```

`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.

```

6921

```

```

6922 \newenvironment{ledgroup}{%
6923   \resetprevpage@num%
6924   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
6925   \let\@footnotetext\@mpfootnotetext
6926   \l@dfetbeginmini%
6927 }{%
6928   \par
6929   \unskip
6930   \ifvoid\@mpfootins\else
6931     \l@duboxmpfoot
6932   \fi
6933   \l@dfetendmini%
6934   \@ledgroupfalse%
6935 }
6936
6937
6938 %

```

`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.

```

6939 \newenvironment{ledgroupsize}[2][1]{%
6940 %

```

Set the various text measures.

```

6941 \hspace #2\relax
6942 %

```

Initialize fills for centering.

```

6943 \let\ledllfill\hfil
6944 \let\ledrlfill\hfil
6945 \def\@tempa{#1}\def\@tempb{1}%
6946 %

```

Left adjusted numbered lines

```

6947 \ifx\@tempa\@tempb
6948 \let\ledllfill\relax
6949 \else
6950 \def\@tempb{r}%
6951 \ifx\@tempa\@tempb
6952 %

```

Right adjusted numbered lines

```

6953 \let\ledrlfill\relax
6954 \fi
6955 \fi
6956 %

```

Set up the footnoting.

```

6957 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
6958 \let\@footnotetext\@mpfootnotetext
6959 \l@dfetbeginmini%
6960 }f%
6961 \par
6962 \unskip
6963 \ifvoid\@mpfootins\else
6964   \l@dunboxmpfoot
6965 \fi
6966 \l@dfetendmini%
6967 }
6968
6969 %

```

Close the \ifnoledgroup@else.

```

6970 \fi%
6971 %

```

`\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`.

```

6972 \newif\ifledgroupnotesL@
6973 \newif\ifledgroupnotesR@
6974 %

```

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`.

```

6975 \AtBeginDocument{%
6976   \unless\ifl@imakeidx%
6977     \@ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
6978   %
6979   \fi%
6980   \unless\ifl@indextools%
6981     \@ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}
6982   %
6983   \fi%
6984   \unless\ifl@footmisc%
6985     \@ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
6986   %
6987   \fi%
6988 }
6989 %

```

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
6987 \newcommand{\pagelinesep}{-}
6988 \newcommand{\edindexlab}{$&}
6989 \newcounter{labidx}
6990 \setcounter{labidx}{0}
6991
6992 %
```

`\doedindexlabel` This macro sets an `\edlabel`.

```
6993 \newcommand{\doedindexlabel}{%
6994   \stepcounter{labidx}%
6995   \edlabel{\edindexlab\thelabidx}%
6996 }
6997
6998 %
```

`\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`.

```
6999 \newcounter{pageline}%
7000 \renewcommand{\thepageline}{%
7001   \thepage%
7002   \pagelinesep%
7003   \xlineref{\edindexlab\thelabidx}%
7004 }
7005 %
```

`\thestartpageline` These macros make up the page/line start/end number when the `\edindex` command
`\theendpageline` is called in critical notes.

```
7006 \newcommand{\thestartpageline}{%
7007   \l@dparsedstartpage%
7008   \pagelinesep%
7009   \l@dparsedstartline%
7010 }
7011 \newcommand{\theendpageline}{%
7012   \l@dparsedendpage%
7013   \pagelinesep%
7014   \l@dparsedendline%
7015 }
7016 %
```

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.

```
7017 \newcount\@edindex@fornote@
7018 %
```

`\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 |.

```
7019 \newcommand{\prepare@edindex@fornote}[1]{%
7020   \l@dp@rsefootspec#1|}%
7021   \advance\@edindex@fornote@\@ne%
7022 }
7023 %
```

`\get@edindex@ledinnote@command` The `\get@edindex@ledinnote@command` macro defines a `\@ledinnote@command` command which is added as an attribute (text inserted after |) of the next index entry. Consequently, we write the definition of the location reference attribute in the .xdy file.

```
7024 \newcommand{\get@edindex@ledinnote@command}{%
7025   \ifxindy%
7026     \gdef\@ledinnote@command{%
7027       ledinnote\thelabidx%
7028     }%
7029     \ifxindyhyperref%
7030       \immediate\write\eledmac@xindy@out{%
7031         (define-attributes ("ledinnote\thelabidx"))^^J
7032         \space\space(markup-locref^^J
7033         \eledmacmarkuplocrefdepth^^J
7034         :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
7035       }{"^^J
7036         :close "}"^^J
7037         :attr "ledinnote\thelabidx"^^J
7038       )
7039     }%
7040   \else%
7041     \immediate\write\eledmac@xindy@out{%
7042       (define-attributes ("ledinnote\thelabidx"))^^J
7043       \space\space(markup-locref^^J
7044       \eledmacmarkuplocrefdepth^^J
7045       :open "\string\ledinnote{\@index@command}{"^^J
7046       :close "}"^^J
7047       :attr "ledinnote\thelabidx"^^J
7048     )
7049   }
```

```

7048 }%
7049 \fi%
7050 %

```

If we do not use xindy option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

7051 \else%
7052 \gdef\@ledinnote@command{%
7053   ledinnote[\edindexlab\thelabidx]{\@index@command}%
7054 }%
7055 \fi%
7056 }
7057 %

```

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 |.

```

7058 \def\get@index@command#1|#2+{%
7059   \gdef\@index@txt{#1}%
7060   \gdef\@index@command{#2}%
7061   \xdef\@index@parenthesis{}%
7062   \IfBeginWith{\@index@command}{(}{%
7063     \StrGobbleLeft{\@index@command}{1}{\@index@command@}%
7064     \global\let\@index@command\@index@command@%
7065     \xdef\@index@parenthesis{(}%
7066   }{)%
7067   \IfBeginWith{\@index@command}{)}{%
7068     \StrGobbleLeft{\@index@command}{1}{\@index@command@}%
7069     \global\let\@index@command\@index@command@%
7070     \xdef\@index@parenthesis{)}%
7071   }{)%
7072 }
7073 %

```

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.

```

7074 \newcommandx{\ledinnote}[3][1,usedefault]{%
7075   \ifboolexpr{%
7076     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
7077     or%
7078     bool {xindyhyperref@}%
7079   }%
7080   {%
7081     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%

```

```

7082 }%
7083 {%
7084 \csuse{#2}{\ledinnotemark{#3}}%
7085 }%
7086 }%
7087 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage
7088 {#2}}}}%
7089 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
7089 %

```

XXVI.6 Main code

Eledmac and ledmac were using the specific indexing tools of the memoir in order to allow multiple index. However, eledmac used imakeidx or indextools tools when one these two package was loaded. This system forced to maintained a double code, which was not very useful. Since reledmac, we use only the imakeidx or indextools tools.

The memoir class provides more flexible indexing than the standard classes. We need different code if the memoir class is being used, except if imakeidx or indextools is used.

```

\edindex Write the index information to the idx file.
\@wredindex
7090 \dummy@edindex \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
7091 \begingroup%
7092 \let\emph\@firstofone%
7093 \let\textbf\@firstofone%
7094 \let\textit\@firstofone%
7095 \let\textmd\@firstofone%
7096 \let\textnormal\@firstofone%
7097 \let\textrm\@firstofone%
7098 \let\textsc\@firstofone%
7099 \let\textsf\@firstofone%
7100 \let\textsl\@firstofone%
7101 \let\texttt\@firstofone%
7102 \let\textup\@firstofone%
7103 \xdef\@tmp{#2}%To be used in IfSubStr instead of #2 directly. Avoid
some expansion bugs (for example with \edindex{textsc{something}})
7104 \endgroup%
7105 \ifl@imakeidx%
7106 \ifnum\@edindex@fornote@>\z@%
7107 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
7108 \get@edindex@ledinnote@command%
7109 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
7110 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%

```



```

7111 \else%
7112 \get@edindex@hyperref{#2}%
7113 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
7114 \fi%
7115 \else%
7116 \ifnum\@edindex@fornote@>\z@%
7117 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}%
7118 %
7119 \get@edindex@ledinnote@command%
7120 \expandafter\protected@write\@indexfile{}%
7121 {\string\indexentry{\@index@txt|(\@ledinnote@command){\thestartpageline}
7122 }%
7123 \expandafter\protected@write\@indexfile{}%
7124 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
7125 }%
7126 \else%
7127 \protected@write\@indexfile{}%
7128 {\string\indexentry{#2}{\thepageline}
7129 }%
7130 \fi%
7131 \endgroup
7132 \@esphack%
7133 }
7134 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

7135 \pretocmd{\makeindex}{%
7136 \def\edindex{%
7137 \ifbool{numbering} or bool{numberingR} or bool{
7138 l@printingpages} or bool{l@printingcolumns}}{%
7139 \bsphack%
7140 \doedindexlabel%
7141 \begin@group%
7142 \sanitizet%
7143 \wredindex%
7144 }%
7145 {%
7146 \led@warn@edinde@outsidenumbering%
7147 \index%
7148 }%
7149 }%
7150 }%
7151 {\led@error@fail@patch@makeindex}%
7152 \newcommand{\edindex}[1]{\bsphack\esphack}
7153 \newcommandx{\dummy@edindex}[2][1=\expandonce\jobname,usedefault]{}%
7154 %

```

XXVI.7 Hyperlink

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```
7155 \newcommand{\hyperlinkformat}[3]{%
7156   \ifstrempy{#1}%
7157   {\hyperlink{#2}{#3}}%
7158   {\csuse{#1}{\hyperlink{#2}{#3}}}%
7159 }%
7160 %
```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```
7161 \newcommand{\hyperlinkR}[2]{%
7162   \hyperlink{#1}{#2\@Rlineflag}%
7163 }%
7164 %
7165 %
```

`\hyperlinkformatR` `\hyperlinkformatR` command is to be used to create a internal hyperlink, a format and a `\@Rlineflag`, when indexing.

```
7166 \newcommand{\hyperlinkformatR}[3]{%
7167   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
7168 }%
7169 %
7170 %
```

`\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`).

```
7171 \newcommand{\get@edindex@hyperref}[1]{%
7172 %
```

We have to disable temporary spaces to work through a `xstring` bug (or feature?)

```
7173 \edef\temp@{%
7174   \catcode`\ =9 %space need for catcode
7175   \detokenize{#1}%For active character in unicode
7176   \catcode`\ =10 % space need for catcode
7177 }%
7178 %
```

Now, we define `\@edindex@hyperref` if the hyperindex of `hyperref` is enabled.

```
7179 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
7180   \IfSubStr{\temp@}{|}%
7181   {\get@index@command#1+%
7182   \ifledRcol%
7183     \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
```

```

7184     hyperlinkformatR{\@index@command}%
7185     {\edindexlab\thelabidx}}%
7186   \else%
7187     \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7188     hyperlinkformat{\@index@command}%
7189     {\edindexlab\thelabidx}}%
7190   \fi%
7191 }%
7192 {\get@index@command#1|+%
7193 \ifledRcol%
7194 \gdef\@edindex@hyperref{|\hyperlinkR{\edindexlab\thelabidx}}%
7195 \else%
7196 \gdef\@edindex@hyperref{|\hyperlink{\edindexlab\thelabidx}}%
7197 \fi%
7198 }%
7199 }%
7200 %

7201 % If we use both xindy and hyperref, first get the \protect\cs{
7202 index@command} command.
7203 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
7204 % \begin{macrocode}
7205 {\ifxindyhyperref@%
7206 \IfSubStr{\temp@}{|}%
7207 {\get@index@command#1|+%
7208 {\get@index@command#1|+%
7209 \gdef\@edindex@hyperref{|\eledmac\thelabidx}}%
7210 }%
7211 }%

```

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`.

```

7210 \IfStrEq{\@index@parenthesis}{(}%
7211 {%
7212 \csxdef{xindy@parenthesis@\@index@txt}{\thelabidx}%
7213 \gdef\@edindex@hyperref{|\eledmac\thelabidx}%
7214 }%
7215 {}%
7216 %

```

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`.

```

7217 \IfStrEq{\@index@parenthesis}{)}%
7218 {%
7219 \xdef\@edindex@hyperref{|\eledmac\csuse{xindy@parenthesis@\@index@txt}}%
7220 \global\csundef{xindy@parenthesis@\@index@txt}%
7221 }%
7222 %

```

Write in the .xdy file the attributes of the location.

```

7223     {%
7224     \immediate\write\eledmac@xindy@out{%
7225     (define-attributes ("eledmac\thelabidx"))^^J
7226     \space\space(markup-locoref^^J
7227     \eledmacmarkuplocorefdepth^^J
7228     :open "\string\hyperlink%
7229           \ifledRcol R\fi%
7230           {\edindexlab\thelabidx}%
7231           {\ifdefempty{\@index@command}%
7232           {}%
7233           {\@backslashchar\@index@command}%
7234           {"^^J
7235     :close "}}^^J
7236     :attr "eledmac\thelabidx"^^J
7237     )
7238     }%
7239     }%
7240     %

```

And now, in any other case.

```

7241     \else%
7242     \gdef\@index@txt{#1}%
7243     \gdef\edindex@hyperref{}%
7244     \fi%
7245     }%
7246   }
7247   %

```

XXVI.8 ‘innote’ and ‘notenumber’ option of indextols package

\led@set@index@fornote The `\led@set@index@fornote` is called when a familiar footnote is inserted — and not when it is read — and changes the `\index` command depending of the option of the `indextools` package. Its only argument is the note series.

```

7248 \newcommand{\led@set@index@fornote}[1]{%
7249   \ifbool{indtl@innote}%
7250   {\let\index\nindex}%
7251   {}%
7252   \ifbool{indtl@notenumber}%
7253   {%
7254     \renewcommand{\index}[2][\indtl@jobname]{%
7255       \orig@@index[##1]{%
7256         ##2|innotenumber{\this@footnoteX@reading}%
7257       }%
7258     }%
7259   }%
7260   {}%

```

```

7261 }%
7262 %

```

`\led@reinit@index@fornote` The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

7263 \newcommand{\led@reinit@index@fornote}{%
7264   \ifbool{indtl@innote}%
7265     {\let\index\orig@index}%
7266     {}%
7267   \ifbool{indtl@notenumber}%
7268     {\let\index\orig@index}%
7269     {}%
7270 }%
7271 %

```

XXVII Glossaries

Here, we define the `\gls`-like commands prefixed by `ed`, only if the package `glossaries` is loaded.

```

7272 \AtBeginDocument{%
7273   \@ifpackageloaded{glossaries}{%
7274     %

```

First those which arguments are `[<options>]{<label>}[<insert>]`.

```

7275   \renewcommand{\do}[1]{%
7276     \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,3,
usedefault]{%
7277       \doedindexlabel%
7278       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
7279     }%
7280     \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,3,usedefault]{%
7281       \doedindexlabel%
7282       \csname#1\endcsname*[counter=pageline,##1]{##2}[##3]%
7283     }%
7284   }%
7285   \docsvlist{%
7286     gls,%
7287     Gls,%
7288     GLS,%
7289     glspl,%
7290     Glspl,%
7291     GLSpl,%
7292     glstext,%
7293     Glstext,%
7294     GLStext,%
7295     Glsfirst,%
7296     GLSfirst,%

```

```

7297     glsplural%
7298     Glsplural,%
7299     GLSplural,%
7300     glsfirstplural,%
7301     Glsfirstplural,%
7302     GLSfirstplural,%
7303     glsname,%
7304     Glsname,%
7305     GLSname,%
7306     glssymbol,%
7307     Glsymbol,%
7308     GLSsymbol,%
7309     glsdesc,%
7310     Glsdesc,%
7311     GLSdesc,%
7312     glsuseri,%
7313     Glsuseri,%
7314     GLSuseri,%
7315     glsuserii,%
7316     Glsuserii,%
7317     GLSuserii,%
7318     glsuseriii,%
7319     Glsuseriii,%
7320     GLSuseriii,%
7321     glsuseriv,%
7322     Glsuseriv,%
7323     GLSuseriv,%
7324     glsuserv,%
7325     Glsuserv,%
7326     GLSuserv,%
7327     glsuservi,%
7328     Glsuservi,%
7329     GLSuservi%
7330     }%
7331 %

```

Then those which arguments are [*options*]{*label*}{*link text*}.

```

7332     \renewcommand{\do}[1]{%
7333     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,
usedefault]{%
7334     \doedindexlabel%
7335     \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
7336     }%
7337     \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,usedefault]{%
7338     \doedindexlabel%
7339     \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
7340     }%
7341     }%
7342     \docsvlist{glsdisp,glslink}%

```

```

7343 %
    Then those which arguments are [options]{label}.
7344 \renewcommand{\do}[1]{%
7345   \expandafter\DeclareRobustCommand\csname ed#1\endcsname[2][1,
usedefault]{%
7346     \doedindexlabel%
7347     \csname#1\endcsname[counter=pageline,##1]{##2}%
7348   }%
7349   \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[2][1,usedefault]{%
7350     \doedindexlabel%
7351     \csname#1\endcsname*[counter=pageline,##1]{##2}%
7352   }%
7353 }%
7354 \docsvlist{glsadd}%
7355 }{}%
7356 }%
7357 %

```

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.

```

7358 \def\@hangingsymbol{}
7359 \newcommand*\sethangingsymbol[1]{%
7360   \gdef\@hangingsymbol{#1}%
7361 }%
7362 \newif\ifinstanza
7363 %

```

`\inserthangingsymbol` The boolean `\inserthangingsymbol` 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 `\inserthangingsymbol` is made in `\do@line` before the printing of line but after the line number calculation.

```

7364 \newif\ifinserthangingsymbol
7365 \newcommand{\inserthangingsymbol}{%
7366 \ifinserthangingsymbol%
7367 \ifinstanza%
7368 \@hangingsymbol%
7369 \fi%
7370 \fi%
7371 }
7372 %

```

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.

```

7373 \newcommand*{\ampersand}{\char`\&}
7374
7375 %

```

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.

```

7376 \chardef\body=\catcode`\@
7377 \catcode`\@=11
7378 \chardef\next=\catcode`\&
7379 \catcode`\&=\active
7380
7381 %

```

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.

```

7382 \newcount\stanza@count
7383 \newlength{\stanzaindentbase}
7384 \setlength{\stanzaindentbase}{20pt}
7385
7386 %

```

\strip@szacnt The indentations of stanza lines are non-negative integer multiples of the unit called
\setstanzavalues `\stanzaindentbase`. To make it easier for the user to specify these numbers, some list macros are defined. These take numerical values in a list separated by commas and assign the values to special control sequences using `\mathchardef`. Though this does

limit the range from 0 to 32767, it should suffice for most applications, including *penalties*, which will be discussed below.

```

7387 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
7388 \newcommand*\setstanzavalues}[2]{\def\@tempa{#2,,|}%
7389   \stanza@count\z@
7390   \def\next{\expandafter\strip@szacnt\@tempa
7391     \ifx\@tempb\empty\let\next\relax\else
7392     \expandafter\mathchardef\csname #1@\number\stanza@count
7393     @\endcsname\@tempb\relax
7394     \advance\stanza@count\@ne\fi\next}%
7395   \next}
7396
7397 %

```

`\setstanzaindents` In the original edmac, `\setstanzavalues{sza}{...}` had to be called to set the indents, and similarly `\setstanzavalues{szp}{...}` to set the penalties. `\setstanzaindents` and `\setstanzapenalties` macros are a convenience to give the user one less thing to worry about (misspelling the first argument).

```

7398 \newcommand*\setstanzaindents}[1]{\setstanzavalues{sza}{#1}}
7399 \newcommand*\setstanzapenalties}[1]{\setstanzavalues{szp}{#1}}
7400 %
7401 %

```

`\managestanza@modulo` Since version 0.13, the `stanzaindentsrepetition` counter can be used when the indentation is repeated every *n* verses. The `\managestanza@modulo` is a command which modifies the counter `stanza@modulo`. The command adds 1 to `stanza@modulo`, but if `stanza@modulo` is equal to the `stanzaindentsrepetition` counter, the command restarts it.

```

7402 \newcounter{stanzaindentsrepetition}
7403 \newcount\stanza@modulo
7404
7405 \newcommand*\managestanza@modulo}[0]{%
7406   \advance\stanza@modulo\@ne%
7407   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
7408   \stanza@modulo\@ne%
7409   \fi%
7410 }
7411 %

```

`\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.

```

7412 \newcommand{\stanzaindent}[1]{%
7413   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
7414   \ignorespaces%
7415 }%

```

```

7416 \WithSuffix\newcommand\stanzaindent*[1]{%
7417   \stanzaindent{#1}%
7418   \global\advance\stanza@modulo-\@ne%
7419   \ifnum\stanza@modulo=0%
7420     \global\stanza@modulo=\value{stanzaindentsrepetition}%
7421   \fi%
7422   \ignorespaces%
7423 }%
7424 %

```

XXVIII.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```

\thestanza25 \newcounter{stanza}
7426 \renewcommand{\thestanza}{%
7427   \textbf{\arabic{stanza}}%
7428 }
7429 %

```

\ifnumberstanza Then, macro to activate automatically numbering of stanza.

```

7430 \newif\ifnumberstanza%
7431 %

```

\@insertstanzanumber Now, macro called at the first line of of verse of a stanza.

```

7432 \newcommand{\@insertstanzanumber}[0]{%
7433   \ifnumberstanza%
7434     \ifl@pairing%
7435       \ifledRcol%
7436         \stanzanumwrapper{\thestanzaR}%
7437       \else%
7438         \stanzanumwrapper{\thestanzaL}%
7439       \fi%
7440     \else%
7441       \stanzanumwrapper{\thestanza}%
7442     \fi%
7443     \setline{1}%
7444   \fi%
7445 }%
7446 %

```

\@advancestanzanumber Also a command to advance the counter of stanza.

```

7447 \newcommand{\@advancestanzanumber}[0]{%
7448   \ifnumberstanza%
7449     \ifl@pairing%
7450       \ifledRcol%

```

```

7451 \addtocounter{stanzaR}{1}%
7452 \else%
7453 \addtocounter{stanzaL}{1}%
7454 \fi%
7455 \else%
7456 \addtocounter{stanza}{1}%
7457 \fi%
7458 \fi%
7459 }%
7460 %

```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

7461 \newcommand{\stanzanumwrapper}[1]{%
7462 \flagstanza{#1}%
7463 }%
7464 %

```

XXVIII.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

7465 \newcommand{\printstanza}[0]{%
7466 \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or bool{
7467 !@dprintingcolumns}}{%
7468 \ifledRcol{%
7469 \thestanzaR%
7470 \else%
7471 \thestanzaL%
7472 \fi%
7473 }{%
7474 \thestanza%
7475 }%
7476 }%

```

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.

```

7477 \newcommandx{\stanza@line}[2][1,2,usedefault]{%
7478 \ifnum\value{stanzaindentrepetition}=0

```

```

7479 \ifcsdef{sza@\number\stanza@count @}%
7480 {%
7481 \parindent=\csname sza@\number\stanza@count @\endcsname\
stanzaindentbase%
7482 }{%
7483 \led@err@StanzaIndentNotDefined%
7484 }%
7485 \else
7486 \ifcsdef{sza@\number\stanza@modulo @}{%
7487 \parindent=\csname sza@\number\stanza@modulo @\endcsname\
stanzaindentbase%
7488 \managestanza@modulo%
7489 }%
7490 {%
7491 \led@err@StanzaIndentNotDefined%
7492 }%
7493 \fi
7494 \pstart[#1][#2]\stanza@hang\ignorespaces%
7495 }%
7496 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
7497 \hangindent\expandafter
7498 \noexpand\csname sza@0@\endcsname\stanzaindentbase
7499 \hangafter\@ne}
7500 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
7501 \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
7502 \penalty\fi\count@}
7503 %

```

\@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
 \AtStartEveryStanza above we have used \xdef to make the stored macros take up a bit less space, but it also
 \BeforeEveryStopStanza makes them more obscure to the reader. Lines of the stanza are delimited by ampersands
 \newverse &. The last line of the stanza must end with \&.

```

7504 \xdef\@startstanza[#1][#2]{%
7505 \noexpand\instanzatrue\expandafter
7506 \begingroup%
7507 \catcode`\noexpand\&\active%
7508 \global\stanza@count\@ne\stanza@modulo\@ne
7509 \noexpand\ifnum\expandafter\noexpand
7510 \csname sza@0@\endcsname=\z@\let\noexpand\stanza@hang\relax
7511 \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
7512 \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
7513 \expandafter\noexpand\csname szp@0@\endcsname=\z@
7514 \let\noexpand\sza@penalty\relax\noexpand\fi%
7515 \def\noexpand&{%

```

```

7516 \noexpand\newverse[] []}%
7517 \def\noexpand\&\noexpand\@stopstanza}%
7518 \noexpand\@advancestanzanumber%
7519 \noexpand\stanza@line[#1] [#2]%
7520 \noexpand\@insertstanzanumber%
7521 \let\par\relax\ignorespaces%No paragraph in verses
7522 }
7523
7524 \newcommandx{\stanza}[2][1,2,usedefault]{%
7525 \ifboolexpr{%
7526 not test{\ifdefvoid{\at@every@stanza}}}%
7527 and test{\ifstrempy{#1}}}%
7528 and test{\ifstrempy{#2}}}%
7529 {\@startstanza[] [\at@every@stanza] \at@start@every@stanza}%
7530 {\@startstanza[#1] [#2] \at@start@every@stanza}%
7531 }%
7532
7533 \newcommandx{\@stopstanza}[2][1,2,usedefault]{%
7534 \unskip%
7535 \endlock%
7536 \ifboolexpr{%
7537 not test{\ifdefvoid{\at@every@stop@stanza}}}%
7538 and test{\ifstrempy{#1}}}%
7539 and test{\ifstrempy{#2}}}%
7540 {\before@every@stop@stanza\pend[] [\at@every@stop@stanza]}%
7541 {\before@every@stop@stanza\pend[#1] [#2]}%
7542 \endgroup%
7543 \instanzafalse%
7544 }
7545
7546 \newcommand{\AtEveryStopStanza}[1]{%
7547 \ifstrempy{#1}%
7548 {\gdef\at@every@stop@stanza{}}%
7549 {\gdef\at@every@stop@stanza{\noindent#1}}%
7550 }%
7551 \WithSuffix\newcommand\AtEveryStopStanza*[1]{%
7552 \ifstrempy{#1}%
7553 {\gdef\at@every@stop@stanza{}}%
7554 {\gdef\at@every@stop@stanza{#1}}%
7555 }%
7556 \def\at@every@stop@stanza{}%
7557
7558 \newcommand{\AtEveryStanza}[1]{%
7559 \ifstrempy{#1}%
7560 {\gdef\at@every@stanza{}}%
7561 {\gdef\at@every@stanza{\noindent#1}}%
7562 }%
7563 \WithSuffix\newcommand\AtEveryStanza*[1]{%
7564 \ifstrempy{#1}%
7565 {\gdef\at@every@stanza{}}%

```

```

7566     {\gdef\at@every@stanza{#1}}%
7567   }%
7568
7569
7570
7571   \newcommand{\AtStartEveryStanza}[1]{%
7572     \ifstrepty{#1}%
7573       {\gdef\at@start@every@stanza{}}%
7574       {\gdef\at@start@every@stanza{#1}}%
7575   }%
7576   \def\at@start@every@stanza{}%
7577
7578   \newcommand{\BeforeEveryStopStanza}[1]{%
7579     \ifstrepty{#1}%
7580       {\gdef\before@every@stop@stanza{}}%
7581       {\gdef\before@every@stop@stanza{#1}}%
7582   }%
7583   \def\before@every@stop@stanza{}%
7584
7585   \newcommand*\newverse}[4][1,2,3,4,usedefault]{%
7586     \unskip%
7587     \endlock\pend[#1][#3]\sza@penalty\global%
7588     \advance\stanza@count\@ne\stanza@line[#2][#4]%
7589   }
7590
7591   %

```

\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`.

```

7592   \newcommand*\flagstanza}[2][\stanzaindentbase]{%
7593     \hskip -#1\llap{#2}\hskip #1\ignorespaces}
7594
7595   %

```

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.

```

7596   \catcode`\&=\next
7597   \catcode`\@=\body
7598   \setstanzavalues{szp}{0}
7599
7600   %

```

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 are two associated etoolbox counters.

```

7601 \def\msdata@c{}%
7602 \def\msdata@cR{}%
7603 \newcommand{\msdata}[1]{%
7604   \leavevmode%
7605   \unless\ifstopmsdata@inserted@%
7606     \stopmsdata%
7607     \led@warning@msdatawithoutstop%
7608   \fi%
7609   \global\stopmsdata@inserted@false%
7610   \unless\ifledRcol%
7611     \numgdef{\msdata@c}{\msdata@c+1}%
7612     \ifdef{\hypertarget}{%
7613       \edlabel{\msdata@c:start:msdata}%
7614     }{}%
7615     \protected@write\linenum@out{}{%
7616       \string\@msd{#1}%
7617     }%
7618   \else%
7619     \numgdef{\msdata@cR}{\msdata@cR+1}%
7620     \ifdef{\hypertarget}{%
7621       \edlabel{\msdata@cR:start:msdata}%
7622     }{}%
7623     \protected@write\linenum@outR{}{%
7624       \string\@msd{#1}%
7625     }%
7626   \fi%
7627 }%
7628 %

```

`\stopmsdata` The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

7629 \newcommand{\stopmsdata}[0]{%
7630   \leavevmode%
7631   \unless\ifledRcol%
7632     \protected@write\linenum@out{}{%
7633       \string\@stopmsd%
7634     }%
7635     \ifdef{\hypertarget}{%
7636       \edlabel{\msdata@c:end:msdata}%
7637     }{}%
7638   \else%
7639     \protected@write\linenum@outR{}{%

```

```

7640     \string\@stopmsd%
7641     }%
7642     \ifdef{\hypertarget}{%
7643         \edlabel{\msdata@cR:end:msdata}%
7644     }{}%
7645     \fi%
7646     \global\stopmsdata@inserted@true%
7647 }%
7648 %

```

\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`

```

7649 \newif\ifstopmsdata@inserted@%
7650 %

```

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?

```

7651 \newcommand{\setmsdataseries}[1]{%
7652     \gdef\@msdata@series{#1}%
7653 }%
7654 \def\@msdata@series{A}%
7655 %

```

\setmsdataposition The label for the manuscripts data.

```

7656 \def\ms@data@position{msdata-regular}%
7657 \newcommand{\setmsdataposition}[1]{%
7658     \gdef\ms@data@position{#1}%
7659 }%
7660 %

```

\setmsdatalabel The label for the manuscripts data.

```

7661 \def\ms@data@label{Ms.}%
7662 \newcommand{\setmsdatalabel}[1]{%
7663     \gdef\ms@data@label{#1}%
7664 }%
7665 %

```


XXIX.3 Counters and lists

\@msd@c \@msd@c is a counter incremented at each \@msd read in auxiliary file.

```
7666 \numdef{\@msd@c}{0}
7667 \numdef{\@msd@cR}{0}
7668 %
```

\add@msd@ \add@msd@ is a counter incremented at each \add@msd@data, that is at each time we prepare the insertion of manuscripts data footnote.

```
7669 \numdef{\add@msd@c}{0}%
7670 \numdef{\add@msd@cR}{0}%
7671 %
```

\@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.

```
7672 \def\@msdata@list{}%
7673 %
```

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.

```
7674 \newcommand{\@msd}[1]{%
7675   \unless\ifledRcol%
7676     \global\numdef{\@msd@c}{\@msd@c+\@ne}%
7677     \csgdef{\@msdata@\@msd@c @data}{#1}%
7678     \csxdef{\@msdata@\@msd@c @linenumber}{\the\line@num}%
7679     \csxdef{\@msdata@\@msd@c @abslinenumber}{\the\absline@num}%
7680     \xright@appenditem{\the\absline@num}\to\actionlines@list%
7681     \xright@appenditem{-1010}\to\actions@list%
7682   \else%
7683     \global\numdef{\@msd@cR}{\@msd@cR+\@ne}%
7684     \csgdef{\@msdata@\@msd@cR @dataR}{#1}%
7685     \csxdef{\@msdata@\@msd@cR @linenumberR}{\the\line@numR}%
7686     \csxdef{\@msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
7687     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
7688     \xright@appenditem{-1010}\to\actions@listR%
7689   \fi%
7690 }%
7691 %
```

\@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.

```

7692 \newcommand{\@stopmsd}[0]{%
7693   \unless\ifledRcol%
7694     \ifcsundef{@msdata@\@msd@c @stoplinenumber}{%
7695       \csxdef{@msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
7696       \csxdef{@msdata@\@msd@c @stoplinenumber}{\the\line@num}%
7697     }{}%
7698   \else%
7699     \ifcsundef{@msdata@\@msd@cR @stoplinenumberR}{%
7700       \csxdef{@msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
7701       \csxdef{@msdata@\@msd@cR @stoplinenumberR}{\the\line@numR}%
7702     }{}%
7703   }{}%
7704   \fi%
7705 }%
7706 %

```

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.

```

7707 \newcommand{\add@msdata}{%
7708   \bgroup%
7709   \normalfont%
7710   \unless\ifledRcol%
7711     \numgdef{\add@msd@c}{\add@msd@c+\@ne}%
7712     \ifcsdef{@msdata@\add@msd@c @data}{%
7713       \letcs{\@data}{@msdata@\add@msd@c @data}%
7714       \edef\l@d@nums{%
7715         000| % Start page = we don't print it
7716         \csuse{@msdata@\add@msd@c @linenumber}| % Start line number
7717         000| % Start subline number, for now, not used
7718         000| % End page number, we don't print it
7719         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
7720         {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
7721         {\csuse{@lastline@forpage@\the\page@num}}}%Otherwiser, last
number of the page
7722         |%
7723         000| % End sub line number, for now, not used
7724         \edfont@info%Font
7725       }%
7726       \@msd@options@fullpagefalse%
7727       \if@firstlineofpage%Try if the data are for the full page. If yes
, will add options to the list.
7728       \unless\if@msdata@insertedfrompreviouspage%

```

```

7729 \ifnumless{\csuse{@lastabsline@forpage@the\page@num}}{\csuse
{\@msdata@add@msd@c @stopabslinenumber}+\@one}%
7730 {%
7731 \numdef{\@tmp}{\add@msd@c+\@one}%
7732 \ifcsdef{\@msdata@\@tmp @abslinenumber}%
7733 {\ifnumequal{\csuse{\@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@the\page@num}}}%
7734 {}%
7735 {\@msd@options@fullpagetrue}%
7736 }%
7737 {\@msd@options@fullpagetrue}%
7738 }%
7739 {}%
7740 \fi%
7741 \fi%
7742 \listxadd{\@msdata@list}{%
7743 \@msd@options@iffullpage%
7744 \ifluatex%
7745 \csxdef{footnote@luatextextdir}{the\textdir}%
7746 \csxdef{footnote@luatexpardir}{the\pardir}%
7747 \fi%
7748 \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
7749 \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
7750 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}}%
7751 \reset@msd@options@iffullpage%
7752 }%
7753 }%
7754 {}%
7755 \else%
7756 \numgdef{\add@msd@cR}{\add@msd@cR+\@one}%
7757 \ifcsdef{\@msdata@\add@msd@cR @dataR}{%
7758 \letcs{\@data}{\@msdata@\add@msd@cR @dataR}%
7759 \edef\l@d@nums{%
7760 000| % Start page = we don't print it
7761 \csuse{\@msdata@\add@msd@cR @linenumberR}| % Start line number
7762 000| % Start subline number, for now, not used
7763 000| % End page number, we don't print it
7764 \ifnumless{\csuse{\@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}}%
7765 {\csuse{\@msdata@\add@msd@cR @stoplinenumberR}} % End line number
if in the same page
7766 {\csuse{@lastline@forpageR@the\page@numR}} % Otherwiser, last
number of the page
7767 |%
7768 000| % End sub line number, for now, not used
7769 \edef@info{Font
7770 }%
7771 \@msd@options@fullpagefalse%
7772 \if@firstlineofpageR%

```

```

7773 \unless\if@msdata@insertedfrompreviouspage%
7774 \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\
csuse{@msdata@add@msd@c @stopabslinenumberR}+\@ne}%
7775 {%
7776 \numdef{\@tmp}{\add@msd@cR+\@ne}%
7777 \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7778 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
7779 {}%
7780 {\@msd@options@fullpagetrue}%
7781 }%
7782 {\@msd@options@fullpagetrue}%
7783 }%
7784 {}%
7785 \fi%
7786 \fi%
7787 \listxadd{\@msdata@list}{%
7788 \@msd@options@iffullpage%
7789 \ifluatex%
7790 \csxdef{footnote@luatextextdir}{\the\textdir}%
7791 \csxdef{footnote@luatexpardir}{\the\pardir}%
7792 \fi%
7793 \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
7794 \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
7795 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
7796 \reset@msd@options@iffullpage%
7797 }%
7798 }%
7799 {}%
7800 \fi%
7801 \egroup%
7802 }%
7803 %

```

`\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.

```

7804 \newif\if@msdata@insertedfrompreviouspage%
7805 %

```

`\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.

```

7806 \newcommand{\add@msdata@firstlineofpage}{%
7807 \bgroup%
7808 \normalfont%
7809 \unless\ifledRcol@%
7810 \ifcsdef{@msdata@\add@msd@c @data}{%

```

```

7811 \ifnumless{\the\absline@num-\@ne}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}%
7812 {%
7813 \global\@msdata@insertedfrompreviouspagetrue%
7814 \letcs{@data}{\@msdata@\add@msd@c @data}%
7815 \edef\l@d@nums{%
7816 000|% Start page = we don't print it
7817 \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
7818 000|% Start subline number, for now, not used
7819 000|% End page number, we don't print it
7820 \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}%
7821 {\csuse{@msdata@\add@msd@c @stoplinenumber}}%End line number if
in the same page
7822 {\csuse{@lastline@forpage@\the\page@num}}%Otherwise, last
number of the page
7823 |%
7824 000|% End sub line number, for now, not used
7825 \edefont@info%Font
7826 }%
7827 \@msd@options@fullpagefalse%
7828 \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
7829 {%
7830 \numdef{\@tmp}{\add@msd@c+\@ne}%
7831 \ifcsdef{@msdata@\@tmp @abslinenumber}%
7832 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}%
7833 {}%
7834 {\@msd@options@fullpagetrue}%
7835 }%
7836 {\@msd@options@fullpagetrue}%
7837 }%
7838 {}%
7839 \listxadd{\@msdata@list}{%
7840 \@msd@options@iffullpage%
7841 \ifluatex%
7842 \csxdef{footnote@luatextextdir}{\the\textdir}%
7843 \csxdef{footnote@luatexpardir}{\the\pardir}%
7844 \fi%
7845 \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
7846 \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
7847 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\{
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
7848 \reset@msd@options@iffullpage%
7849 }%
7850 }%

```

```

7851     {\global\@msdata@insertedfrompreviouspagefalse}%
7852   }{}%
7853   \else%
7854     \ifcsdef{@msdata@\add@msd@cR @dataR}{%
7855       \ifnumless{\the\absline@numR-\@one}{\csuse{@msdata@\add@msd@cR
@stopabslinenumberR}}%
7856       {%
7857         \global\@msdata@insertedfrompreviouspagetrue%
7858         \letcs{\@data}{@msdata@\add@msd@cR @dataR}%
7859         \edef\l@d@nums{%
7860           000}% Start page = we don't print it
7861         \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
7862         000}% Start subline number, for now, not used
7863         000}% End page number, we don't print it
7864         \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@\the\page@numR}}%
7865         {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
7866         {\csuse{@lastline@forpageR@\the\page@numR}}%Otherwise, last
number of the page
7867         |%
7868         000}% End sub line number, for now, not used
7869         \edfont@info%Font
7870       }%
7871       \@msd@options@fullpagefalse%
7872       \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\csuse{
@msdata@\add@msd@cR @stopabslinenumberR}+\@one}%
7873       {%
7874         \numdef{\@tmp}{\add@msd@cR+\@one}%
7875         \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7876         {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}}%
7877         {}%
7878         {\@msd@options@fullpagetrue}%
7879       }%
7880       {\@msd@options@fullpagetrue}%
7881     }%
7882   {}%
7883   \listxadd{\@msdata@list}{%
7884     \@msd@options@iffullpage%
7885     \ifluatex%
7886       \csxdef{footnote@luatextextdir}{\the\textdir}%
7887       \csxdef{footnote@luatexpardir}{\the\pardir}%
7888     \fi%
7889     \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
7890     \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
7891     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%

```

```

7892         \reset@msd@options@iffullpage%
7893     }%
7894 }%
7895 {\global\@msdata@insertedfrompreviouspagefalse}%
7896 }{}%
7897 \fi%
7898 \egroup%
7899 }%
7900 %

```

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@msdata01 \newcommand{\insert@msdata}{%
7902     \def\do##1{##1}%
7903     \dolistloop{\@msdata@list}%
7904     \global\let\@msdata@list\relax%
7905 }%
7906 %

```

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.

```

7907 \newif\if@msd@options@fullpage%
7908 \newcommand{\@msd@options@iffullpage}[0]{%
7909     \if@msd@options@fullpage%
7910         \noexpand\toggletrue{nonum@}%
7911         \ifdefvoid{\ms@data@label}%
7912             {\noexpand\toggletrue{nosep@}}%
7913         {}%
7914     \fi%
7915 }%
7916 \newcommand{\reset@msd@options@iffullpage}[0]{%
7917     \noexpand\togglefalse{nonum@}%
7918     \noexpand\togglefalse{nosep@}%
7919 }%
7920 %

```

XXX Arrays and tables

XXX.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread ‘`eeq` and `amstex`’, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsgen` package.

```
7921 \newtoks\@emptytoks
7922
7923 %
```

The rest is from `amsmath`.

`\l@denbody` A token register to contain the body.

```
7924 \newtoks\l@denbody
7925
7926 %
```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```
7927 \newcommand{\addtol@denbody}[1]{%
7928   \global\l@denbody\expandafter{\the\l@denbody#1}}
7929
7930 %
```

`\l@dcollect@body` The macro `\l@dcollect@body` starts the scan for the `\end{env}` command of the current environment. It takes a macro name as argument. This macro is supposed to take the whole body of the environment as its argument. For example, given `cenv#1{...}` as a macro that processes `#1`, then the environment form, `\begin{env}` would call `\l@dcollect@body\cenv`.

```
7931 \newcommand{\l@dcollect@body}[1]{%
7932   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}}%
7933   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}}%
7934   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
7935   \begingroup
7936     \expandafter\let\csname\@currenvir\endcsname\l@dcollect@body
7937     \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
7938     \processl@denbody%
7939   }%
7940
7941 %
```


`\l@dpush@begins` When adding a piece of the current environment's contents to `\l@denvbody`, we scan it to check for additional `\begin` tokens, and add a 'b' to the stack for any that we find.

```

7942 \def\l@dpush@begins#1\begin#2{%
7943   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
7944
7945 %

```

`\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.

```

7946 \def\l@dcollect@@body#1\end#2{%
7947   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
7948     \expandafter\@gobble\l@dbegin@stack}%
7949   \ifx\@empty\l@dbegin@stack
7950     \endgroup
7951     \@checkend{#2}%
7952     \addtol@denvbody{#1}%
7953   \else
7954     \addtol@denvbody{#1\end{#2}}%
7955   \fi
7956   \processl@denvbody % A little tricky! Note the grouping
7957 }
7958
7959 %

```

There was a question on CTT about how to use `\collect@body` for a macro taking an argument. The following is part of that thread.

From: Heiko Oberdiek <oberdiek@uni-freiburg.de>
 Newsgroups: comp.text.tex
 Subject: Re: Using `\collect@body` with commands that take >1 argument
 Date: Fri, 08 Aug 2003 09:03:20 +0200

eed132@psu.edu (Evan) wrote:
 > I'm trying to make a new Latex environment that acts like the
 > `\colorbox` command that is part of the `color` package. I looked through
 > the FAQ and ran across this bit about using the `\collect@body` command
 > that is part of AMSLaTeX:
 > <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=cmdasenv>
 >
 > It almost works. If I do something like the following:
 > `\newcommand{\redbox}[1]{\colorbox{red}{#1}}`
 >
 > `\makeatletter`
 > `\newenvironment{redbox}{\collect@body \redbox}{}`

You will get an error message: Command `\redbox` already defined.
Thus you must rename either the command `\redbox` or the environment name.

```
> \begin{coloredbox}{blue}
>   Yadda yadda yadda... this is on a blue background...
> \end{coloredbox}
> and can't figure out how to make the \collect@body take this.

> \collect@body \colorbox{red}
> \collect@body {\colorbox{red}}
```

The argument of `\collect@body` has to be one token exactly.

```
\documentclass{article}
\usepackage{color}
\usepackage{amsmath}

\newcommand{\redbox}[1]{\colorbox{red}{#1}}
\makeatletter
\newenvironment{coloredbox}[1]{%
  \def\next@{\colorbox{#1}}%
  \collect@body\next@
}{%

% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}[1]{%
  \def\next@{\mycoloredbox{#1}}%
  \collect@body\next@
}{%
\newcommand{\mycoloredbox}[2]{%
  \colorbox{#1}{\ignorespaces#2\unskip}%
}

% support of optional color model argument
\newcommand\coloredboxIII\endcsname{}
\def\coloredboxIII#1#2{%
  \@coloredboxIII{#1}%
}
\def\@coloredboxIII#1#2{%
  \def\next@{\mycoloredboxIII{#1}{#2}}%
  \collect@body\next@
}
\newcommand{\mycoloredboxIII}[3]{%
  \colorbox{#1}{#2}{\ignorespaces#3\unskip}%
}

\makeatother
```

```

\begin{document}
  Black text before
  \begin{coloredbox}{blue}
    Hello World
  \end{coloredbox}
  Black text after

  Black text before
  \begin{coloredboxII}{blue}
    Hello World
  \end{coloredboxII}
  Black text after

  Black text before
  \begin{coloredboxIII}[rgb]{0,0,1}
    Hello World
  \end{coloredboxIII}
  Black text after

\end{document}

Yours sincerely
Heiko <oberdiek@uni-freiburg.de>

```

XXX.2 Tabular environments

This is based on the work by Herbert Breger in developing `tabmac.tex`.

The original `tabmac.tex` file was void of comments or any explanatory text other than the above notice. The algorithm is Breger's. Peter Wilson have made some cosmetic changes to the original code and reimplemented some things so they are more LaTeX-like. All the commentary are from Peter Wilson, as are any mistake or errors.

However, Maïeul Rouquette has modified code in order to add new features of `eledmac` and `reledmac`.

XXX.2.1 Disabling and restoring commands

`\l@dtabnoexpands` More no expansion for critical and familiar footnotes in tabular environment.

```

7960 \newcommand*\l@dtabnoexpands{%
7961   \let\rtab=0%
7962   \let\ctab=0%
7963   \let\ltab=0%
7964   \let\rtabtext=0%
7965   \let\ltabtext=0%
7966   \let\ctabtext=0%
7967   \let\edbeforetab=0%
7968   \let\edaftertab=0%
7969   \let\edatleft=0%

```

```

7970 \let\edatright=0%
7971 \let\edvertline=0%
7972 \let\edvertdots=0%
7973 \let\edrowfill=0%
7974 }
7975
7976 %

```

\disable@familiarnotes Macros to disable and restore familiar notes, to prevent them from printing multiple times in edtabularx and edarrayx environments.

\restore@familiarnotes

```

7977 \newcommand{\disable@familiarnotes}{%
7978   \unless\ifnofamiliar@%
7979     \def\do##1{%
7980       \csletcs{footnote@@##1}{footnote##1}%
7981       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
7982         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
7983         \csuse{@footnotemark##1}%
7984       }%
7985     }%
7986     \dolistloop{\@series}%
7987   \fi%
7988 }%
7989 \newcommand{\restore@familiarnotes}{%
7990   \unless\ifnofamiliar@%
7991     \def\do##1{%
7992       \csletcs{footnote##1}{footnote@@##1}%
7993     }%
7994     \dolistloop{\@series}%
7995   \fi%
7996 }%
7997
7998 %

```

\disable@sidenotes The same, for side notes.

\restore@sidenotes

```

7999 \newcommand{\disable@sidenotes}{%
8000   \let\@@ledrightnote\ledrightnote%
8001   \let\@@ledleftnote\ledleftnote%
8002   \let\@@ledsidenote\ledsidenote%
8003   \let\ledrightnote@gobble%
8004   \let\ledleftnote@gobble%
8005   \let\ledsidenote@gobble%
8006 }%
8007 \newcommand{\restore@sidenotes}{%
8008   \let\ledrightnote\@@ledrightnote%
8009   \let\ledleftnote\@@ledleftnote%
8010   \let\ledsidenote\@@ledsidenote%
8011 }%
8012 %

```

`\disable@notes` Disable/restore side and familiar notes.

```
\restore@notes
8013 \newcommand{\disable@notes}{%
8014   \disable@sidenotes%
8015   \disable@familiarnotes%
8016 }%
8017 \newcommand{\restore@notes}{%
8018   \restore@sidenotes%
8019   \restore@familiarnotes%
8020 }%
8021 %
```

`\EDTEXT` We need to be able to modify the `\edtext` macros and also restore their original definitions.

```
\xedtext
8022 \let\EDTEXT=\edtext
8023 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
8024 %
```

`\EDLABEL` We need to be able to modify and restore the `\edlabel` macro.

```
\xedlabel
8025 \let\EDLABEL=\edlabel
8026 \newcommand*\{\xedlabel}[1]{\EDLABEL{#1}}
8027 %
```

`\xedindex` Macros supporting modification and restoration of `\edindex`.

```
\nulledindex
8028 \AtBeginDocument{\let\xedindex\edindex}%
8029 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
8030
8031 %
```

`\@line@num` Macro supporting restoration of `\linenum`.

```
8032 \let\@line@num=\linenum
8033 %
```

`\l@dgobblearg` `\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```
8034 \newcommand*\{\l@dgobbleoptarg}[2][\relax]%
8035
8036 %
```

`\Relax` `\let\Relax=\relax`

`\NEXT` `\let\NEXT=\next`

```
8039
8040 %
```

\l@modforedtext Modify and restore various macros for when \edtext is used.
\l@drestoreforedtext

```

8041 \newcommand{\l@modforedtext}{%
8042   \let\edtext\relax
8043   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
8044   \dolistloop{\@series}%
8045   \let\edindex\nulledindex
8046   \let\linenum@gobble}
8047 \newcommand{\l@drestoreforedtext}{%
8048   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
8049   \dolistloop{\@series}%
8050   \let\edindex\xedndex}
8051 %

```

\l@dnnullfills Nullify and restore some column fillers, etc.

\l@drestorefills

```

8052 \newcommand{\l@dnnullfills}{%
8053   \def\edlabel##1{%
8054     \def\edrowfill##1##2##3{%
8055     }
8056   \newcommand{\l@drestorefills}{%
8057     \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
8058   }
8059
8060 %

```

\letsforverteilen Gathers some lets and other code that is common to the *verteilen* macros.

```

8061 \newcommand{\letsforverteilen}{%
8062   \let\edtext\xedndex
8063   \let\edindex\xedndex
8064   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
8065   \dolistloop{\@series}%
8066   \let\linenum@line@num
8067   \hilfsskip=\l@dcolwidth%
8068   \advance\hilfsskip by -\wd\hilfsbox
8069   \def\edlabel##1{\xedlabel{##1}}
8070
8071 %

```

\disablel@dtabfeet Declarations for using or using \edtext inside tabulars. The default at this point is for
\enablel@dtabfeet \edtext.

```

8072 \newcommand\disablel@dtabfeet{\l@modforedtext}%
8073 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
8074 %

```

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.

```

8075 \newcount\l@dampcount
8076 \l@dampcount=1\relax
8077 \newcount\l@dcolcount
8078 \l@dcolcount=0\relax
8079
8080 %

```

`\hilfsbox` Some (temporary) helper items.

```

\hilfsskip
\Hilfsbox
\hilfscount
8081 \newbox\hilfsbox
8082 \newskip\hilfsskip
8083 \newbox\Hilfsbox
8084 \newcount\hilfscount
8085
8086 %

```

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).

```

8087 \newdimen\dcoli
8088 \newdimen\dcolii
8089 \newdimen\dcoliii
8090 \newdimen\dcoliv
8091 \newdimen\dcolv
8092 \newdimen\dcolvi
8093 \newdimen\dcolvii
8094 \newdimen\dcolviii
8095 \newdimen\dcolix
8096 \newdimen\dcolx
8097 \newdimen\dcolxi
8098 \newdimen\dcolxii
8099 \newdimen\dcolxiii
8100 \newdimen\dcolxiv
8101 \newdimen\dcolxv
8102 \newdimen\dcolxvi
8103 \newdimen\dcolxvii
8104 \newdimen\dcolxviii
8105 \newdimen\dcolxix
8106 \newdimen\dcolxx
8107 \newdimen\dcolxxi
8108 \newdimen\dcolxxii
8109 \newdimen\dcolxxiii
8110 \newdimen\dcolxxiv
8111 \newdimen\dcolxxv
8112 \newdimen\dcolxxvi
8113 \newdimen\dcolxxvii

```

```

8114 \newdimen\dcollxxviii
8115 \newdimen\dcollxxix
8116 \newdimen\dcollxxx
8117 \newdimen\dcollerr    % added for error handling
8118
8119 %

```

\l@dcollwidth This is a cunning way of storing the columnwidths indexed by the column number \l@dcollcount, like an array. (was \Dimenzuordnung)

```

8120 \newcommand{\l@dcollwidth}{\ifcase \the\l@dcollcount \dcoli %???
8121 \or \dcoli \or \dcolii \or \dcoliii
8122 \or \dcoliv \or \dcolv \or \dcolvi
8123 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
8124 \or \dcolxi \or \dcolxii \or \dcolxiii
8125 \or \dcolxiv \or \dcolxv \or \dcolxvi
8126 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
8127 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
8128 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
8129 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
8130 \else \dcollerr \fi}
8131
8132 %

```

\step1@dcollcount This increments the column counter, and issues an error message if it is too large.

```

8133 \newcommand*{\step1@dcollcount}{\advance\l@dcollcount\@ne
8134 \ifnum\l@dcollcount>30\relax
8135 \led@err@TooManyColumns
8136 \fi}
8137
8138 %

```

\l@dsetmaxcollwidth Sets the column width to the maximum value seen so far.

```

8139 \newcommand{\l@dsetmaxcollwidth}{%
8140 \ifdim\l@dcollwidth < \wd\hilfsbox
8141 \l@dcollwidth = \wd\hilfsbox
8142 \else \relax \fi}
8143
8144 %

```

\measuremcell Measure (recursively) the width required for a math cell.

```

8145 \def\measuremcell #1&{%
8146 \ifx #1\\ \ifnum\l@dcollcount=0\let\NEXT\relax%
8147 \else\l@dcheckcols%
8148 \l@dcollcount=0%
8149 \let\NEXT\measuremcell%
8150 \fi%

```



```

8151 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8152 \step1@dcolcount%
8153 \l@setmaxcolwidth%
8154 \let\NEXT\measuremcell%
8155 \fi\NEXT}
8156
8157 %

```

\measuretcell Measure (recursively) the width required for a text cell.

```

8158 \def\measuretcell #1{%
8159 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8160 \else\l@dcheckcols%
8161 \l@dcolcount=0%
8162 \let\NEXT\measuretcell%
8163 \fi%
8164 \else\setbox\hilfsbox=\hbox{#1}%
8165 \step1@dcolcount%
8166 \l@setmaxcolwidth%
8167 \let\NEXT\measuretcell%
8168 \fi\NEXT}
8169
8170 %

```

\measuremrow Measure (recursively) the width required for a math row.

```

8171 \def\measuremrow #1{%
8172 \ifx #1\let\NEXT\relax%
8173 \else\measuremcell #1\&\&\&%
8174 \let\NEXT\measuremrow%
8175 \fi\NEXT}
8176 %

```

\measuretrow Measure (recursively) the width required for a text row.

```

8177 \def\measuretrow #1{%
8178 \ifx #1\let\NEXT\relax%
8179 \else\measuretcell #1\&\&\&%
8180 \let\NEXT\measuretrow%
8181 \fi\NEXT}
8182
8183 %

```

\edtabcolsep The length \edtabcolsep controls the distance between columns.

```

8184 \newskip\edtabcolsep
8185 \global\edtabcolsep=10pt
8186
8187 %

```

```
\variab88 \newcommand{\variab}{\relax}
```

```
8189
```

```
8190 %
```

\l@dcheckcols Check that the number of columns is consistent.

```
8191 \newcommand*{\l@dcheckcols}{%
8192   \ifnum\l@dcolcount=1\relax
8193   \else
8194     \ifnum\l@dampcount=1\relax
8195     \else
8196       \ifnum\l@dcolcount=\l@dampcount\relax
8197       \else
8198         \l@d@err@UnequalColumns
8199       \fi
8200     \fi
8201     \l@dampcount=\l@dcolcount
8202   \fi}
8203
8204 %
```

\edfilldimen A length.

```
8205 \newdimen\edfilldimen
8206 \edfilldimen=0pt
8207
8208 %
```

\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

```
8209 \newcounter{addcolcount}
8210 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
8211 %
```

XXX.2.3 Tabular typesetting

\setmcellright Typeset (recursively) cells of display math right justified.

```
8212 \def\setmcellright #1&{\def\edlabel##1}%
8213   \let\edindex\nulledindex
8214   \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
8215     \let\Next\relax%
8216   \else\l@dcolcount=0%
8217     \let\Next=\setmcellright%
8218   \fi%
8219 \else%
8220   \disablel@dtabfeet%
```

```

8221         \stepl@dc colcount%
8222         \disable@notes%
8223         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8224         \restore@notes%
8225         \letsforverteilen%
8226         \hskip\hilfsskip$\displaystyle{#1}$%
8227         \hskip\edtabcolsep%
8228         \let\Next=\setmcellright%
8229     \fi\Next}
8230
8231 %

```

\settcellright Typeset (recursively) cells of text right justified.

```

8232 \def\settcellright #1&{\def\edlabel##1{}}%
8233     \let\edindex\nulledindex
8234     \ifx #1\\ \ifnum\l@dc colcount=0\removelastskip
8235         \let\Next\relax%
8236     \else\l@dc colcount=0%
8237         \let\Next=\settcellright%
8238     \fi%
8239 \else%
8240     \disablel@dtabfeet%
8241     \stepl@dc colcount%
8242     \disable@notes%
8243     \setbox\hilfsbox=\hbox{#1}%
8244     \restore@notes%
8245     \letsforverteilen%
8246     \hskip\hilfsskip#1%
8247     \hskip\edtabcolsep%
8248     \let\Next=\settcellright%
8249 \fi\Next}
8250 %

```

\setmcellleft Typeset (recursively) cells of display math left justified.

```

8251 \def\setmcellleft #1&{\def\edlabel##1{}}%
8252     \let\edindex\nulledindex
8253     \ifx #1\\ \ifnum\l@dc colcount=0 \let\Next\relax%
8254     \else\l@dc colcount=0%
8255         \let\Next=\setmcellleft%
8256     \fi%
8257 \else \disablel@dtabfeet%
8258     \stepl@dc colcount%
8259     \disable@notes%
8260     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8261     \restore@notes%
8262     \letsforverteilen%
8263     $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
8264     \let\Next=\setmcellleft%

```

```

8265 \fi\Next}
8266
8267 %

```

\settcclleft Typeset (recursively) cells of text left justified.

```

8268 \def\settcclleft #1&{\def\edlabel##1{}%
8269 \let\edindex\nulledindex
8270 \ifx #1\\ \ifnum\l@dc@lcount=0 \let\Next\relax%
8271 \else\l@dc@lcount=0%
8272 \let\Next=\settcclleft%
8273 \fi%
8274 \else \disablel@dtabfeet%
8275 \stepl@dc@lcount%
8276 \disable@notes%
8277 \setbox\hilfsbox=\hbox{#1}%
8278 \restore@notes%
8279 \letsforverteilen%
8280 #1\hskip\hilfsskip\hskip\edtabcolsep%
8281 \let\Next=\settcclleft%
8282 \fi\Next}
8283 %

```

\setmcellcenter Typeset (recursively) cells of display math centered.

```

8284 \def\setmcellcenter #1&{\def\edlabel##1{}%
8285 \let\edindex\nulledindex
8286 \ifx #1\\ \ifnum\l@dc@lcount=0\let\Next\relax%
8287 \else\l@dc@lcount=0%
8288 \let\Next=\setmcellcenter%
8289 \fi%
8290 \else \disablel@dtabfeet%
8291 \stepl@dc@lcount%
8292 \disable@notes%
8293 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8294 \restore@notes%
8295 \letsforverteilen%
8296 \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
8297 \hskip\edtabcolsep%
8298 \let\Next=\setmcellcenter%
8299 \fi\Next}
8300
8301 %

```

\settcclcenter Typeset (recursively) cells of text centered.

```

8302 \def\settcclcenter #1&{\def\edlabel##1{}%
8303 \let\edindex\nulledindex
8304 \ifx #1\\ \ifnum\l@dc@lcount=0 \let\Next\relax%
8305 \else\l@dc@lcount=0%

```

```

8306             \let\Next=\settcellcenter%
8307             \fi%
8308     \else    \disablel@dtabfeet%
8309             \stepl@dcolcount%
8310             \disable@notes%
8311             \setbox\hilfsbox=\hbox{#1}%
8312             \restore@notes%
8313             \letsforverteilen%
8314             \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
8315             \hskip\edtabcolsep%
8316             \let\Next=\settcellcenter%
8317     \fi\Next}
8318
8319 %

```

\NEXT₂₀ \let\NEXT=\relax

```

8321
8322 %

```

\setmrowright Typeset (recursively) rows of right justified math.

```

8323 \def\setmrowright #1\{%
8324     \ifx #1& \let\NEXT\relax
8325     \else \centerline{\setmcellright #1&\\&\\&}
8326         \let\NEXT=\setmrowright
8327     \fi\NEXT}
8328 %

```

\settroright Typeset (recursively) rows of right justified text.

```

8329 \def\settroright #1\{%
8330     \ifx #1& \let\NEXT\relax
8331     \else \centerline{\settcellright #1&\\&\\&}
8332         \let\NEXT=\settroright
8333     \fi\NEXT}
8334
8335 %

```

\setmrowleft Typeset (recursively) rows of left justified math.

```

8336 \def\setmrowleft #1\{%
8337     \ifx #1&\let\NEXT\relax
8338     \else \centerline{\setmcellleft #1&\\&\\&}
8339         \let\NEXT=\setmrowleft
8340     \fi\NEXT}
8341 %

```

\settrorleft Typeset (recursively) rows of left justified text.

```

8342 \def\settrorleft #1\{\%
8343   \ifx #1& \let\NEXT\relax
8344   \else \centerline{\settcclleft #1&\&\&\&}
8345   \let\NEXT=\settrorleft
8346   \fi\NEXT}
8347
8348 %

```

\setmrowcenter Typeset (recursively) rows of centered math.

```

8349 \def\setmrowcenter #1\{\%
8350   \ifx #1& \let\NEXT\relax%
8351   \else \centerline{\setmcellcenter #1&\&\&\&}
8352   \let\NEXT=\setmrowcenter
8353   \fi\NEXT}
8354 %

```

\settrorcenter Typeset (recursively) rows of centered text.

```

8355 \def\settrorcenter #1\{\%
8356   \ifx #1& \let\NEXT\relax
8357   \else \centerline{\settrcellcenter #1&\&\&\&}
8358   \let\NEXT=\settrorcenter
8359   \fi\NEXT}
8360
8361 %

```

\nullsetzen₆₂ \newcommand{\nullsetzen}{\%

```

8363   \stepl@dc@colcount%
8364   \l@dc@colwidth=0pt%
8365   \ifnum\l@dc@colcount=30\let\NEXT\relax%
8366   \l@dc@colcount=0\relax
8367   \else\let\NEXT\nullsetzen%
8368   \fi\NEXT}
8369
8370 %

```

\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.

```

8371 \newcommand{\edatleft}[3][\@empty]{\%
8372   \ifx#1\@empty
8373     \vbox to 10pt{\vss\hbox{\$ \left#2\vrule width0pt height #3
8374       depth 0pt \right. \$\hss}\vfil}
8375   \else
8376     \vbox to 4pt{\vss\hbox{\$#1\left#2\vrule width0pt height #3
8377       depth 0pt \right. \$}\vfil}
8378   \fi}
8379 %

```

\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.

```

8380 \newcommand{\edatright}[3][\@empty]{%
8381   \ifx#1\@empty
8382     \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
8383       depth 0pt \right#2 $\hss}\vfil}
8384   \else
8385     \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
8386       depth 0pt \right#2 #1 $\vfil}
8387   \fi}
8388
8389 %

```

\edvertline `\edvertline{ $\langle len \rangle$ }` vertical line $\langle len \rangle$ high.

```

8390 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
8391
8392 %

```

\edvertdots `\edvertdots{ $\langle len \rangle$ }` vertical dotted line $\langle len \rangle$ high.

```

8393 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
8394   {\cleaders\hbox{$\math\hbox{.}\vbox to 0.5em{ }\vfil}}}
8395
8396 %

```

\l@dtabaddcols `\l@dtabaddcols{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ }` adds the widths of the columns $\langle startcol \rangle$ through $\langle endcol \rangle$ to `\edfilldimen`. It is a \TeX style reimplementaion of the original `\@add@`.

```

8397 \newcommand{\l@dtabaddcols}[2]{%
8398   \l@dccheckstartend{#1}{#2}%
8399   \ifl@dstartendok
8400     \setcounter{addcolcount}{#1}%
8401     \@whilenum \value{addcolcount}<#2\relax \do
8402       {\advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
8403        \advance\edfilldimen by \edtabcolsep
8404        \stepcounter{addcolcount}}%
8405     \advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
8406   \fi
8407 }
8408
8409 %

```

\ifl@dstartendok `\l@dccheckstartend{ $\langle startcol \rangle$ }{ $\langle endcol \rangle$ }` checks that the values of $\langle startcol \rangle$ and $\langle endcol \rangle$ are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```

8410 \newif\ifl@dstartendok
8411 \newcommand{\l@dccheckstartend}[2]{%

```

```

8412 \l@dstartendoktrue
8413 \ifnum #1<\@ne
8414   \l@dstartendokfalse
8415   \led@err@LowStartColumn
8416 \fi
8417 \ifnum #2>30\relax
8418   \l@dstartendokfalse
8419   \led@err@HighEndColumn
8420 \fi
8421 \ifnum #1>#2\relax
8422   \l@dstartendokfalse
8423   \led@err@ReverseColumns
8424 \fi
8425 }
8426
8427 %

```

`\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.

```

8428 \newcommand*{\edrowfill}[3]{%
8429   \l@dtabaddcols{#1}{#2}%
8430   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
8431 \let\@edrowfill=\edrowfill
8432 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
8433
8434 %

```

`\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`.

```

8435 \newcommand{\leftltab}[1]{%
8436   \hb@xt@\z@{\vbox{\edtabindent%
8437     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8438
8439 %

```

`\leftrtab` `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

8440 \newcommand{\leftrtab}[2]{%
8441   #2\hb@xt@\z@{\vbox{\edtabindent%
8442     \advance\Hilfsskip by\dcoli%
8443     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8444
8445 %

```


`\leftctab` `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

8446 \newcommand{\leftctab}[2]{%
8447     \hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8448     \advance\Hilfsskip by 0.5\dcoli%
8449     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8450     \disablel@dtabfeet$\displaystyle{#2}$}%
8451     \advance\Hilfsskip by -0.5\wd\hilfsbox%
8452     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
8453     #2}
8454
8455 %

```

`\rightctab` `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

8456 \newcommand{\rightctab}[2]{%
8457     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8458     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8459     #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8460     \advance\Hilfsskip by 0.5\l@dcolwidth%
8461     \advance\Hilfsskip by -\wd\hilfsbox%
8462     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8463     \disablel@dtabfeet$\displaystyle{#1}$}%
8464     \advance\Hilfsskip by -0.5\wd\hilfsbox%
8465     \advance\Hilfsskip by \edtabcolsep%
8466     \moveright\Hilfsskip\hbox{ #2}}\hss}%
8467     }
8468
8469 %

```

`\rightltab` `\rightltab{<math>}{<text>}` for `\edaftertab` in `\ltab`.

```

8470 \newcommand{\rightltab}[2]{%
8471     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8472     \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8473     #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8474     \advance\Hilfsskip by\l@dcolwidth%
8475     \advance\Hilfsskip by-\wd\hilfsbox%
8476     \setbox\hilfsbox=\hbox{\def\edlabel##1{}%
8477     \disablel@dtabfeet$\displaystyle{#1}$}%
8478     \advance\Hilfsskip by-\wd\hilfsbox%
8479     \advance\Hilfsskip by\edtabcolsep%
8480     \moveright\Hilfsskip\hbox{ #2}}\hss}%
8481     }
8482
8483 %

```

`\rightrtab` `\rightrtab{<math>}{<text>}` for `\edaftertab` in `\rtab`.

```

8484 \newcommand{\rightrtab}[2]{%

```

```

8485 \setbox\hilfsbox=\hbox{\def\edlabel##1{}}%
8486 \disablel@dtabfeet#2}%
8487 #1\hb@xt@{z@{\vbox{\edtabindent%
8488 \advance\Hilfsskip by-\wd\hilfsbox%
8489 \advance\Hilfsskip by\edtabcolsep%
8490 \moveright\Hilfsskip\hbox{ #2}}\hss}%
8491 }
8492
8493 %

```

\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.

```

8494 \newcommand{\rtab}[1]{%
8495 \l@dnnullfills
8496 \def\edbeforetab##1##2{\lefttrtab{##1}{##2}}%
8497 \def\edaftertab##1##2{\righttrtab{##1}{##2}}%
8498 \measuretbody{#1}%
8499 \l@drestorefills
8500 \variab
8501 \setmrowright #1\\&\\%
8502 \enablel@dtabfeet}
8503
8504 %

```

\measuretbody `\measuretbody{<body>}` measures the array `<body>`.

```

8505 \newcommand{\measuretbody}[1]{%
8506 \disablel@dtabfeet%
8507 \l@dcolcount=0%
8508 \nullsetzen%
8509 \l@dcolcount=0
8510 \measuremrow #1\\&\\%
8511 \global\l@dampcount=1}
8512
8513 %

```

\rtabtext `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```

8514 \newcommand{\rtabtext}[1]{%
8515 \l@dnnullfills
8516 \measuretbody{#1}%
8517 \l@drestorefills
8518 \variab
8519 \settrrowright #1\\&\\%
8520 \enablel@dtabfeet}
8521
8522 %

```

`\measuretbody` `\measuretbody{<body>}` measures the tabular `<body>`.

```

8523 \newcommand{\measuretbody}[1]{%
8524   \disable@notes%
8525   \disablel@dtabfeet%
8526   \l@dc@colcount=0%
8527   \nullsetzen%
8528   \l@dc@colcount=0
8529   \measuretrrow #1\\&\\%
8530   \restore@notes%
8531   \global\l@dampcount=1}
8532 %
8533 %

```

`\ltab` Array with entries left justified.

```

\edbeforetab
\edaftertab
8534 \newcommand{\ltab}[1]{%
8535   \l@dnullfills
8536   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
8537   \def\edaftertab##1##2{\rightltab{##1}{##2}}%
8538   \measuretbody{#1}%
8539   \l@drestorefills
8540   \variab
8541   \setmrowleft #1\\&\\%
8542   \enablel@dtabfeet}
8543 %
8544 %

```

`\ltabtext` Tabular with entries left justified.

```

8545 \newcommand{\ltabtext}[1]{%
8546   \l@dnullfills
8547   \measuretbody{#1}%
8548   \l@drestorefills
8549   \variab
8550   \settrrowleft #1\\&\\%
8551   \enablel@dtabfeet}
8552 %
8553 %

```

`\ctab` Array with centered entries.

```

\edbeforetab
\edaftertab
8554 \newcommand{\ctab}[1]{%
8555   \l@dnullfills
8556   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
8557   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
8558   \measuretbody{#1}%
8559   \l@drestorefills
8560   \variab
8561   \setmrowcenter #1\\&\\%

```

```

8562 \enablel@dtabfeet}
8563
8564 %

```

\ctabtext Tabular with entries centered.

```

8565 \newcommand{\ctabtext}[1]{%
8566   \l@dnnullfills
8567   \measuretbody{#1}%
8568   \l@drestorefills
8569   \variab
8570   \settrrowcenter #1\\&\\%
8571   \enablel@dtabfeet}
8572
8573 %

```

\spreadtext₇₄ \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
 %

\spreadmath₇₇ \newcommand{\spreadmath}[1]{%
 \hb@xt@ \the\l@dcolwidth{\hbox{\$\displaystyle{#1}\$}\hss}}
 %

\HILFSskip More helpers.

\Hilfsskip
 \newskip\HILFSskip
 \newskip\Hilfsskip
 %

\EDTABINDENT₈₅ \newcommand{\EDTABINDENT}{%
 \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
 \else\step\l@dcolcount%
 \advance\Hilfsskip by\l@dcolwidth%
 \ifdim\l@dcolwidth=0pt\advance\hilfscount\@ne
 \else\advance\Hilfsskip by \the\hilfscount\edtabcolsep%
 \hilfscount=1\fi%
 \let\NEXT=\EDTABINDENT%
 \fi\NEXT}%
 %

\edtabindent (was \tabindent)

```

8595 \newcommand{\edtabindent}{%
8596     \l@dcolcount=0\relax
8597     \Hilfsskip=0pt%
8598     \hilfscount=1\relax
8599     \EDTABINDENT%
8600     \hilfsskip=\hsize%
8601     \advance\hilfsskip -\Hilfsskip%
8602     \Hilfsskip=0.5\hilfsskip%
8603 }%
8604
8605 %

```

\EDTAB (was \TAB)

```

8606 \def\EDTAB #1|#2|{%
8607     \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
8608     \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
8609     \advance\tabelskip -\wd\tabhilfbox%
8610     \advance\tabelskip -\wd\tabHilfbox%
8611     \unhbox\tabhilfbox\hskip\tabelskip%
8612     \unhbox\tabHilfbox}%
8613
8614 %

```

\EDTABtext (was \TABtext)

```

8615 \def\EDTABtext #1|#2|{%
8616     \setbox\tabhilfbox=\hbox{#1}%
8617     \setbox\tabHilfbox=\hbox{#2}%
8618     \advance\tabelskip -\wd\tabhilfbox%
8619     \advance\tabelskip -\wd\tabHilfbox%
8620     \unhbox\tabhilfbox\hskip\tabelskip%
8621     \unhbox\tabHilfbox}%
8622 %

```

\tabhilfbox Further helpers.

\tabHilfbox

```

8623 \newbox\tabhilfbox
8624 \newbox\tabHilfbox
8625
8626 %

```

XXX.2.4 Environments

`edarrayl edarrayc edarrayr` The ‘environment’ forms for `\ltab`, `\ctab` and `\rtab`.

```

8627 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{}
8628 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{}
8629 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{}
8630
8631 %

```

edtabularl edtabularc edtabularr The ‘environment’ forms for \ltabtext, \ctabtext and \rtabtext.

```

8632 \newenvironment{edtabularl}{\l@collect@body\ltabtext}{}
8633 \newenvironment{edtabularc}{\l@collect@body\ctabtext}{}
8634 \newenvironment{edtabularr}{\l@collect@body\rtabtext}{}
8635
8636 %

```

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      8637 \newcommand{\initnumbering@quote}{
8638   \ifnoquotation@else
\endquote   8639   \renewcommand{\quotation}{\par\leavevmode%
8640                                     \parindent=1.5em%
8641                                     \skipnumbering%
8642                                     \ifautopar%
8643                                       \vskip-\parskip%
8644                                       \else%
8645                                         \vskip\topsep%
8646                                         \fi%
8647                                       \global\leftskip=\leftmargin%
8648                                       \global\rightskip=\leftmargin%
8649                                     }
8650   \renewcommand{\endquotation}{\par%
8651                                     \global\leftskip=0pt%
8652                                     \global\rightskip=0pt%
8653                                     \leavevmode%
8654                                     \skipnumbering%
8655                                     \ifautopar%
8656                                       \vskip-\parskip%
8657                                       \else%
8658                                         \vskip\topsep%
8659                                         \fi%
8660                                     }
8661   \renewcommand{\quote}{\par\leavevmode%
8662                                     \parindent=0pt%
8663                                     \skipnumbering%
8664                                     \ifautopar%
8665                                       \vskip-\parskip%
8666                                       \else%
8667                                         \vskip\topsep%
8668                                         \fi%

```

```

8669             \global\leftskip=\leftmargin%
8670             \global\rightskip=\leftmargin%
8671         }
8672     \renewcommand{\endquote}{\par%
8673         \global\leftskip=0pt%
8674         \global\rightskip=0pt%
8675         \leavevmode%
8676         \skipnumbering%
8677         \ifautopar%
8678             \vskip-\parskip%
8679         \else%
8680             \vskip\topsep%
8681         \fi%
8682     }
8683 \fi
8684 }
8685 %

```

XXXII Section's title commands

XXXII.1 Commands to disable some feature

\ledsectnotoc The \ledsectnotoc only disables the \addcontentsline macro.

```

8686 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
8687 %

```

\ledsectnomark The \ledsectnomark only disables the \chaptermark, \sectionmark and \subsectionmark macros.

```

8688 \newcommand{\ledsectnomark}{%
8689     \let\chaptermark\@gobble%
8690     \let\sectionmark\@gobble%
8691     \let\subsectionmark\@gobble%
8692 }
8693 %

```

XXXII.2 General overview

The system of \eledxxxx commands to section text work like this:

1. When one of these commands is called, reledmac writes to an auxiliary files:
 - The section level.
 - The section title.
 - The side (when eledpar is used).
 - The pstart where the command is called.

- If we have starred version or not.
2. `reledmac` adds the title of the section to `pstart`, as normal content. This is to enable critical notes.
 3. When \LaTeX is run a other time, this file is read. That:
 - Adds the `pstart` number to a list of `pstarts` where a sectioning command is used.
 - Defines a command, the name of which contains the `pstart` number, and which calls the normal \LaTeX sectioning command.
 4. This last command is called when the `pstart` is effectively printed.

XXXII.3 `\beforeeledchapter` command

We do not define commands for `\eledsection` and related if the `noeledsec` option is loaded. We use `etoolbox` tests and not the `\ifxxx...else...\fi` structure to prevent problem of expansions with command after the `\ifxxx` which contains `\fi`. As we patch command inside this test, we need to change the category code of `#` character *before* `\notbool` statement, because the second argument is read with the standard `catcode` (read *The TeXbook* to understand when the `catcode`'s change has effect).

```
8694 \catcode`\#=12
8695 \notbool{@noeled@sec}{%
8696 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
8697 \ifl@dmemoir
8698   \newcommand\beforeeledchapter{%
8699     \clearforchapter%
8700   }
8701 \else
8702   \newcommand\beforeeledchapter{%
8703     \if@openright%
8704       \cleardoublepage%
8705     \else%
8706       \clearpage%
8707     \fi%
8708   }
8709 \fi
8710 %
```

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.


```

8711 \def\print@rightmargin@eledsection{%
8712   \if@eled@sectioning%
8713     \begingroup%
8714     \if@RTL%
8715       \let\llap\rlap%
8716       \let\leftlinenum\rightlinenum%
8717       \let\leftlinenumR\rightlinenumR%
8718       \let\l@drd@ta\l@dld@ta%
8719       \let\l@drsn@te\l@dlsn@te%
8720     \fi%
8721     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
8722     \endgroup%
8723   \fi%
8724 }%
8725
8726 \def\print@leftmargin@eledsection{%
8727   \if@eled@sectioning%
8728     \leavevmode%
8729     \begingroup%
8730     \if@RTL%
8731       \let\rlap\llap%
8732       \let\rightlinenum\leftlinenum%
8733       \let\rightlinenumR\leftlinenumR%
8734       \let\l@dld@ta\l@drd@ta%
8735       \let\l@dlsn@te\l@drsn@te%
8736     \fi%
8737     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
8738     \endgroup%
8739   \fi%
8740 }%
8741
8742 %

```

XXXII.5 Patching standard commands

`\M@sect`
`\@mem@old@ssect`
`\@makechapterhead`
`\@makechapterhead`
`\@makeschapterhead`
`\@sect`
`\@ssect`

We have to patch \LaTeX , book and memoir sectioning commands in order to:

- Disable `\edtext` inside.
- Disable page breaking (for `\chapter`).
- Add line numbers and sidenotes.

Unfortunately, Maïeul Rouquette was not able to try if memoir is loaded. That is why `eledmac` tries to define for both standard class and memoir class.

```

8743 \AtBeginDocument{%
8744
8745
8746 \pretocmd{\M@sect}

```

```

8747 {\let\old@edtext=\edtext%
8748 \let\edtext=\dummy@edtext@showlemma%
8749 }
8750 {}
8751 {}
8752
8753 \apptocmd{\M@sect}
8754 {\let\edtext=\old@edtext}
8755 {}
8756 {}
8757
8758 \patchcmd{\M@sect}
8759 { #9}
8760 { #9%
8761 \print@rightmargin@eledsection%
8762 }
8763 {}
8764 {}
8765
8766 \patchcmd{\M@sect}
8767 {\hskip #3\relax}
8768 {\hskip #3\relax%
8769 \print@leftmargin@eledsection%
8770 }
8771 {}
8772 {}
8773
8774 \patchcmd{\@mem@old@ssect}
8775 {#5}
8776 {#5%
8777 \print@leftmargin@eledsection%
8778 }
8779 {}
8780 {}
8781
8782 \patchcmd{\@mem@old@ssect}
8783 {\hskip #1}
8784 {\hskip #1%
8785 \print@rightmargin@eledsection%
8786 }
8787 {}
8788 {}
8789
8790
8791
8792 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8793 \if@eled@sectioning\else%
8794 \ifl@dprintingpages\else%
8795 \if@openright\cleardoublepage\else\clearpage\fi}%No clearpage inside a

```

```

\Pages: will keep critical notes from printing on the title page. Here for
scrbook.
    \fi%
8796 \fi%
8797 \fi%
8798 }
8799 {}
8800 {}
8801
8802 \patchcmd{\@makechapterhead}
8803 {#1}
8804 {\print@leftmargin@eledsection%
8805   #1%
8806   \print@rightmargin@eledsection%
8807 }
8808 {}
8809 {}
8810
8811 \patchcmd{\@makechapterhead}% For BIDI
8812 {\if@RTL\raggedleft\else\raggedright\fi}%
8813 {\if@eled@sectioning\else%
8814   \if@RTL\raggedleft\else\raggedright\fi%
8815 \fi%
8816 }%
8817 {}%
8818 {}%
8819
8820 \patchcmd{\@makeschapterhead}
8821 {#1}
8822 {\print@leftmargin@eledsection%
8823   #1%
8824   \print@rightmargin@eledsection%
8825 }
8826 {}
8827 {}
8828
8829 \pretocmd{\@sect}
8830 {\let\old@edtext=\edtext
8831 \let\edtext=\dummy@edtext@showlemma%
8832 }
8833 {}
8834 {}
8835
8836 \apptocmd{\@sect}
8837 {\let\edtext=\old@edtext}
8838 {}
8839 {}
8840
8841 \pretocmd{\@ssect}
8842 {\let\old@edtext=\edtext%
8843 \let\edtext=\dummy@edtext@showlemma%

```

```

8844 }
8845 {}
8846 {}
8847
8848 \apptocmd{\@ssect}
8849 {\let\edtext=\old@edtext}
8850 {}
8851 {}
8852
8853 %

```

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.

```

8854 \@ifpackageloaded{nameref}{
8855
8856   \patchcmd{\NR@sect}
8857     {#8}
8858     {#8%
8859       \print@rightmargin@eledsection%
8860     }
8861     {}
8862     {}
8863
8864   \patchcmd{\NR@sect}
8865     {\hskip #3\relax}
8866     {\hskip #3\relax%
8867       \print@leftmargin@eledsection%
8868     }
8869     {}
8870     {}
8871
8872   \patchcmd{\NR@ssect}
8873     {#5}
8874     {#5%
8875       \print@rightmargin@eledsection%
8876     }
8877     {}
8878     {}
8879
8880   \patchcmd{\NR@ssect}
8881     {\hskip #1}
8882     {\hskip #1%
8883       \print@leftmargin@eledsection%
8884     }
8885     {}
8886     {}
8887   }%
8888   {
8889     \patchcmd{\@sect}
8890       {#8}

```

```

8891     {#8%
8892     \print@rightmargin@eledsection%
8893     }
8894     {}
8895     {}
8896
8897     \patchcmd{\@sect}
8898     {\hskip #3\relax}
8899     {\hskip #3\relax%
8900     \print@leftmargin@eledsection%
8901     }
8902     {}
8903     {}
8904
8905     \patchcmd{\@ssect}
8906     {#5}
8907     {#5%
8908     \print@rightmargin@eledsection%
8909     }
8910     {}
8911     {}
8912
8913     \patchcmd{\@ssect}
8914     {\hskip #1}
8915     {\hskip #1%
8916     \print@leftmargin@eledsection%
8917     }
8918     {}
8919     {}
8920     }%
8921 }%
8922 %

```

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 `#`.

```

8923 {}}%
8924 \protect\catcode`\#=6 %Space NEEDS by \catcode
8925 %

```

\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.

```

8926 \AtBeginDocument{%
8927 \patchcmd{\chapter}{\clearforchapter}{%
8928 \if@eled@sectioning\else%
8929 \ifl@dprintingpages\else%
8930 \clearforchapter%
8931 \fi%
8932 \fi%

```

```

8933 }%
8934 {}%
8935 {}%
8936
8937 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8938   \if@eled@sectioning\else%
8939     \ifl@dprintingpages%
8940       \endgraf%
8941     \else%
8942       \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
8943   \fi%
8944 \fi%
8945 }%
8946 {}%
8947 {}%
8948 }%
8949 %

```

\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.

```

8950 \newif\if@eled@sectioning%
8951 %

```

We reopen a new `\notbool{@noeled@sec}` statement, as we will define the `\elesection` commands.

```

8952 \notbool{@noeled@sec}{%
8953 %

```

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.

```

8954 \newwrite\eled@sectioning@out
8955 %

```

\eledchapter **\eledsection** And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsubsection
8956 \newcommand{\eledchapter}[2][]{%
\eledsubsubsection
8957   \disable@familiarnotes%
\eledchapter*
8958   #2%
\eledsection*
8959   \restore@familiarnotes%
\eledsubsection*
8960   \ifledRcol%
\eledsubsubsection*

```

```

8961 \immediate\write\eled@sectioningR@out{%
8962 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
8963 }%
8964 \else%
8965 \immediate\write\eled@sectioning@out{%
8966 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{}}
8967 }%
8968 \fi%
8969 }
8970
8971 \newcommand{\eledsection}[2][{}]{%
8972 \disable@familiarnotes%
8973 #2%
8974 \restore@familiarnotes%
8975 \ifledRcol%
8976 \immediate\write\eled@sectioningR@out{%
8977 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
8978 }%
8979 \else%
8980 \immediate\write\eled@sectioning@out{%
8981 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{}}
8982 }%
8983 \fi%
8984 }
8985
8986 \newcommand{\eledsubsection}[2][{}]{%
8987 \disable@familiarnotes%
8988 #2%
8989 \restore@familiarnotes%
8990 \ifledRcol%
8991 \immediate\write\eled@sectioningR@out{%
8992 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
8993 }%
8994 \else%
8995 \immediate\write\eled@sectioning@out{%
8996 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{}}
8997 }%
8998 \fi%
8999 }
9000 \newcommand{\eledsubsubsection}[2][{}]{%
9001 \disable@familiarnotes%
9002 #2%
9003 \restore@familiarnotes%
9004 \ifledRcol%
9005 \immediate\write\eled@sectioningR@out{%
9006 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
9007 }{R}
9008 }%
9009 \else%

```

```

9009 \immediate\write\eled@sectioning@out{%
9010 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
9011 }{}{}
9012 }%
9013 \fi%
9014 }
9015
9016 \WithSuffix\newcommand\eledchapter*[2][]{%
9017 \disable@familiarnotes%
9018 #2%
9019 \restore@familiarnotes%
9020 \ifledRcol%
9021 \immediate\write\eled@sectioningR@out{%
9022 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
9023 }%
9024 \else%
9025 \immediate\write\eled@sectioning@out{%
9026 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{*}{R}
9027 }%
9028 \fi%
9029 }
9030
9031 \WithSuffix\newcommand\eledsection*[2][]{%
9032 \disable@familiarnotes%
9033 #2%
9034 \restore@familiarnotes%
9035 \ifledRcol%
9036 \immediate\write\eled@sectioningR@out{%
9037 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
9038 }%
9039 \else%
9040 \immediate\write\eled@sectioning@out{%
9041 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{*}{R}
9042 }%
9043 \fi%
9044 }
9045
9046 \WithSuffix\newcommand\eledsubsection*[2][]{%
9047 \disable@familiarnotes%
9048 #2%
9049 \restore@familiarnotes%
9050 \ifledRcol%
9051 \immediate\write\eled@sectioningR@out{%
9052 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
9053 R}
9054 }%
9055 \else%
9056 \immediate\write\eled@sectioning@out{%
9057 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL

```



```

}{}{}
9057 }%
9058 \fi%
9059 }
9060
9061 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
9062 \disable@familiarnotes%
9063 #2%
9064 \restore@familiarnotes%
9065 \ifledRcol%
9066 \immediate\write\eled@sectioningR@out{%
9067 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR
}{}{}
9068 }%
9069 \else%
9070 \immediate\write\eled@sectioning@out{%
9071 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
}{}{}
9072 }%
9073 \fi%
9074 }
9075 %

```

XXXII.7 Macros written in the auxiliary file

`\eled@chapter`
`\eled@section`
`\eled@subsection`
`\eled@subsubsection`

The sectioning macros, called in the auxiliary file. They have five arguments:

1. Optional arguments of \LaTeX sectioning command.
2. Mandatory arguments of \LaTeX sectioning command.
3. Pstart number.
4. Side: R if right, nothing if left.
5. Starred or not.

```

9076 \def\eled@chapter#1#2#3#4#5{%
9077 \ifstrempy{#4}%
9078 {%
9079 \ifstrempy{#1}%
9080 {%
9081 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}}%
9082 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}}%
9083 }%Need for \pairs, because of using parbox.
9084 {%
9085 \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}}%

```

```

9086 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
9087 {#2}}%Need for \pairs, because of using parbox.
9088 }%
9089 {%
9090 \ifstrempy{#1}%
9091 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
9092 {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#1}{#2}}}%Bug in LaTeX!
9093 }%
9094 \listcsgadd{eled@sections#5@@}{#3}%
9095 }
9096 \def\eled@section#1#2#3#4#5{%
9097 \ifstrempy{#4}%
9098 {\ifstrempy{#1}%
9099 {%
9100 \csgdef{eled@sectioning@#3#5}{\section{#2}}%
9101 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
9102 {#2}}%Need for \pairs, because of using parbox.
9103 }%
9104 {\csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
9105 \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
9106 {#1}}%Need for \pairs, because of using parbox.
9107 }%
9108 {\ifstrempy{#1}%
9109 {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
9110 {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
9111 }
9112 \listcsgadd{eled@sections#5@@}{#3}%
9113 }
9114 \def\eled@subsection#1#2#3#4#5{%
9115 \ifstrempy{#4}%
9116 {\ifstrempy{#1}%
9117 {%
9118 \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
9119 \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)
9120 }%
9121 {%
9122 \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
9123 \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)
9124 }%
9125 }%
9126 {\ifstrempy{#1}%

```

```

9127     {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
9128     {\csgdef{eled@sectioning@#3#5}{\subsection*[#1]{#2}}}%Bug in LaTeX!
9129   }
9130   \listcsgadd{eled@sections#5@@}{#3}%
9131   }
9132   \def\eled@subsubsection#1#2#3#4#5{%
9133     \ifstrempy{#4}%
9134     {\ifstrempy{#1}%
9135       {\csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
9136       {\csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
9137     }%
9138     {\ifstrempy{#1}%
9139       {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
9140       {\csgdef{eled@sectioning@#3#5}{\subsubsection*[#1]{#2}}}%Bug in
LaTeX!
9141     }
9142     \listcsgadd{eled@sections#5@@}{#3}%
9143   }
9144
9145   %

```

End of the conditional test about noeledsec option.

```

9146   }{}
9147   %

```

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.

```

9148 \def\normal@page@break{}
9149 %

```

\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.

```

9150 \def\l@prev@pb{}
9151 \def\l@prev@nopb{}
9152 %

```

`\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.

```

9153 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
9154 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
9155 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
9156 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
9157 %

```

`\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.

```

9158 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
9159 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
9160 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
9161 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
9162 %

```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

`\led@pb@setting`

```

9163 \def\led@pb@setting{before}
9164 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
9165 %

```

`\led@check@pb` The `\led@check@pb` and `\led@check@nopb` are called before or after each line. They check if a page break must occur, depending on the current line and on the content of `\l@pb`.

`\led@check@nopb`

```

9166 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
9167 \newcommand{\led@check@nopb}{%
9168   \IfStrEq{\led@pb@setting}{before}{%
9169     \xifinlist{\the\absline@num}{\l@prev@nopb}%
9170     {\numdef{\abs@prevline}{\the\absline@num-1}%
9171     \xifinlist{\abs@prevline}{\normal@page@break}%
9172     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9173     {}}%
9174   {}}%
9175   {}%
9176   {}%
9177   \IfStrEq{\led@pb@setting}{after}{%
9178     \xifinlist{\the\absline@num}{\l@prev@nopb}%
9179     \xifinlist{\the\absline@num}{\normal@page@break}%
9180     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9181     {}}%
9182 }%

```

```

9183     {}}%
9184     {}%
9185     {}%
9186 }
9187 %

```

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.

```

9188 \newcommand{\check@pb@in@verse}{%
9189   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
9190   \ifnum\page@num=\last@page@num\else%If we have change page
9191   \IfStrEq{\led@pb@setting}{before}{%
9192     \numdef{\abs@line@verse}{\the\absline@num-1}%
9193     \ledpbnum{\abs@line@verse}%
9194   }{}%
9195   \IfStrEq{\led@pb@setting}{after}{%
9196     \numdef{\abs@line@verse}{\the\absline@num-1}%
9197     \lednopbnum{\abs@line@verse}%
9198   }{}%
9199   \fi%
9200 \fi\fi\fi%
9201 }
9202 %

```

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

³⁵<http://tex.stackexchange.com/a/17138/7712>.

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

```

9203 \def\Hy@raisedlink@left#1{%
9204   \ifvmode
9205     #1%
9206   \else
9207     \Hy@SaveSpaceFactor
9208     \llap{\smash{%
9209       \begingroup
9210         \let\HyperRaiseLinkLength\@tempdima
9211         \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
9212         \HyperRaiseLinkHook
9213       \expandafter\endgroup
9214       \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
9215         \Hy@RestoreSpaceFactor
9216         #1%
9217         \Hy@SaveSpaceFactor
9218       }%
9219     }}%
9220     \Hy@RestoreSpaceFactor
9221     \penalty\@M\hskip\z@ \relax
9222   \fi
9223 }
9224 %

```

XXXVI Compatibility with eledmac

Here, we define some commands for the `eledmac-compat` option.

```

9225 \ifeledmaccompat%
9226
9227   \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
9228   \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
9229   \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
9230   \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
9231
9232   \unless\ifnocritical@
9233     \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
9234     \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
9235     \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
9236     \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
9237     \let\hsizetwocol\Xhsizetwocol
9238     \let\hsizethreecol\Xhsizethreecol
9239     \let\bhookXnote\Xbhooknote

```

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-coupure-de-ligne/801>.

```

9240 \let\boxsymlinenum\Xboxsymlinenum
9241 \let\symlinenum\Xsymlinenum
9242 \let\beforenumberinfootnote\Xbeforenumber
9243 \let\afternumberinfootnote\Xafternumber
9244 \let\beforeXsymlinenum\Xbeforesymlinenum
9245 \let\afterXsymlinenum\Xaftersymlinenum
9246 \let\inplaceofnumber\Xinplaceofnumber
9247 \let\Xlemmaseparator\lemmaseparator
9248 \let\afterlemmaseparator\Xafterlemmaseparator
9249 \let\beforelemmaseparator\Xbeforelemmaseparator
9250 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
9251 \let\txbeforeXnotes\Xtxbeforenotes
9252 \let\afterXrule\Xafterrule
9253 \let\numberonlyfirstinline\Xnumberonlyfirstinline
9254 \let\numberonlyfirstintwolines\Xnumberonlyfirstintwolines
9255 \let\nonumberinfootnote\Xnonumberinfootnote
9256 \let\pstartinfootnote\Xpstart
9257 \let\pstartinfootnoteeverytime\Xpstarteverytime
9258 \let\onlyXpstart\Xonlypstart
9259 \let\Xnonumberinfootnote\Xnonumber
9260 \let\nonbreakableafternumber\Xnonbreakableafternumber
9261 \let\maxhXnotes\Xmaxhnotes
9262 \let\beforeXnotes\Xbeforenotes
9263 \let\boxlinenum\Xboxlinenum
9264 \let\boxlinenumalign\Xboxlinenumalign
9265 \let\boxstartlinenum\Xboxstartlinenum
9266 \let\boxendlinenum\Xboxendlinenum
9267 \let\twolines\Xtwolines
9268 \let\morethantwolines\Xmorethantwolines
9269 \let\twolinesbutnotmore\Xtwolinesbutnotmore
9270 \let\twolinesonlyinsamepage\Xtwolinesonlyinsamepage
9271 \fi
9272
9273 \unless\ifnofamiliar@
9274 \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
9275 \fi
9276 \newcommandx{\parafootsep}[2][1,usedefault]{%
9277 \Xparafootsep[#1]{#2}%
9278 \parafootsepX[#1]{#2}
9279 }%
9280
9281 \newcommandx{\afternote}[2][1,usedefault]{%
9282 \Xafternote[#1]{#2}%
9283 \afternoteX[#1]{#2}%
9284 }%
9285
9286 \unless\ifnoend@
9287 \let\XendXtwolines\Xendtwolines
9288 \let\XendXmorethantwolines\Xendmorethantwolines
9289 \let\bhookXendnote\Xendbhooknote

```

```

9290 \let\boxXendlinenum\Xendboxlinenum%
9291 \let\boxXendlinenumalign\Xendboxlinenumalign%
9292 \let\boxXendstartlinenum\Xendboxstartlinenum%
9293 \let\boxXendendlinenum\Xendboxendlinenum%
9294 \let\XendXlemmaseparator\Xendlemmaseparator
9295 \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
9296 \let\XendXafterlemmaseparator\Xendafterlemmaseparator
9297 \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
9298 \fi
9299
9300 \AtBeginDocument{%
9301 \ifdef\lineref{}\let\lineref\edlineref}%
9302 }%
9303
9304
9305 \fi%
9306 %

```

</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:

<pre>I saw my friend \critext{Smith} \Afootnote{Jones C, D.}/ on Tuesday.</pre>	<pre>1 I saw my friend 2 Smith on Tuesday. 2 Smith] Jones C, D.</pre>
---	---

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

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

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

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

The second argument of the `\critext` macro, `⟨commands⟩`, is the same as the second argument to the `\edtext` macro.

It is possible to define aliases for `\critext`, which can be easier to type. You can make a single character substitute for `\critext` by saying this:

```
\catcode`\<=\active
```

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. 125 to find out why.) Aliases of the second kind may be nested without any problem.

If you really have to use `\critext` in any of the tabular or array environments, then `\edtext` must not be used in the same environment. If you use `\critext` in one of these environments then you have to issue the declaration `\usingcritext` beforehand. The declaration `\usingedtext` must be issued to revert to the default assumption that `\edtext` will be used.

Appendix A.2 Migration from ledmac to eledmac

In `eledmac`, some changes were made in the code to allow easy customization. This may cause problems for people who have already made their own. The next sections explain how to handle this.

If you have created your own series using `\addfootins` and `\addfootinsX`, you must use instead the `\newseries` command (see 6.7.1 p. 35), 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. 37) or `\Xfootnote` options (6.2.2 p. 25). 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. 51). 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. 68.

Appendix A.5 Migration to eledmac 17.1

This version changes the default setting of `\Xpstart`. Henceforth, `pstart` numbers will be printed in footnotes within the section of text where you have called `\numberpstarttrue`.

We do not see any reason to print them in the other sections. However, if you want to print the `pstart` numbers in all of the footnotes, whatever the section, without having to use `\numberpstarttrue`, you can use `\Xpstarteverytime`.

Appendix A.6 Migration to eledmac 1.21.0

Appendix A.6.1 `\Xledsetnormalparstuff` and `\ledsetnormalparstuffX`

The `\ledsetnormalparstuff` has been split into two different commands:

- `\Xledsetnormalparstuff` for critical notes;
- `\ledsetnormalparstuffX` for familiar notes.

Both commands can take an optional argument which is the series letter. If you have redefined `\ledsetnormalparstuff` or any of the commands which call them, you must change them accordingly.

Appendix A.6.2 Endnotes

In any case, delete the `.end` file before the next run.

The previous version of Eledmac had a bug: there were two spaces between the starting page number and the starting line number, but only one space between the ending page number and the ending line number.

As a matter of fact, a spurious space was added after the first `\printnnum`. This spurious space has been deleted. However, if you want to keep the previous spurious space, you may load the package with the `oldprintnnumspace` option.

If you have redefined `\endprint`, you must:

- Contact us and ask for the feature that required your hack, in order to avoid such a hack in the future.
- Use the new fifth argument.
- Add `\xdef\@currentseries{#4}` at the beginning of your own command.

Appendix A.7 Migration to eledmac 1.22.0

The `\ledinnote` command now takes a first optional argument, which is the label for the hyperreference. If you have redefined it, change your redefinition, and check whether you can avoid this redefinition by only redefining `\ledinnotemark`.

Appendix A.8 Migration to eledmac 1.23.0

You must delete the numbered auxiliary files before compiling with the new version of eledmac.

Appendix A.9 Migration from eledmac to reledmac

There are many changes in reledmac which require the user to make modifications.

Appendix A.9.1 Risk of ‘no room for a new’

The risk to obtain a ‘no room for a new something’ error is greater in reledmac than it is in eledmac. See 19.1.3 p. 70 in order to know how to limit it.

Appendix A.9.2 Multiple indices with memoir

Eledmac and ledmac used the specific indexing tools of the memoir class designed to produce multiple indices. However, eledmac could also use imakeidx or indextools tools independently of the memoir class. This system forced to maintain redundant code. Since reledmac, we use only the imakeidx or indextools tools.

Consequently: Users of memoir are invited to use indextool or imakeidx to produce multiple indices.

Appendix A.9.3 Deprecated commands and options

The table of deprecated commands and their alternatives follows. Note that the way some commands must be used may have changed. Please read the handbook.

<i>Deprecated command</i>	<i>Replaced with</i>
<code>\addfootins</code>	<code>\newseries</code>
<code>\addfootinsX</code>	<code>\newseries</code>
<code>\critext</code>	<code>\edtext</code>
<code>\falseverse</code>	<code>\newverse</code>
<code>\interparanoteglue</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\ledchapter</code>	<code>\eledchapter</code>
<code>\ledsection</code>	<code>\eledsection</code>
<code>\ledsetnormalparstuff</code>	<code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code>
<code>\ledsubsection</code>	<code>\eledsubsection</code>
<code>\ledsubsubsection</code>	<code>\eledsubsubsection</code>
<code>\noeledsec</code>	Package option <code>noeledsec</code>
<code>\noendnotes</code>	Package option <code>noendnotes</code>
<code>\pageparbreak</code>	<code>\ledpb</code>

The `ledsecnolinenumber` option has been removed, because it was related to deprecated commands.

The `oldprintnpnumspace` option has been removed too, because it was related to a historical bug. The `\usingedtext` and `\usingcritext` commands are also deprecated.

Appendix A.9.4 `\renewcommand` replaced by command

Many uses of `\renewcommand` have been replaced with uses of specific commands. Please read handbook about specific commands.

<i>Deprecated <code>\renewcommand</code></i>	<i>Replaced with</i>
<code>\@led@extranofeet</code>	<code>\newseries</code>
<code>\apprefprefixmore</code>	<code>\setapprefprefixmore</code>
<code>\apprefprefixsingle</code>	<code>\setapprefprefixsingle</code>
<code>\endstanzaextra</code>	Optional argument of <code>\&</code>
<code>\hangingsymbol</code>	<code>\sethangingsymbol</code>
<code>\ledfootinsdim</code>	<code>\Xmaxhnotes</code> and <code>\maxhnotesX</code>
<code>\parafootftmsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\notenumfont</code>	<code>\Xnotenumfont</code> , <code>\Xendnotenumfont</code> and <code>\notenumfontX</code>
<code>\notefontsetup</code>	<code>\Xnotefontsize</code> , <code>\Xendnotefontsize</code> and <code>\notefontsizeX</code>
<code>\sidenotesep</code>	<code>\setsidenotsep</code>
<code>\startstanzahook</code>	Optional argument of <code>\stanza</code>
<code>\symplinenum</code>	<code>\Xsymplinenum</code>

Appendix A.9.5 Commands the names of which have been changed

In order to help the migration from `eledmac` to `reledmac`, you may load `reledmac` with `eledmac-compat` option. However, it is advised not to, and to change the command names themselves instead. In many cases, you use only a few of them, except the `\footparagraph` command.

<i>Old command</i>	<i>New command</i>
<code>\footparagraph</code>	<code>\Xarrangement</code>
<code>\footnormal</code>	<code>\Xarrangement</code>
<code>\foottwocol</code>	<code>\Xarrangement</code>
<code>\footthreecol</code>	<code>\Xarrangement</code>
<code>\footparagraphX</code>	<code>\arrangementX</code>
<code>\footnormalX</code>	<code>\arrangementX</code>
<code>\foottwocolX</code>	<code>\arrangementX</code>
<code>\footthreecolX</code>	<code>\arrangementX</code>
<code>\afterlemmaseparator</code>	<code>\Xafterlemmaseparator</code>
<code>\afternote</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\afternumberinfootnote</code>	<code>\Xafternumber</code>
<code>\afterXrule</code>	<code>\Xafterrule</code>
<code>\afterXsymplinenum</code>	<code>\Xaftersymplinenum</code>
<code>\beforelemmaseparator</code>	<code>\Xbeforelemmaseparator</code>
<code>\beforenumberinfootnote</code>	<code>\Xbeforenumber</code>
<code>\beforeXnotes</code>	<code>\Xbeforenotes</code>
<code>\beforeXsymplinenum</code>	<code>\Xbeforesymplinenum</code>

<i>Old command</i>	<i>New command</i>
<code>\bhookXnote</code>	<code>\Xbhookendnote</code>
<code>\bhookXnote</code>	<code>\Xbhooknote</code>
<code>\boxendlinenum</code>	<code>\Xboxendlinenum</code>
<code>\boxlinenum</code>	<code>\Xboxlinenum</code>
<code>\boxlinenumalign</code>	<code>\Xboxlinenumalign</code>
<code>\boxstartlinenum</code>	<code>\Xboxstartlinenum</code>
<code>\boxsymlinenum</code>	<code>\Xboxsymlinenum</code>
<code>\boxXendlinenum</code>	<code>\Xendboxlinenum</code>
<code>\boxXendlinenumalign</code>	<code>\Xendboxlinenumalign</code>
<code>\boxXendstartlinenum</code>	<code>\boxXendstartlinenum</code>
<code>\letboxXendendlinenum</code>	<code>\Xendletboxendlinenum</code>
<code>\hsizetwocol</code>	<code>\Xhsizetwocol</code>
<code>\hsizethreecol</code>	<code>\Xhsizethreecol</code>
<code>\inplaceoflemmaseparator</code>	<code>\Xinplaceoflemmaseparator</code>
<code>\inplaceofnumber</code>	<code>\Xinplaceofnumber</code>
<code>\lemmaseparator</code>	<code>\Xlemmaseparator</code>
<code>\maxhXnotes</code>	<code>\Xmaxhnotes</code>
<code>\morethantwolines</code>	<code>\Xmorethantwolines</code>
<code>\nonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\notesXwidthliketwocolumns</code>	<code>\noteswidthliketwocolumnsX</code>
<code>\noXlemmaseparator</code>	<code>\Xnolemmaseparator</code>
<code>\numberonlyfirstinline</code>	<code>\Xnumberonlyfirstinline</code>
<code>\numberonlyfirstintwolines</code>	<code>\Xnumberonlyfirstintwolines</code>
<code>\nonbreakableafternumber</code>	<code>\Xnonbreakableafternumber</code>
<code>\onlyXpstart</code>	<code>\Xonlypstart</code>
<code>\parafootsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\pstartinfootnote</code>	<code>\Xpstart</code>
<code>\pstartinfootnoteeverytime</code>	<code>\Xpstarteverytime</code>
<code>\symlinenum</code>	<code>\Xsymlinenum</code>
<code>\twolines</code>	<code>\Xtwolines</code>
<code>\twolinesbutnotmore</code>	<code>\Xtwolinesbutnotmore</code>
<code>\twolinesonlyinsamepage</code>	<code>\Xtwolinesonlyinsamepage</code>
<code>\txtbeforeXnotes</code>	<code>\Xtxtbeforenotes</code>
<code>\XendXafterlemmaseparator</code>	<code>\Xendafterlemmaseparator</code>
<code>\XendXbeforelemmaseparator</code>	<code>\Xendbeforelemmaseparator</code>
<code>\XendXinplaceoflemmaseparator</code>	<code>\Xendinplaceoflemmaseparator</code>
<code>\XendXlemmaseparator</code>	<code>\Xendlemmaseparator</code>
<code>\XendXmorethantwolines</code>	<code>\Xendmorethantwolines</code>
<code>\XendXtwolines</code>	<code>\Xendtwolines</code>
<code>\Xnonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\lineref</code>	<code>\edlineref</code>

Appendix A.9.6 Endnotes

With `reledmac`, there is now one auxiliary file for every endnotes set (`.Aend`, `.Bend`, `.Cend` etc.). If you have overridden `\doendnotes` (which you would not have done) you must adapt your code.

Appendix A.9.7 Z Series

The ‘Z’ series of notes has been removed. Only five series are provided now by default: A, B, C, D, E.

Appendix A.9.8 Internal commands

Users who have overridden internal commands, which is wrong, must adapt according to the following. Or better, they should not override any of such commands and use `reledmac` options instead.

- If you have modified `\Xfootfmt`, note that the fourth argument is now mandatory.
- `\unvxh` has been replaced with `\Xunvxh` and `\unvxhX` with two mandatory arguments.

Appendix A.10 Migration to `reledmac` 2.1.0

`Reledmac` 2.1.0 fix some bugs when using `\Xbhooknote` and `\bhooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }.`
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` in the argument of `\Xbhooknote/\bhooknoteX`.

Appendix A.11 Migration to `reledmac` 2.1.3

`Reledmac` 2.1.3 fix an historical bug, (style in `ledmac` 0.7!) which doubled the space before the rules of paragraphed familiar footnotes. Consequently, if you use paragraphed familiar footnotes, you should maybe adapt it, playing with `\beforenotesX`.

Appendix A.12 Migration to `reledmac` 2.3.0

Before `reledmac` 2.3.0, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use optional argument of `\stanza` or `\endverse`.

Appendix A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{\langle...\rangle}` to the use of `\Xlinerangeseparator` or `/` and `\Xendlinerangeseparator` (7.2.4 p. 39).

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. 43.

`\xlineref` does not print anymore the side flag (R for right side), because it is incompatible with numerical test. Use `\xflagref` to obtain it.

The `\printlines` and `\printendlines` commands take now an eighth argument, which is the side flag. It is strongly recommended to NEVER redefine these two commands and to use the setting commands instead (or to ask for new setting commands if the actual does not answer to your needs). However, if you have done it, just change your redefinition to have a new argument.

It is strongly recommended to stop redefining `\fullstop` and to use `\Xsublinesep` instead.

Appendix A.15 Migration to reledmac 2.7.0

`\Serefonlypage` (introduced in reledmac 2.5.0) added an parenthesis after the page number. This was just an error, linked to a bad imitation of `\Serefwithpage`. That has been deleted. And so, the `\XendafterpagenumberSerefonlypage` to set it was also deleted.

`\rigidbalance` is split to two new commands: `\Xrigidbalance` for critical footnotes and `\rigidbalanceX` for familiar footnotes. If you have redefined it — but why should you have ?—, you should split your single redefinition in two redefinitions.

Appendix A.16 Migration to reledmac 2.7.2

`\Xhsize` is already defined in the `floatrow` package. It becomes `\Xwidth`, and, consequently, `\hsizeX` becomes `\widthX`.

The ancient names are temporarily maintained as aliases.

Appendix A.17 Migration to reledmac 2.8.0

Reledmac 2.8.0 fix spurious indents for paragraphed critical and familiar footnotes in `ledgroup` and `minipage`. You can re-establish the indent with `\Xparinden` and `\parindentX`.

Appendix A.18 Migration to reledmac 2.13.1

Reledmac 2.5.0 added a bug, which makes the right flag to be printed on the right side of critical footnotes, even if not explicitly requested by using `\Xlineflag`.

Version 2.13.1 solves this issue. Please use `\Xlineflag` if you want to add the right flag.

Appendix A.19 Migration to reledmac 2.18.0

After updating reledmac, and before any new compilation, you need to clean your `.aux` files, if you use `\edlabel` or related.

Appendix A.20 Migration to reledmac 2.21.0

Previously, there was a bug, which meant that the description in the handbook was incorrect. If you wrote

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

“elephant” was indexed in the main text and in the critical footnotes. With the new version of reledmac, it is indexed only in main text. If you also want to index it in critical footnotes, do

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

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>	50
<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
\@insertstanza	1
\@k	1
\@l@dttempcnta	1
\@l@dttempcntb	1
\@lab	1
\@led@testifnofoot	1
\@lemma	1
\@line@num	1
\@lock	1
\@lopL	1
\@lopR	1
\@makechapterhead	1
\@makeschapterhead	1
\@mem@extranofeet	1
\@mem@old@ssect	1
\@mpfnpos	1
\@msd	1
\@msd@c	1
\@msd@options@iffullpage	1
\@msdata@list	1
\@nl	1
\@nl@reg	1
\@opXfeet	1
\@pend	1
\@pendR	1
\@ref	1
\@ref@reg	1
\@ref@reg@parse	1
\@sect	1
\@series	1
\@set	1
\@sidenotesep	1
\@ssect	1
\@startstanza	1
\@stopmsd	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1

\@xloop	1
\@xympar	1
CLASSarticle	70
CLASSbook	70, 345
CLASSmemoir	201, 261–263, 296, 345, 365, 424, 428
CLASSscrbook	428
COMMAND*footnote	72
COMMAND\...\@footnotemark...	204
COMMAND\...d@ta	151
COMMAND\<hook	
@<series	249
COMMAND\<hookname	
<pseudoserries	251, 252
COMMAND\<type	
footfmt	191
COMMAND\@@line	182
COMMAND\@MM	168, 425
COMMAND\@Rlineflag	298, 425
COMMAND\@Serefprefix	275
COMMAND\@Serefprefixmore	275
COMMAND\@add@	335
COMMAND\@adv	109
COMMAND\@apprefprefixmore	275
COMMAND\@apprefprefixsingle	275
COMMAND\@beforeinsertofthisedtext	131, 132
COMMAND\@bsphack	264
COMMAND\@doclearpage	262, 263, 418, 428
COMMAND\@doreinfeetX	428
COMMAND\@dprintingcolumns	425
COMMAND\@edindex@hyperref	298, 299
COMMAND\@edtext@level	128
COMMAND\@esphack	264
COMMAND\@fnpos	222, 258
COMMAND\@footnotemark	201, 202, 418, 428
COMMAND\@footnotetext	201, 202, 418
COMMAND\@gobble	127, 244
COMMAND\@gobblefive	427
COMMAND\@gobblefour	424
COMMAND\@gobbleseven	246
COMMAND\@gobblethree	417
COMMAND\@h	185
COMMAND\@hangingsymbol	303
COMMAND\@iiiminipage	287, 289, 417, 428
COMMAND\@iiiminipage	287
COMMAND\@l	423
COMMAND\@l@tempcnta	153, 155, 162
COMMAND\@l@tempcntb	155
COMMAND\@l@reg	423
COMMAND\@lab	105, 264, 267, 271, 417
COMMAND\@ldunboxmpfoot	290

COMMAND\@led@extranofeet	366
COMMAND\@ledinnote@command	294, 295
COMMAND\@lemma	131, 133
COMMAND\@lock	100, 303
COMMAND\@lopL	418
COMMAND\@lopR	418
COMMAND\@makecol	258, 259, 261, 428
COMMAND\@mpfnpos	222
COMMAND\@msd	313
COMMAND\@msd@c	313
COMMAND\@msd@options@iffullpage	319
COMMAND\@msdata@list	313, 314
COMMAND\@nl	105, 106, 108–110, 120, 267, 417, 418
COMMAND\@nl@reg	106, 363, 418, 423
COMMAND\@opXfeet	418
COMMAND\@opfeetX	428
COMMAND\@opxtrafeeti	428
COMMAND\@page	108, 267
COMMAND\@pend	418
COMMAND\@pendR	418
COMMAND\@ref	106, 115–117, 120, 121, 126
COMMAND\@ref@later	117, 122
COMMAND\@ref@reg	116, 418
COMMAND\@ref@reg@parsearg	117
COMMAND\@reinserts	258–261, 428
COMMAND\@secondoftwo	72
COMMAND\@sect	348
COMMAND\@series	248
COMMAND\@set	110
COMMAND\@sidenotesep	286
COMMAND\@stopmsd	313
COMMAND\@sw	117, 135, 138, 139
COMMAND\@tag	128, 129, 133
COMMAND\@tempcnta	85
COMMAND\@tempcntb	85
COMMAND\@toksa	91
COMMAND\@toksb	91
COMMAND\@xloop	163, 164
COMMAND\@xympar	279, 428
COMMAND\Aendnote	16, 25
COMMAND\Afootfmt	168
COMMAND\Afootgroup	168
COMMAND\Afootnote	8, 16, 24, 25, 28, 130, 177, 201, 223, 239, 427
COMMAND\Afootstart	168
COMMAND\AtBeginDocument	261
COMMAND\AtEndEveryPend	19, 434
COMMAND\AtEveryPend	18, 19, 53, 145, 425, 426, 428, 434
COMMAND\AtEveryPend*	19
COMMAND\AtEveryPstart	18, 19, 53, 425, 426, 428, 431, 434
COMMAND\AtEveryPstart*	19

COMMAND\AtEveryStanza	53, 432, 434
COMMAND\AtEveryStopStanza	54, 432, 434
COMMAND\AtStartEveryPstart	19, 434
COMMAND\AtStartEveryStanza	53, 54, 434
COMMAND\BeforeEveryStopStanza	434
COMMAND\Bendnote	16, 24
COMMAND\Bfootnote	8, 16, 201, 223, 239
COMMAND\Centering	46
COMMAND\Cfootnote	201
COMMAND\Columns	86, 173
COMMAND\Dfootnote	201
COMMAND\Efootnote	201
COMMAND\Gls	64
COMMAND\Hy@raisedlink	357
COMMAND\Hy@raisedlink@left	358
COMMAND\LTR	46
COMMAND\NR	348
COMMAND\Pages	86, 259, 260
COMMAND\ProcessOptionsX	76
COMMAND\RL	45
COMMAND\RaggedLeft	46
COMMAND\RaggedRight	46
COMMAND\SEonlypage	273, 430
COMMAND\SEref	58–60, 273, 276, 431, 433
COMMAND\SErefonlypage	58–60, 369, 430
COMMAND\SErefwithpage	58–60, 273, 276, 369, 430, 432
COMMAND\Stanza	423
COMMAND\Waklam	336
COMMAND\X@doreinfeet	260, 428
COMMAND\XR@prefix	279
COMMAND\XR@test	279
COMMAND\XR@test@mac	279
COMMAND\XR@test@mac@test	279
COMMAND\XXXXXXfmt	362
COMMAND\XXXXXfmt	362
COMMAND\Xafterlemmaseparator	43, 366
COMMAND\Xafternote	47, 365, 366
COMMAND\Xafternumber	41, 366
COMMAND\Xafterrule	48, 224, 366, 423, 426
COMMAND\Xaftersymlinenum	41, 366
COMMAND\Xarrangement	37, 47, 72, 169, 170, 250, 366
COMMAND\Xarrangement@footparagraph	175
COMMAND\Xarrangement@normal	170
COMMAND\Xarrangement@paragraph	175
COMMAND\Xbeforeinserting	46
COMMAND\Xbeforelemmaseparator	43, 366
COMMAND\Xbeforenotes	48, 223, 366, 423, 426
COMMAND\Xbeforenumber	39, 41, 366
COMMAND\Xbeforesymlinenum	41, 366
COMMAND\Xbhookendnote	367

COMMAND\Xbhookgroup	48, 430, 431
COMMAND\Xbhooknote	46, 367, 368, 428, 429
COMMAND\Xboxendlinenum	42, 367, 427
COMMAND\Xboxlinenum	42, 367
COMMAND\Xboxlinenumalign	42, 367, 427
COMMAND\Xboxstartlinenum	42, 367, 427
COMMAND\Xboxsymlinenum	42, 367
COMMAND\Xcolalign	46, 426
COMMAND\Xdo@feet	428, 433
COMMAND\Xend	246
COMMAND\XendXafterlemmaseparator	367
COMMAND\XendXbeforelemmaseparator	367
COMMAND\XendXinplaceoflemmaseparator	367
COMMAND\XendXlemmaseparator	367
COMMAND\XendXmorethantwolines	367
COMMAND\XendXtwolines	367
COMMAND\Xendafterenumber	41, 429
COMMAND\Xendafterlemmaseparator	44, 367
COMMAND\Xendafternote	50, 431
COMMAND\Xendafternumber	43
COMMAND\Xendafterpagenumbe	432
COMMAND\Xendafterpagenumber	43, 59
COMMAND\XendafterpagenumberSErefonlypage	369
COMMAND\Xendaftersymlinenum	41, 43, 429
COMMAND\Xendahookinplaceofnumber	43, 429
COMMAND\Xendahooklinenum	43, 429
COMMAND\Xendbeforelemmaseparator	44, 367
COMMAND\Xendbeforelinenum	43
COMMAND\Xendbeforenumber	41, 429
COMMAND\Xendbeforepagenumber	43, 59
COMMAND\XendbeforepagenumberSErefonlypage	59
COMMAND\Xendbeforesymlinenum	41, 43, 429
COMMAND\Xendbhookinplaceofnumber	43, 429
COMMAND\Xendbhooklinenum	43, 429
COMMAND\Xendbhooknote	46
COMMAND\Xendboxendlinenum	42, 427
COMMAND\Xendboxlinenum	42, 367, 425
COMMAND\Xendboxlinenumalign	42, 367, 427
COMMAND\Xendboxstartlinenum	42, 427
COMMAND\Xendboxsymlinenum	42, 429
COMMAND\Xendhangindent	45, 429, 431
COMMAND\Xendinplaceoflemmaseparator	26, 44, 367
COMMAND\Xendinplaceofnumber	41, 428
COMMAND\Xendinplaceofpagenumber	39, 433
COMMAND\Xendinsertsep@	230
COMMAND\Xendlemmadisablefontselection	44
COMMAND\Xendlemmafont	44, 430
COMMAND\Xendlemmaseparator	26, 44, 367
COMMAND\Xendletboxendlinenum	367
COMMAND\Xendlineflag	59

COMMAND\Xendlineprefixmore	43, 59
COMMAND\Xendlineprefixsingle	43, 59
COMMAND\Xendlinangeseparator	39, 59, 189, 369, 429
COMMAND\Xendmorethantwolines	26, 40, 59, 367, 426, 427
COMMAND\Xendnonumber	40, 428
COMMAND\Xendnote	226, 245, 246, 426
COMMAND\Xendnotefontsize	45, 366
COMMAND\Xendnotenumfont	43, 44, 366
COMMAND\Xendnotes	229
COMMAND\Xendnumberonlyfirstinline	38, 429
COMMAND\Xendnumberonlyfirstintwolines	38, 429
COMMAND\Xendpagenumberonlyfirst	38, 433
COMMAND\Xendpagenumberonlyfirstifsingle	38, 433
COMMAND\Xendpagenumberonlyfirstintwo	39, 433
COMMAND\Xendparagraph	50, 423
COMMAND\Xendsep	50
COMMAND\Xendsublinesep	41, 59, 189
COMMAND\Xendsymlinenum	38, 429
COMMAND\Xendsympagenum	39, 433
COMMAND\Xendtwolines	26, 40, 59, 367, 426, 427
COMMAND\Xendtwolinesbutnotmore	40, 59, 426, 427
COMMAND\Xendtwolinesonlyinsamepage	40, 59, 426, 427
COMMAND\Xendwrapcontent	45, 432
COMMAND\Xendwraplemma	45, 432
COMMAND\Xfootfmt	368
COMMAND\Xfootgroup	174
COMMAND\Xfootins	173
COMMAND\Xfootnote	56, 62, 128, 362, 420, 424, 426, 430, 432
COMMAND\Xfootstarts	174
COMMAND\Xgroupbyline	47, 48, 161, 199
COMMAND\Xgroupbylines	433
COMMAND\Xgroupbylineseparetwolines	48
COMMAND\Xhangindent	45, 429
COMMAND\Xhsize	369, 430, 431
COMMAND\Xhsizethreecol	47, 49, 367
COMMAND\Xhsizetwocol	47, 49, 251, 367
COMMAND\Xinplaceoflemmaseparator	25, 43, 367
COMMAND\Xinplaceofnumber	41, 367, 426, 427
COMMAND\Xinsertparafootsep	179, 181
COMMAND\Xledsetnormalparstuff	364, 365, 426
COMMAND\Xlemmadisablefontselection	44
COMMAND\Xlemmafont	44, 430
COMMAND\Xlemmaseparator	43, 189, 253, 255, 257, 367
COMMAND\Xlineflag	59, 369, 370, 432
COMMAND\Xlinangeseparator	39, 59, 189, 369, 429
COMMAND\Xmaxhnotes	49, 70, 72, 366, 367, 423, 425
COMMAND\Xmorethantwolines	25, 39, 40, 59, 367, 425
COMMAND\Xnoindent	429
COMMAND\Xnolemmaseparator	43, 257, 367
COMMAND\Xnonbreakableafternumber	41, 367, 421

COMMAND\Xnonumber	40, 367
COMMAND\Xnonumberinfootnote	367
COMMAND\Xnotefontsize	45, 366
COMMAND\Xnotefontsize@⟨s⟩	180, 184, 185
COMMAND\Xnotenumfont	44, 366
COMMAND\Xnoteswidthliketwocolumns	49, 424
COMMAND\Xnumberonlyfirstinline	38, 39, 47, 102, 191, 252, 253, 255, 367, 420, 425, 434
COMMAND\Xnumberonlyfirstintwolines	38, 48, 367, 420, 434
COMMAND\Xonlypstart	40, 367, 420, 425
COMMAND\Xpagelinesep	41, 433
COMMAND\Xparafootsep	47, 103, 366, 367, 432, 434
COMMAND\Xparafootsep@series	179
COMMAND\Xparinden	369
COMMAND\Xparindent	45, 426, 429, 431
COMMAND\Xprenotes	48, 224, 432
COMMAND\Xprenotes@	173, 224, 420
COMMAND\Xpstart	40, 364, 367, 420, 425
COMMAND\Xpstarteverytime	40, 364, 367, 425
COMMAND\Xragged	47
COMMAND\Xrigidbalance	182, 369, 430
COMMAND\Xstanza	40, 54
COMMAND\Xstanzaseparator	40
COMMAND\Xstorelineinfo	191
COMMAND\Xsublinesep	22, 41, 59, 189, 369
COMMAND\Xsublinesepside	22, 41
COMMAND\Xsymlinenum	38, 47, 366, 367, 427, 434
COMMAND\Xtextbeforenotes	164
COMMAND\Xtoendnotes	27, 246
COMMAND\Xtwolines	25, 39, 40, 59, 196, 197, 251, 367, 425
COMMAND\Xtwolinesappref	251
COMMAND\Xtwolinesbutnotmore	40, 59, 367, 426
COMMAND\Xtwolinesbutnotmoreappref	252
COMMAND\Xtwolinesonlyinsamepage	40, 59, 367, 426
COMMAND\Xtxtbeforenotes	48, 367, 432, 433, 435
COMMAND\Xtxtbeforenotesonlyonce	48, 435
COMMAND\Xunvxh	177, 368
COMMAND\Xwidth	49, 369, 431
COMMAND\Xwrapcontent	45, 432
COMMAND\Xwraplemma	45, 46, 432
COMMAND\&	366
COMMAND\⟨XXX⟩vfootnote	199
COMMAND\absline@num	99, 152
COMMAND\accent	127
COMMAND\actionlines@list	100, 154
COMMAND\actions@list	100
COMMAND\add@Xgroupbyline	161
COMMAND\add@inserts	100, 160, 161
COMMAND\add@inserts@next	160, 161
COMMAND\add@msd@	313
COMMAND\add@msdata	313, 314

COMMAND\add@msdata@firstlineofpage	316
COMMAND\add@msddata	313
COMMAND\add@penalties	152, 162
COMMAND\addcontentsline	343
COMMAND\addfootins	362, 365
COMMAND\addfootinsX	362, 365
COMMAND\addtoendnotes	246
COMMAND\advancelabel@refs	266
COMMAND\advanceline	23, 102, 109, 123, 428
COMMAND\affixlin@num	286
COMMAND\affixline@num	155, 157, 159, 418
COMMAND\affixpstart@num	159
COMMAND\afterXrule	366
COMMAND\afterXsymlinenum	366
COMMAND\afterenumber	41
COMMAND\aftergroup	126, 130
COMMAND\afterlemmaseparator	366
COMMAND\afternote	366
COMMAND\afternoteX	47, 365, 366
COMMAND\afternumberinfootnote	366
COMMAND\afterruleX	48, 423, 426
COMMAND\applabel	58, 268, 269, 426, 432
COMMAND\appref	56, 58–60, 273, 276, 430, 431
COMMAND\apprefprefixmore	58, 366
COMMAND\apprefprefixsingle	58, 366
COMMAND\apprefwithpage	58–60, 273, 276, 427, 430
COMMAND\arrangementX	37, 72, 205, 250, 366
COMMAND\arrangementX@normal	210
COMMAND\article	15
COMMAND\at@every@pend	145
COMMAND\autopar	18, 141, 146, 147, 220, 419, 421, 422, 426
COMMAND\ballast	71
COMMAND\ballast@count	152, 162
COMMAND\baselineskip	37, 176, 180
COMMAND\beforeXnotes	366
COMMAND\beforeXsymlinenum	366
COMMAND\beforeeledchapter	10, 68, 344
COMMAND\beforeinsertingX	46
COMMAND\beforelemmaseparator	366
COMMAND\beforenotesX	48, 368, 422, 423, 426
COMMAND\beforenumberinfootnote	366
COMMAND\begin	321
COMMAND\beginnumbering	16, 17, 19, 20, 86, 87, 90, 99, 103, 104, 119, 146, 226, 312, 420, 423, 427, 428, 433, 434
COMMAND\bf	420
COMMAND\bfseries	44, 420
COMMAND\bhookXnote	367
COMMAND\bhookgroupX	48, 430
COMMAND\bhooknoteX	46, 368, 428, 429
COMMAND\body	304

COMMAND\bodyfootmarkA	34
COMMAND\book	15
COMMAND\boxXendlinenum	367
COMMAND\boxXendlinenumalign	367
COMMAND\boxXendstartlinenum	367
COMMAND\boxendlinenum	367
COMMAND\boxlinefootnote	193
COMMAND\boxlinenum	367
COMMAND\boxlinenumalign	367
COMMAND\boxstartlinenum	367
COMMAND\boxsymlinenum	367
COMMAND\break	38, 177
COMMAND\brokenpenalty	162
COMMAND\centering	46
COMMAND\ch@ck@l@ck	418
COMMAND\ch@cksub@l@ck	157, 418
COMMAND\chapter	67, 345, 349, 423, 426, 428, 432
COMMAND\chaptermark	343
COMMAND\check@pb@in@verse	357
COMMAND\colalignX	46, 426
COMMAND\collect@body	321
COMMAND\color	432
COMMAND\colorbox	72
COMMAND\columns	49
COMMAND\columnwidth	176, 424
COMMAND\command names	251, 252
COMMAND\copyright	127
COMMAND\correct@Xfootins@box	425
COMMAND\correct@footinsX@box	425
COMMAND\count	183
COMMAND\critex	419
COMMAND\critext	133, 361, 362, 365
COMMAND\csname	77, 136, 137
COMMAND\csquotes	243
COMMAND\ctab	337, 341
COMMAND\ctabtext	342
COMMAND\dc col	330
COMMAND\def	74
COMMAND\detokenize	137
COMMAND\dimen	183
COMMAND\dimexpr	49
COMMAND\discretionary	177
COMMAND\displaywidowpenalty	162
COMMAND\do@Xfeet	259, 418, 428, 433
COMMAND\do@actions	152–154, 418
COMMAND\do@actions@fixedcode	418
COMMAND\do@actions@next	153
COMMAND\do@ballast	152, 162
COMMAND\do@feet@custom@order	258
COMMAND\do@insidelinehook	421

COMMAND\do@line	100, 125, 144, 147, 150, 151, 160, 162, 303, 418, 419, 421, 423
COMMAND\do@linehook	418
COMMAND\do@lockoff	101
COMMAND\do@lockon	101
COMMAND\dodoreintrafeet	417
COMMAND\doendnotes	26, 230, 368, 427, 434
COMMAND\doendnotesbysection	26, 230, 246, 427, 434
COMMAND\doennotes	434
COMMAND\doinsidelinehook	24, 424
COMMAND\dolinehook	24, 424
COMMAND\doreintrafeeti	428
COMMAND\doreintrafeetii	428
COMMAND\doxtrafeet	258, 417
COMMAND\doxtrafeeti	428
COMMAND\doxtrafeetii	428
COMMAND\dummy@ref	126
COMMAND\edaftertab	66, 336, 337
COMMAND\edatleft	66, 334
COMMAND\edatright	66, 67, 335
COMMAND\edbforetab	66, 336, 337
COMMAND\edfilldimen	335
COMMAND\edfont@info	133
COMMAND\edgls	64, 293
COMMAND\edglsadd	434
COMMAND\edgls...	431
COMMAND\edindex	62–64, 293, 294, 297, 299, 325, 421, 424, 425, 428, 429, 433–435
COMMAND\edindexlab	63
COMMAND\edlabel	55–58, 60, 127, 264, 266, 267, 270, 271, 278, 293, 325, 358, 370, 417, 420–422, 425, 430
COMMAND\edlabelE	58, 269
COMMAND\edlabelS	57, 269
COMMAND\edlabelSE	58
COMMAND\edlineref	55, 56, 264, 367, 425, 427, 430, 434
COMMAND\edmakelabel	57, 278
COMMAND\edpageref	55, 56, 264, 270, 278
COMMAND\edrowfill	336
COMMAND\edsublineref	55
COMMAND\edtabcolsep	329
COMMAND\edtext	6, 24, 25, 27, 28, 30–32, 50, 56–58, 62, 64, 72, 100, 115–117, 120, 121, 125–133, 135, 136, 138–140, 268, 269, 272, 325, 326, 345, 358, 361, 362, 365, 417, 419, 421, 423–427, 432–435
COMMAND\edtext@level	427
COMMAND\edtextlater	118
COMMAND\edvertdots	67, 335
COMMAND\edvertline	67, 335
COMMAND\elechapter	68
COMMAND\eled@sectioning@out	350
COMMAND\eledchapter	68, 365, 424, 428
COMMAND\eledchapter*	68
COMMAND\eledmac@error	417

COMMAND\eledsection	7, 16, 68, 126, 150, 344, 365, 426, 434
COMMAND\eledsection*	68
COMMAND\eledsubsection	68, 365
COMMAND\eledsubsection*	68
COMMAND\eledsubsubsection	68, 365
COMMAND\eledsubsubsection*	68
COMMAND\eledxxx	10, 68, 350, 423
COMMAND\eledxxxx	343
COMMAND\elesection	350
COMMAND\else	292, 344
COMMAND\empty	85, 156, 264
COMMAND\end	320, 321
COMMAND\end@lemmas	126
COMMAND\endashchar	189
COMMAND\endgraf	144, 179, 220
COMMAND\endlock	22, 101, 124, 308
COMMAND\endminipage	287, 289, 417, 428
COMMAND\endmsdata	34
COMMAND\endnotes	426, 430
COMMAND\endnumbering	16, 17, 20, 86, 87, 89, 418, 427, 433, 434
COMMAND\endprint	226, 229, 246, 364
COMMAND\endstanzaextra	366
COMMAND\endsub	22, 101, 122
COMMAND\endverse	368
COMMAND\everypar	146
COMMAND\extensionchars	70, 87
COMMAND\externaldocument	60, 278
COMMAND\f@x@l@cks	418
COMMAND>falseverse	365, 421, 423
COMMAND\fi	344
COMMAND\firstlinenum	21, 155, 419
COMMAND\firstsublinenum	21, 419
COMMAND\fix@page	106, 108, 418
COMMAND\flag@end	120, 121, 132, 423
COMMAND\flag@end@RTL	121
COMMAND\flag@end@later	122
COMMAND\flag@start	120, 121, 132, 423, 424
COMMAND\flag@start@RTL	121
COMMAND\flag@start@later	122
COMMAND\flagstanza	54
COMMAND\floatingpenalty	168, 425
COMMAND\flush@notes	163
COMMAND\fnpos	36, 222, 422, 433
COMMAND\footfmt	168, 170
COMMAND\footfmt...	205
COMMAND\footfootmarkA	34
COMMAND\footfudgefactor	177
COMMAND\footfudgefiddle	71, 175, 176, 417
COMMAND\footgroup	168
COMMAND\footins	173

COMMAND\footnormal	251, 366, 417
COMMAND\footnormalX	366
COMMAND\footnote	34, 71, 201, 202, 363, 418
COMMAND\footnote@lang	189
COMMAND\footnoteA	16, 34
COMMAND\footnoteB	16
COMMAND\footnoteC	24
COMMAND\footnoteE	34
COMMAND\footnoteX	8, 35, 243, 244, 432
COMMAND\footnoteX@reading	244
COMMAND\footnoteXmark	244, 434
COMMAND\footnoteXmk	257
COMMAND\footnoteXtext	244, 434
COMMAND\footnote⟨X⟩	127
COMMAND\footnote⟨X⟩mark	35
COMMAND\footnote⟨X⟩mk	35
COMMAND\footnote⟨X⟩nomk	35
COMMAND\footnote⟨X⟩text	35
COMMAND\footnotelang@lua	167
COMMAND\footnotelang@poly	167
COMMAND\footnotemark	35, 244
COMMAND\footnoteoption@	166, 429
COMMAND\footnoterule	183
COMMAND\footnotesize	45
COMMAND\footnotetext	35, 244
COMMAND\footparagraph	176, 251, 366, 423
COMMAND\footparagraphX	215, 366, 423
COMMAND\footsplitskips	418, 425
COMMAND\footstart	168, 173, 183
COMMAND\footstrut	179
COMMAND\footthreecol	366
COMMAND\footthreecolX	366, 427
COMMAND\foottwocol	366
COMMAND\foottwocolX	366, 427
COMMAND\foreignlanguage	45
COMMAND\fullstop	369
COMMAND\get@edindex@hyperref	298
COMMAND\get@edindex@ledinnote@command	294
COMMAND\get@fnmark	202
COMMAND\get@index@command	422
COMMAND\get@linelistfile	418
COMMAND\get@thisfootnote	209
COMMAND\getline@num	151, 153
COMMAND\gl@p	92
COMMAND\global	106
COMMAND\globaldefs	106
COMMAND\gls	64, 301
COMMAND\hangindentX	45, 426, 429
COMMAND\hangingsymbol	366, 419
COMMAND\hbox	177

COMMAND\hfill	422
COMMAND\hidenumbering	23, 114, 426
COMMAND\hidenumberingonleftpage	23, 114, 432
COMMAND\hidenumberingonrightpage	23, 114, 432
COMMAND\hline	64
COMMAND\hrulefill	336
COMMAND\hsize	38, 173, 176, 178, 184, 187, 221, 418, 424
COMMAND\hsizeX	369, 430, 431
COMMAND\hsizethreecol	367
COMMAND\hsizethreecolX	47, 49
COMMAND\hsizetwocol	367
COMMAND\hsizetwocolX	47, 49
COMMAND\hyperlinkR	298
COMMAND\hyperlinkformat	298
COMMAND\hyperlinkformatR	298
COMMAND\if@RTL	78, 130
COMMAND\if@edtext@	424, 427
COMMAND\if@eled@sectioning	350
COMMAND\if@firstlineofpage	78
COMMAND\if@firstlineofpageR	78
COMMAND\if@msd@options@fullpage	319
COMMAND\if@msdata@insertedfrompreviouspage	316
COMMAND\if@nobreak	145
COMMAND\if@noneed@Footnote	120
COMMAND\ifXnote@	86
COMMAND\ifbypage@	92
COMMAND\ifbypage@R	92
COMMAND\ifbypstart@	92
COMMAND\ifbypstart@R	92
COMMAND\iffirst@linenum@out@	118, 119
COMMAND\ifndtl@innote	86
COMMAND\ifndtl@notenumber	86
COMMAND\ifinserthangingsymbol	303
COMMAND\ifinstanza	303
COMMAND\ifstwofollowinglines	197
COMMAND\ifl@d@Xmorethantwolines	194, 425
COMMAND\ifl@d@Xtwolines	194
COMMAND\ifl@d@dash	194
COMMAND\ifl@d@elin	194
COMMAND\ifl@d@esl	194
COMMAND\ifl@d@pnum	194
COMMAND\ifl@d@ssub	194
COMMAND\ifl@dend@X	245
COMMAND\ifl@dmemoir	417
COMMAND\ifl@dpaging	424
COMMAND\ifl@dpairing	86, 419
COMMAND\ifl@dprintingpages	425
COMMAND\ifl@dskipnumber	155
COMMAND\ifl@dstartendok	335
COMMAND\ifl@imakeidx	78

COMMAND\ifledRcol	86, 419
COMMAND\ifledRcol@	86, 423
COMMAND\iflemmacommand@	424
COMMAND\ifnoend@	231
COMMAND\ifnoledgroup@	292
COMMAND\ifnoteschanged@	102
COMMAND\ifnumberedpar@	141
COMMAND\ifnumbering	87, 90
COMMAND\ifnumberingR	86, 419
COMMAND\ifnumberline	132, 155
COMMAND\ifpst@rted	419
COMMAND\ifpst@rtedL	87
COMMAND\ifseriesbefore	249
COMMAND\ifstopmsdata@inserted@	312
COMMAND\ifsublines@	99, 111
COMMAND\iftrue	427
COMMAND\ifvmode	266
COMMAND\ifxxx	344
COMMAND\ignorespaces	130
COMMAND\imki@wrindexentry	78
COMMAND\immediate	118, 119, 225
COMMAND\indent	18, 146, 368
COMMAND\index	300, 301, 433
COMMAND\indtl@wrindexentry	78
COMMAND\initnumbering@quote	342, 428
COMMAND\initnumbering@reg	418
COMMAND\initnumbering@sectcmd	428
COMMAND\inplaceoflemmaseparator	367
COMMAND\inplaceofnumber	367
COMMAND\insert	132, 160, 168, 170, 171, 185, 199, 205
COMMAND\insert@Xtxtbeforenotes	164, 185
COMMAND\insert@count	115, 120, 130
COMMAND\insert@countR	130
COMMAND\insert@msdata	313, 319
COMMAND\insert@txtbeforenotesX	164
COMMAND\insertthangingsymbol	422
COMMAND\insertlines@list	100, 115, 116
COMMAND\insertparafootsepX	219
COMMAND\inserts@list	125, 141, 160, 177
COMMAND\interAfootnotelinepenalty	419
COMMAND\interfootnotelinepenalty	419
COMMAND\interlinepenalty	168
COMMAND\interparanoteglue	365
COMMAND\justifying	46
COMMAND\l@advance@parledegroupp@beforenormalnotes	428
COMMAND\l@d@@wrindexhyp	424
COMMAND\l@d@add	134
COMMAND\l@d@end	226, 245
COMMAND\l@d@nums	129, 132–134, 194
COMMAND\l@d@section	226

COMMAND\l@d@set	110, 124
COMMAND\l@dampcount	327
COMMAND\l@dbfnote	202, 418
COMMAND\l@dcheckstartend	335
COMMAND\l@dchset@num	110
COMMAND\l@dcolcount	327, 328
COMMAND\l@dcollect@@body	321
COMMAND\l@dcollect@body	320
COMMAND\l@dcsnote	423
COMMAND\l@dcsnotetext	151, 283
COMMAND\l@dcsnotetext@l	151, 283
COMMAND\l@dcsnotetext@r	151, 283
COMMAND\l@ddodoreintrafeet	259, 417
COMMAND\l@ddoxtrafeet	259, 417
COMMAND\l@demptyd@ta	419
COMMAND\l@dend@close	225
COMMAND\l@dend@open	225
COMMAND\l@dend@stuff	226
COMMAND\l@denvbody	321
COMMAND\l@dfeetbeginmini	417
COMMAND\l@dfeetendmini	417
COMMAND\l@dgetline@margin	419
COMMAND\l@dgetlock@disp	419
COMMAND\l@dgetref@num	271, 272
COMMAND\l@dgetsidenote@margin	280, 419
COMMAND\l@dgobbeloptarg	424
COMMAND\l@dgobblearg	424
COMMAND\l@dgobbleoptarg	325
COMMAND\l@dlabel@parse	271, 272
COMMAND\l@dld@ta	155, 157
COMMAND\l@dlp@rbox	285
COMMAND\l@dlsn@te	419
COMMAND\l@dlsnote	423
COMMAND\l@dmake@labels	266, 267, 279
COMMAND\l@dmake@labelsR	279
COMMAND\l@dnumpsstartsL	87, 419
COMMAND\l@dp@rsefootspec	194
COMMAND\l@dp@rsefootspec	194
COMMAND\l@dpush@begins	321
COMMAND\l@drd@ta	155, 157
COMMAND\l@dref@undefined	271
COMMAND\l@drsn@te	419
COMMAND\l@drsnote	423
COMMAND\l@dtabaddcols	335
COMMAND\l@dtabnoexpands	417
COMMAND\l@dumboxmpfoot	428
COMMAND\l@dunboxmpfoot	419
COMMAND\l@dzeropenalties	419, 424
COMMAND\l@pb	356
COMMAND\l@prev@nopb	355

COMMAND\l@prev@pb	355
COMMAND\l@reg	363
COMMAND\label	19, 57, 60, 63, 264, 265, 272
COMMAND\label@refs	264
COMMAND\labelstarttrue	19, 420
COMMAND\labelref@list	264, 267
COMMAND\language	177
COMMAND\last@page@num	418
COMMAND\lastbox	146
COMMAND\lastskip	123
COMMAND\latex@makecol	261
COMMAND\leavevmode	18, 146
COMMAND\led@check@nopb	356
COMMAND\led@check@pb	356
COMMAND\led@nopb	356, 357
COMMAND\led@nopbnum	356
COMMAND\led@pb	356, 357
COMMAND\led@pb@macro	356
COMMAND\led@pbnum	356
COMMAND\led@reinit@index@fornote	301
COMMAND\led@set@index@fornote	300
COMMAND\ledRflag	298
COMMAND\ledchapter	365, 421
COMMAND\ledfootinsdim	366
COMMAND\ledinnernote	60, 281, 423, 432
COMMAND\ledinnote	295, 364, 427
COMMAND\ledinnotemark	62, 364, 426
COMMAND\ledleftnote	60, 281
COMMAND\ledlinenum	98, 419
COMMAND\ledllfill	151
COMMAND\ledlsnotefontsetup	432
COMMAND\ledlsnotesep	61
COMMAND\ledlsnotewidth	61
COMMAND\lednopb	69, 356
COMMAND\lednopbinverse	357
COMMAND\lednopbinversetrue	53, 69
COMMAND\lednopbnum	356
COMMAND\ledouternote	60, 281, 423, 432
COMMAND\ledpb	69, 356, 365
COMMAND\ledpbnum	356
COMMAND\ledpbsetting	69, 356, 429
COMMAND\ledrightnote	60, 281
COMMAND\ledrsnotefontsetup	432
COMMAND\ledrsnotesep	61
COMMAND\ledrsnotewidth	61
COMMAND\ledsection	365
COMMAND\ledsectnomark	343
COMMAND\ledsectnotoc	343
COMMAND\ledsetnormalparstuff	364, 365, 426
COMMAND\ledsetnormalparstuff@common	220

COMMAND\ledsetnormalparstuffX	364, 365, 426
COMMAND\ledsidenote	60, 281, 283
COMMAND\ledsubsection	365
COMMAND\ledsubsubsection	365
COMMAND\ledxxx	423
COMMAND\left	66
COMMAND\leftctab	337
COMMAND\leftheadline	98
COMMAND\leftlinenum	22, 97, 417, 419
COMMAND\leftltab	336
COMMAND\leftnoteupfalse	61
COMMAND\leftpstartnum	159
COMMAND\leftrtab	336
COMMAND\leftsidenote	283
COMMAND\leftskip	173, 176, 177
COMMAND\lemma	3, 25, 27, 28, 30–32, 125, 129, 130, 133, 135, 361, 419, 420, 427, 428, 430
COMMAND\lemmaseparator	367
COMMAND\let	28, 50, 127, 132, 244, 261, 308, 417
COMMAND\letboxXendendlinenum	367
COMMAND\line	182, 185
COMMAND\line@list	100, 116, 132
COMMAND\line@list@stuff	87, 103, 104, 119, 417, 419
COMMAND\line@list@version	106
COMMAND\line@margin	94, 157, 280
COMMAND\line@num	99, 101, 155, 417
COMMAND\line@set	134
COMMAND\lineation	21, 93
COMMAND\linebreak	38
COMMAND\linenum	25, 27, 28, 56–58, 125, 133, 134, 270, 272, 278, 361, 431
COMMAND\linenum@out	118, 264, 267
COMMAND\linenumberlist	21, 85, 156, 417
COMMAND\linenumberstyle	23, 97, 417, 434
COMMAND\linenumincrement	21, 419
COMMAND\linenummargin	21, 94, 280
COMMAND\linenumr@p	97, 417, 419
COMMAND\linenumrep	97, 419
COMMAND\linenumsep	22, 61, 98, 281
COMMAND\linerangesep@	257
COMMAND\lineref	264, 270, 278, 367, 425
COMMAND\list@clear	91
COMMAND\list@clearing@reg	419
COMMAND\list@create	91
COMMAND\lock@disp	96
COMMAND\lock@off	113
COMMAND\lock@on	112
COMMAND\lockdisp	22, 96
COMMAND\loop	163, 164, 304
COMMAND\ltab	336, 337, 341
COMMAND\ltabtext	342
COMMAND\m@mmf@prepare	202

COMMAND\makeatletter	151
COMMAND\makehboxofhboxes	178, 180
COMMAND\makeindex	62, 297
COMMAND\makelabel	278
COMMAND\managestanza@modulo	305
COMMAND\marginpar	60, 71, 279, 280, 418
COMMAND\marginparwidth	61, 281
COMMAND\markboth	150
COMMAND\mathchardef	304
COMMAND\maxhXnotes	367
COMMAND\maxhnotesX	49, 72, 366, 422, 423, 425–427
COMMAND\maxlinesinpar@list	103
COMMAND\measurebody	338
COMMAND\measuretbody	339
COMMAND\memorybreak	20
COMMAND\morenoexpands	72, 125, 127
COMMAND\morethantwolines	367
COMMAND\mpfnpos	36, 222, 422, 433
COMMAND\mpnormalfootgroup	418
COMMAND\mpnormalvfootnote	418
COMMAND\msdata	33, 34, 311–313, 434
COMMAND\msdataposition	34
COMMAND\multfootsep	35, 201
COMMAND\multiplefootnotemarker	201
COMMAND\musixtex	423
COMMAND\n@num	419, 426
COMMAND\n@num@ref	426
COMMAND\new@line	120, 418
COMMAND\new@series	127
COMMAND\newcommand	28, 74, 201, 267
COMMAND\newcommandx	28, 29
COMMAND\newhookarg@specific	257
COMMAND\newhookcommand@series	251, 252, 426
COMMAND\newhookcommand@series@reload	252
COMMAND\newhookcommand@toggle@reload	252, 424
COMMAND\newhooktoggle@series	252, 426
COMMAND\newhooktoggle@specific	257
COMMAND\newif	426
COMMAND\newline	38
COMMAND\newlinechar	245
COMMAND\newseries	35, 362, 365, 366
COMMAND\newseries@	236, 237, 250
COMMAND\newverse	53, 54, 365, 423, 434
COMMAND\next	304
COMMAND\next@action	104
COMMAND\next@actionline	104
COMMAND\next@insert	161
COMMAND\nl@regR	106
COMMAND\no@expands	72, 127, 133, 244, 417
COMMAND\noXlemmaseparator	367

COMMAND\nobreak	193
COMMAND\nocritical	237
COMMAND\noeledsec	68, 365
COMMAND\noendnotes	365
COMMAND\noexpand	363
COMMAND\nofamiliar	254
COMMAND\noindent	18, 19, 53, 54, 146, 429
COMMAND\noindentX	429
COMMAND\nomk@	257
COMMAND\nonbreakableafternumber	367
COMMAND\nonnumberinfootnote	367
COMMAND\norelax	51
COMMAND\normal@footnotemarkX	205
COMMAND\normal@page@break	355
COMMAND\normal@pars	220
COMMAND\normalbfnoteX	419
COMMAND\normalbodyfootmarkX	205
COMMAND\normalfont	433
COMMAND\normalfootfmt	50, 172, 179, 189, 226, 431
COMMAND\normalfootfmtX	206
COMMAND\normalfootfootmarkX	206
COMMAND\normalfootgroup	174
COMMAND\normalfootgroupX	207
COMMAND\normalfootnoterule	169
COMMAND\normalfootstart	173, 176
COMMAND\normalfootstartX	207
COMMAND\normalvfootnote	170, 172
COMMAND\normalvfootnote@inserted	171, 172
COMMAND\normalvfootnoteX	205
COMMAND\notbool	344
COMMAND\notfontsetup	366
COMMAND\notfontsizeX	45, 366
COMMAND\notenumfont	366
COMMAND\notenumfontX	44, 366
COMMAND\notesXwidthliketwocolumns	367
COMMAND\noteswidthliketwocolumnsX	49, 367, 424, 426
COMMAND\num@lines	141, 162
COMMAND\numberlinefalse	20
COMMAND\numberlinetrue	20
COMMAND\numberonlyfirstinline	249, 367
COMMAND\numberonlyfirstintwolines	367
COMMAND\numberpstartfalse	19
COMMAND\numberpstarttrue	19, 40, 364, 420, 428, 434
COMMAND\numberstanza	40
COMMAND\numberstanzafalse	54
COMMAND\numberstanzatrue	54
COMMAND\numlabfont	22, 50, 98
COMMAND\none@line	141
COMMAND\nonehalfspacing	429
COMMAND\nonlyXpstart	367

COMMAND\onlysideX	243
COMMAND\page@action	101, 111
COMMAND\page@start	101, 419
COMMAND\pagecontents	101
COMMAND\pageparbreak	365
COMMAND\pageref	57, 270
COMMAND\par	27, 38, 146, 220
COMMAND\par@line	141, 162
COMMAND\para@footgroup	176
COMMAND\para@footgroupX	218
COMMAND\para@footsetup	176, 417
COMMAND\para@footsetupX	216, 417, 424
COMMAND\para@vfootnoteX	217
COMMAND\parafootfmt	178, 179, 431
COMMAND\parafootfmtX	218
COMMAND\parafootftm	181
COMMAND\parafootftmX	219
COMMAND\parafootftmsep	366
COMMAND\parafootsep	367, 422, 427
COMMAND\parafootsepX	47, 103, 366, 367, 432
COMMAND\parafootstart	176
COMMAND\parafootstartX	216
COMMAND\paravfootnote	177, 180
COMMAND\parfillskip	179
COMMAND\parindent	429
COMMAND\parindentX	45, 369, 429, 431
COMMAND\parshape	71
COMMAND\parskip	19, 53, 54, 146
COMMAND\pausenumbering	20, 89, 90, 104, 105, 147, 422, 424, 431
COMMAND\penalty	179
COMMAND\pend	2, 7, 17–21, 23, 68, 69, 123, 125, 128, 134, 141–147, 159, 160, 363, 422, 423, 432–434
COMMAND\preXnotes	426, 432
COMMAND\prenotesX	48, 225, 426
COMMAND\prepare@Xgroupbyline	199
COMMAND\prepare@Xprenotes	223
COMMAND\prev@nopb	356
COMMAND\prev@pb	356
COMMAND\prevlineX	102
COMMAND\prevpageX@num	103
COMMAND\print@Xfootnoterule	426
COMMAND\print@Xnotes	259, 260
COMMAND\print@Xnotes@forpages	425
COMMAND\print@eledsection	150
COMMAND\print@footnoteXrule	426
COMMAND\print@leftmargin@eledsection	344
COMMAND\print@lemma	190
COMMAND\print@line	148
COMMAND\print@notesX@forpages	425
COMMAND\print@rightmargin@eledsection	344
COMMAND\printendlines	231, 236, 276, 369, 417, 419

- COMMAND\printlinefootnote 191, 192, 425
- COMMAND\printlinefootnotearea 192, 193, 425
- COMMAND\printlinefootnotenumbers 191
- COMMAND\printlines 172, 189, 194, 195, 231, 276, 369, 417, 419, 425, 430
- COMMAND\printnpnum 364, 369
- COMMAND\printpstart 189
- COMMAND\protect 127, 363
- COMMAND\providecommand 201, 417
- COMMAND\pstart 2, 7, 17–21, 23, 67–69, 110, 123,
124, 128, 134, 141, 142, 144–146, 150, 160, 349, 363, 419–421, 423, 424, 426–428, 431–434
- COMMAND\pstartinfootnote 367
- COMMAND\pstartinfootnoteeverytime 367
- COMMAND\pstartnum 159
- COMMAND\pstartref 55, 56, 264, 271, 422, 434
- COMMAND\pstarts 420
- COMMAND\raggedX 47
- COMMAND\raggedleft 46
- COMMAND\raggedright 46
- COMMAND\raw@text 141
- COMMAND\rbracket 43, 44
- COMMAND\read@linelist 103, 105
- COMMAND\ref 57, 60, 63
- COMMAND\reformatted@ 276
- COMMAND\reformattedwithpage 276
- COMMAND\relax 19, 110, 132, 153, 160, 308, 325, 363
- COMMAND\renewcommand 71, 366, 369
- COMMAND\reset@msd@options@iffullpage 319
- COMMAND\resetprevline@ 102
- COMMAND\resetprevpage@ 103
- COMMAND\resumenumbering 20, 86, 90, 104, 105, 147, 419, 423, 424, 431
- COMMAND\right 66
- COMMAND\rightctab 337
- COMMAND\rightlinenum 22, 97, 417, 419
- COMMAND\rightltab 337
- COMMAND\rightnoteupfalse 61
- COMMAND\rightrtab 337
- COMMAND\rightrightnote 283
- COMMAND\rightrightskip 173, 176, 177, 179
- COMMAND\rightrightstartnum 159
- COMMAND\rigidbalance 182, 183, 185, 369, 430
- COMMAND\rigidbalanceX 182, 369, 430
- COMMAND\robustify 38
- COMMAND\roman 330, 430
- COMMAND\rtab 336–338, 341
- COMMAND\rtabtext 338, 342
- COMMAND\sameword 15, 29–32, 118, 134–137, 139, 425, 427, 429, 432, 434, 435
- COMMAND\sameword@inedtext 136
- COMMAND\saweword 135
- COMMAND\scriptsize 98
- COMMAND\section 67, 419, 433

COMMAND\section@num	86
COMMAND\sectionmark	343
COMMAND\select@lemmafont	50, 166
COMMAND\series	236, 237
COMMAND\series@	236
COMMAND\seriesatbegin	36, 248, 426
COMMAND\seriesatend	36, 249, 426
COMMAND\set@Xtxtbeforenotes	164
COMMAND\set@continuousnumberingforl	90
COMMAND\set@line	132
COMMAND\set@line@action	101, 111
COMMAND\set@txtbeforenotesX	164
COMMAND\setSErefonlypageprefixmore	58, 59, 275, 431, 434
COMMAND\setSErefonlypageprefixsingle	58, 59, 275, 431, 434
COMMAND\setSErefprefixmore	58
COMMAND\setSErefprefixsingle	58
COMMAND\setapprefprefixmore	58, 366
COMMAND\setapprefprefixsingle	58, 366, 430
COMMAND\setcommand@series	250
COMMAND\sethangingsymbol	52, 303, 366, 429
COMMAND\sethangingsymbol	51
COMMAND\setistwofollowinglines	197
COMMAND\setl@dlprbox	285
COMMAND\setline	23, 102, 106, 110, 123, 127, 144, 428
COMMAND\setlinenum	23, 106, 110, 124, 417
COMMAND\setmsdatalabel	34
COMMAND\setmsdataposition	434
COMMAND\setmsdataseries	34
COMMAND\setprintendlines	231, 233, 419
COMMAND\setprintlines	195, 197, 231, 419
COMMAND\setsidenotesep	61
COMMAND\setsidenotsep	366
COMMAND\setstanzaindent	305
COMMAND\setstanzaindents	52, 305, 363
COMMAND\setstanzapenalties	305
COMMAND\setstanzavalues	305
COMMAND\settoggle@series	250, 420, 424
COMMAND\showlemma	126, 417, 418
COMMAND\showwordrank	32, 136
COMMAND\sidenote@margin	418
COMMAND\sidenotemargin	60, 418, 423
COMMAND\sidenotesep	366
COMMAND\sidedstartnumtrue	19
COMMAND\skip	173
COMMAND\skipnumbering	23, 113, 114, 124, 419, 426, 427
COMMAND\skipnumbering@reg	426
COMMAND\small	45
COMMAND\special	13
COMMAND\splitmaxdepth	168, 184
COMMAND\splitoff	182

COMMAND\splittopskip	168, 184, 185
COMMAND\stanza	22, 23, 53, 54, 308, 366, 368, 429, 434
COMMAND\stanza@hang	307
COMMAND\stanza@line	307
COMMAND\stanzaindent	52, 305, 425
COMMAND\stanzaindent*	52
COMMAND\stanzaindentbase	304
COMMAND\stanzanumwrapper	54
COMMAND\startlock	22, 101, 124, 308
COMMAND\startstanzahook	366
COMMAND\startsub	22, 101, 122
COMMAND\stopmsd	313
COMMAND\stopmsdata	33, 311, 312
COMMAND\strip@pt	176
COMMAND\strutbox	184
COMMAND\sub@action	101, 111
COMMAND\sub@lock	100
COMMAND\sub@off	109, 267
COMMAND\sub@on	109, 267
COMMAND\subline@num	99, 101
COMMAND\sublinenum@rep	417
COMMAND\sublinenumberstyle	23, 97, 417
COMMAND\sublinenumincrement	21
COMMAND\sublinenumr@p	97, 417, 419
COMMAND\sublinenumrep	97, 419
COMMAND\sublineref	56, 264, 270
COMMAND\subsectionmark	343
COMMAND\sw@inthisedtext	129
COMMAND\sw@list@inedtext	133, 140
COMMAND\sw@txt	137
COMMAND\symlinenum	367
COMMAND\symplinenum	366
COMMAND\sza@penalty	307
COMMAND>tag	425
COMMAND\text	361
COMMAND\text⟨ <i>language</i> ⟩	45
COMMAND\textbeforenotesX	164
COMMAND\textcolor	72
COMMAND\textheight	72
COMMAND\the	417
COMMAND\thefootnoteA	34
COMMAND\thefootnoteX	421
COMMAND\thelabidx	299
COMMAND\thepage	106
COMMAND\thepstart	19
COMMAND\thepstartL	420
COMMAND\thepstartR	420
COMMAND\thestanza	54
COMMAND\this@line@list@version	118
COMMAND\thisfootnote	209

COMMAND\threecol@begin@insert	185
COMMAND\threecolfootfmt	184, 185, 431
COMMAND\threecolfootfmtX	214
COMMAND\threecolfootgroup	183
COMMAND\threecolfootgroupX	214
COMMAND\threecolfootsetup	183
COMMAND\threecolfootsetupX	213
COMMAND\threecolvfootnote	184
COMMAND\threecolvfootnote@inserted	184
COMMAND\threecolvfootnoteX	214
COMMAND\toendnotes	26, 229, 432
COMMAND\twocolfootfmt	431
COMMAND\twocolfootfmtX	212
COMMAND\twocolfootgroupX	212
COMMAND\twocolfootsetupX	211
COMMAND\twocolvfootnoteX	212
COMMAND\twolines	249, 367
COMMAND\twolines@A	249
COMMAND\twolines@B	250
COMMAND\twolines@C	250
COMMAND\twolinesbutnotmore	367
COMMAND\twolinesonlyinsamepage	367
COMMAND\txtbeforeXnotes	367
COMMAND\txtbeforenotesX	48, 435
COMMAND\txtbeforenotesonlyonceX	48, 435
COMMAND\uline	35
COMMAND\unhbox	177
COMMAND\unpenalty	178–180
COMMAND\unskip	179
COMMAND\unvxh	179, 368
COMMAND\unvxhX	368
COMMAND\upbracefill	336
COMMAND\usingcritext	362, 365
COMMAND\usingedtext	362, 365
COMMAND\vAfootnote	168
COMMAND\vadjust	121
COMMAND\variant	28
COMMAND\ vbox	144, 146, 177, 182, 222
COMMAND\vfootnote	168, 173, 177, 184
COMMAND\vl@dbfnote	202, 418
COMMAND\vl@disnote	284
COMMAND\vl@dlsnote	284
COMMAND\vl@dosnote	284
COMMAND\vl@drsnote	284
COMMAND\vnumfootnoteX	419
COMMAND\vsizer	49, 72
COMMAND\vsplit	162
COMMAND\waklam	336
COMMAND\waklamec	336
COMMAND\wapunktel	336

COMMAND\wastricht	336
COMMAND\widthX	49, 369, 431
COMMAND\wrap@edcrossref	269, 424
COMMAND\wrapcontentX	45, 432
COMMAND\wrapped@bodyfootmarkX	220
COMMAND\wrapped@footfootmarkX	219
COMMAND\x...	56
COMMAND\xabslineref	271
COMMAND\xdef	91, 308
COMMAND\xflagref	56, 271, 369, 430
COMMAND\xleft@appenditem	92, 126
COMMAND\xlineref	56, 369, 430
COMMAND\xpageref	56
COMMAND\xpstartref	56, 422
COMMAND\xr	60
COMMAND\xright@appenditem	91, 92
COMMAND\xsublineref	56
COMMAND\xxref	57, 272, 278, 422, 425, 432
COMMAND\zz@@@	417
ENVIRONMENTastanza	430
ENVIRONMENTedarrayc	341
ENVIRONMENTedarrayl	341
ENVIRONMENTedarrayr	341
ENVIRONMENTedtabularc	342
ENVIRONMENTedtabularl	342
ENVIRONMENTedtabularr	342
ENVIRONMENTledgroup	78, 290, 369, 430
ENVIRONMENTledgroupsize	291
PACKAGE(r)(e)ledmac	36
PACKAGEEledmac	12, 74, 101, 296, 364, 365, 425, 426
PACKAGEEledpar	426
PACKAGEEtoolbox	77
PACKAGEParallel	371
PACKAGEReledmac	368, 369
PACKAGEamsgen	320
PACKAGEamsmath	320
PACKAGEbabel	45, 73, 330, 430
PACKAGEbiblatex	70
PACKAGEbidi	45, 46, 78, 429
PACKAGEccaption	85
PACKAGEcolor	72
PACKAGEcsquotes	434
PACKAGEedmac	1, 6, 10, 12–14, 74, 194, 201, 264, 305, 361, 371, 417
PACKAGEedstanza	1, 13, 303
PACKAGEeledmac	1, 10, 11, 14–16, 63, 201, 292, 296, 323, 345, 358, 362, 364–366, 421, 423, 425
PACKAGEeledpar	86, 168, 343, 371, 419, 423–425
PACKAGEetex	429
PACKAGEetoolbox	91, 134, 236, 249, 258, 283, 344, 355
PACKAGEfancyhdr	261, 433
PACKAGEfloatrow	70, 369

PACKAGEfootmisc	35, 73, 78, 201, 371, 433
PACKAGEgeometry	15
PACKAGEglossaries	63, 64, 301, 430
PACKAGEhandout	424
PACKAGEhyperlink	244
PACKAGEhyperref	56, 129, 219, 220, 265, 298, 348, 357, 422–424, 431, 434
PACKAGEifluatex	77
PACKAGEifxetex	77
PACKAGEimakeidx	61, 62, 71, 78, 292, 296, 365, 421–423, 425
PACKAGEindextols	300
PACKAGEindextool	365
PACKAGEindextools	61, 62, 71, 78, 86, 292, 296, 300, 365, 425, 430, 433
PACKAGEinputenc	136
PACKAGEledarab	73
PACKAGEledmac	1, 10, 13, 14, 73, 91, 296, 361, 362, 365, 368
PACKAGEledpar	73
PACKAGEMemoir	77, 296, 365, 371, 424
PACKAGEMorewrites	71
PACKAGEMusixtex	423
PACKAGEperpage	430
PACKAGEpolyglossia	43, 73, 167, 189, 430
PACKAGERagged2e	46, 77
PACKAGEReledmac	1, 2, 11, 12, 14–17, 20, 21, 24, 25, 28–30, 33, 35, 38, 39, 42, 45, 46, 48–50, 52, 54, 56–58, 60–64, 68–75, 92, 95, 100, 102, 105, 106, 108, 118, 127, 130, 132, 135, 160, 169, 173, 177, 189, 201, 226, 237, 241, 242, 250, 258, 269, 271, 273, 278, 279, 296, 316, 323, 343, 344, 357, 358, 365, 366, 368–370, 428, 431, 433
PACKAGEReledpar	1, 4, 6, 8, 16, 19, 35, 49, 56, 59, 69, 70, 73, 75, 86, 92, 103, 108, 130, 132, 170, 173, 221, 222, 237, 243, 257, 259, 260, 292, 303, 429, 430, 432, 433, 435
PACKAGESuffix	77
PACKAGETabmac	1, 13, 371
PACKAGEulem	35
PACKAGEuninormalize	29
PACKAGExargs	29, 77
PACKAGExkeyval	75, 257
PACKAGEXr	5, 60, 279, 431
PACKAGEXref	278
PACKAGEXstring	77, 298

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

<code>\addtol@denbody</code>	1
Adelard II	13
<code>\advancelabel@refs</code>	1
<code>\advanceline</code>	1, 22
<code>\Aendnote</code>	25
<code>\affixline@num</code>	1
<code>\affixpstart@num</code>	1
<code>\affixside@note</code>	1
<code>\Afootnote</code>	25
<code>\afternoteX</code>	47
<code>\afterruleX</code>	48
<code>\ampersand</code>	1, 54
<code>\append@notesX</code>	1
<code>\append@Xnotes</code>	1
<code>\applabel</code>	1, 58
<code>\appref</code>	1, 58
<code>\apprefwithpage</code>	1, 58
<code>\arrangementX</code>	1, 37
<code>\arrangementX@normal</code>	1
<code>\arrangementX@threecol</code>	1
<code>\arrangementX@twocol</code>	1
<code>\at@every@pend</code>	1
<code>\AtEndEveryPend</code>	1, 19
<code>\AtEveryPend</code>	1, 19
<code>\AtEveryPstart</code>	1, 19
<code>\AtEveryStanza</code>	1, 53
<code>\AtEveryStopStanza</code>	1, 54
<code>\AtStartEveryPstart</code>	1, 19
<code>\AtStartEveryStanza</code>	1, 53, 54
<code>\autopar</code>	1, 18

B

<code>\ballast</code>	71
<code>\ballast@count</code>	1
Beeton, Barbara Ann Neuhaus Friend	18
<code>\beforeeledchapter</code>	1
<code>\BeforeEveryStopStanza</code>	1
<code>\beforeinsertingX</code>	46
<code>\beforeinsertion@X</code>	1
<code>\beforenotesX</code>	48
<code>\beginnumbering</code>	1, 16
<code>\Bendnote</code>	25
<code>\Bfootnote</code>	25
<code>\hookgroupX</code>	48
<code>\hooknoteX</code>	46
<code>\bodyfootmarkA</code>	34
<code>\boxfootnotenotenumbers</code>	1
Bredon, Simon	13
Breger, Herbert	13, 323
Brey, Gerhard	13

Busard, Hubert L. L.	13
\bypage@false	<u>1</u>
\bypage@true	<u>1</u>
\bypstart@false	<u>1</u>
\bypstart@true	<u>1</u>

C

\c@addcolcount	<u>1</u>
\c@ballast	<u>1</u>
\c@firstlinenum	<u>1</u>
\c@firstsublinenum	<u>1</u>
\c@labidx	<u>1</u>
\c@linenumincrement	<u>1</u>
\c@sublinenumincrement	<u>1</u>
\Cendnote	25
\Cfootnote	25
\ch@ck@l@ck	<u>1</u>
\ch@cksub@l@ck	<u>1</u>
\chapter	<u>1</u>
\check@pb@in@verse	<u>1</u>
Chester, Robert of	13
Claassens, Geert H. M.	13
\colalignX	46
Copernicus, Nicolaus	13
\critext	361
\ctab	<u>1</u>
\ctabtext	<u>1</u>

D

Dekker, Dirk-Jan	72
\Dendnote	25
\Dfootnote	25
\disable@familiarnotes	<u>1</u>
\disable@notes	<u>1</u>
\disable@sidenotes	<u>1</u>
\disablel@dtabfeet	<u>1</u>
\do@actions	<u>1</u>
\do@actions@fixedcode	<u>1</u>
\do@actions@next	<u>1</u>
\do@ballast	<u>1</u>
\do@feet@custom@order	<u>1</u>
\do@feetX	<u>1</u>
\do@insidelinehook	<u>1</u>
\do@line	<u>1</u>
\do@linehook	<u>1</u>
\do@lockoff	<u>1</u>
\do@lockoffL	<u>1</u>
\do@lockon	<u>1</u>
\do@lockonL	<u>1</u>
\do@Xfeet	<u>1</u>

<code>\doedindexlabel</code>	1
<code>\doendnotes</code>	1, 26
<code>\doendnotesbysection</code>	1, 26
<code>\doinsidelinehook</code>	1, 24
<code>\dolinehook</code>	1, 24
<code>\dosplits</code>	1
Downes, Michael	71, 177, 179
<code>\doextrafeet</code>	1
<code>\dummy@edindex</code>	1
<code>\dummy@edtext</code>	1
<code>\dummy@edtext@showlemma</code>	1
<code>\dummy@ref</code>	1

E

<code>\edaftertab</code>	1, 66, 336
<code>edarrayc</code> (environment)	64
<code>edarrayl</code> (environment)	64
<code>edarrayr</code> (environment)	64
<code>\edatleft</code>	1, 66
<code>\edatright</code>	1, 66
<code>\edbeforetab</code>	1, 66, 336
<code>\edfilldimen</code>	1
<code>\edfont@info</code>	1
<code>\edindex</code>	1, 61
<code>\edindexlab</code>	1, 63
<code>\EDLABEL</code>	1
<code>\edlabel</code>	1, 55
<code>\edlabelE</code>	1, 58
<code>\edlabelS</code>	1, 57
<code>\edlabelSE</code>	1, 58
<code>\edlineref</code>	1, 55
<code>\edmakelabel</code>	1, 57
<code>\edpageref</code>	1, 55
<code>\edrowfill</code>	1, 65
<code>\EDTAB</code>	1
<code>\edtabcolsep</code>	1, 65
<code>\EDTABINDENT</code>	1
<code>\edtabindent</code>	1
<code>\EDTABtext</code>	1
<code>edtabularc</code> (environment)	64
<code>edtabularl</code> (environment)	64
<code>edtabularr</code> (environment)	64
<code>\EDTEXT</code>	1
<code>\edtext</code>	1, 24
<code>\edvertdots</code>	1, 67
<code>\edvertline</code>	1, 67
<code>\Eendnote</code>	25
<code>\Efootnote</code>	25
<code>\eled@chapter</code>	1
<code>\eled@section</code>	1

<code>\eled@sectioning@out</code>	1
<code>\eled@subsection</code>	1
<code>\eled@subsubsection</code>	1
<code>\eledchapter</code>	1
<code>\eledchapter*</code>	1
<code>\eledsection</code>	1
<code>\eledsection*</code>	1
<code>\eledsubsection</code>	1
<code>\eledsubsection*</code>	1
<code>\eledsubsubsection</code>	1
<code>\eledsubsubsection*</code>	1
<code>\enablel@dtabfeet</code>	1
<code>\end@lemmas</code>	1
<code>\endashchar</code>	1
<code>\endline@num</code>	1
<code>\endlock</code>	1, 22
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 16
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 22
<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	64
<code>edarrayl</code>	64
<code>edarrayr</code>	64
<code>edtabularc</code>	64
<code>edtabularl</code>	64
<code>edtabularr</code>	64
<code>ledgroup</code>	55
<code>ledgroupsize</code>	55
<code>minipage</code>	55
<code>Euclid</code>	13
<code>\extensionchars</code>	1, 70

F

<code>\f@x@l@cks</code>	1
Fairbairns, Robin	35
<code>\first@linenum@out@false</code>	1
<code>\first@linenum@out@true</code>	1
<code>\firstlinenum</code>	1, 21
<code>\firstseriesX@</code>	1
<code>\firstsublinenum</code>	1, 21
<code>\firstXseries@</code>	1
<code>\fix@page</code>	1
<code>\flag@end</code>	1
<code>\flag@end@later</code>	1
<code>\flag@end@RTL</code>	1

<code>\flag@start</code>	1
<code>\flag@start@later</code>	1
<code>\flag@start@RTL</code>	1
<code>\flagstanza</code>	1, 54
<code>\flush@notes</code>	1
<code>\fnpos</code>	1, 36
Folkerts, Menso	13
<code>\footfootmarkA</code>	34
<code>\footfudgefiddle</code>	1, 71
<code>\footnote</code>	1
<code>\footnoteA</code>	34
<code>\footnoteB</code>	34
<code>\footnoteC</code>	34
<code>\footnoteD</code>	34
<code>\footnoteE</code>	34
<code>\footnotelang@lua</code>	1
<code>\footnotelang@poly</code>	1
<code>\footnoteoptions@</code>	1
<code>\footnoteXmark</code>	35
<code>\footnoteXtext</code>	35
<code>\footplitskips</code>	1
<code>\fullstop</code>	1

G

Gädeke, Nora	13
<code>\get@edindex@hyperref</code>	1
<code>\get@edindex@ledinnote@command</code>	1
<code>\get@fnmark</code>	1
<code>\get@fnmarkX</code>	1
<code>\get@index@command</code>	1
<code>\get@linelistfile</code>	1
<code>\get@sw@txt</code>	1
<code>\get@thisfootnote</code>	1
<code>\get@thisfootnoteX</code>	1
<code>\getline@num</code>	1
<code>\gl@p</code>	1

H

<code>\h@num</code>	1
<code>\hangindentX</code>	45
<code>\hidenumbering</code>	1, 23
<code>\hidenumberingonleftpage</code>	1, 23
<code>\hidenumberingonrightpage</code>	1
<code>\Hilfsbox</code>	1
<code>\hilfsbox</code>	1
<code>\hilfscount</code>	1
<code>\HILFSskip</code>	1
<code>\Hilfsskip</code>	1
<code>\hilfsskip</code>	1
<code>\hsizethreecolX</code>	47

\hsizetwocolX	47
\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
\ifat@every@pend@star@	1
\ifat@every@pstart@star@	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
\ifl@dhidenumber	1
\ifl@dmemoir	1
\ifl@dpaging	1
\ifl@dpairing	1
\ifl@dprintingcolumns	1
\ifl@dprintingpages	1
\ifl@dskipnumber	1
\ifl@dskipversenumber	1
\ifl@dstartendok	1
\ifl@footmisc	1
\ifl@imakeidx	1

<code>\ifl@indextools</code>	1
<code>\ifledfinal</code>	1, 70
<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@txtbeforenotesX</code>	1
<code>\insert@Xtxtbeforenotes</code>	1
<code>\insertthangingymbol</code>	1
<code>\insertlines@list</code>	1
<code>\insertparafootsepX</code>	1
<code>\inserts@list</code>	1

J

Jayaditya	13
-----------------	----

K

Kabelschacht, Alois	164
---------------------------	-----

L

<code>\l@advance@parledgroup@beforenormalnotes</code>	1
<code>\l@d@add</code>	1
<code>\l@d@nums</code>	1

<code>\l@d@section</code>	1
<code>\l@d@set</code>	1
<code>\l@d@Xend</code>	1
<code>\l@dampcount</code>	1
<code>\l@dbfnote</code>	1
<code>\l@dcheckcols</code>	1
<code>\l@dcheckstartend</code>	1
<code>\l@dchset@num</code>	1
<code>\l@dcolcount</code>	1
<code>\l@dcollect@@body</code>	1
<code>\l@dcollect@body</code>	1
<code>\l@dcolwidth</code>	1
<code>\l@dcsnote</code>	1
<code>\l@dcsnotetext</code>	1
<code>\l@dcsnotetext@l</code>	1
<code>\l@dcsnotetext@r</code>	1
<code>\l@ddodoreinextrafeet</code>	1
<code>\l@dedbeginmini</code>	1
<code>\l@dedendmini</code>	1
<code>\l@demptyd@ta</code>	1
<code>\l@dend@close</code>	1
<code>\l@dend@open</code>	1
<code>\l@dend@stuff</code>	1
<code>\l@dend@Xfalse</code>	1
<code>\l@dend@Xtrue</code>	1
<code>\l@denvbody</code>	1
<code>\l@desnote</code>	1
<code>\l@dfambeginmini</code>	1
<code>\l@dfamendmini</code>	1
<code>\l@dfeetbeginmini</code>	1
<code>\l@dfeetendmini</code>	1
<code>\l@dgetline@margin</code>	1
<code>\l@dgetlock@disp</code>	1
<code>\l@dgetref@num</code>	1
<code>\l@dgetsidenote@margin</code>	1
<code>\l@dobblearg</code>	1
<code>\l@disnote</code>	1
<code>\l@dlabel@parse</code>	1
<code>\l@dld@ta</code>	1
<code>\l@dlp@rbox</code>	1
<code>\l@dlsn@te</code>	1
<code>\l@dlsnote</code>	1
<code>\l@dmake@labels</code>	1
<code>\l@dmodfordtext</code>	1
<code>\l@dnullfills</code>	1
<code>\l@dnumstartsl</code>	1
<code>\l@doldold@footnotetext</code>	1
<code>\l@dp@rsefootspec</code>	1
<code>\l@dparsedendline</code>	1
<code>\l@dparsedendpage</code>	1

<code>\l@dparsedendsub</code>	1
<code>\l@dparsedstartline</code>	1
<code>\l@dparsedstartpage</code>	1
<code>\l@dparsedstartsub</code>	1
<code>\l@dparsefootspec</code>	1
<code>\l@dpush@begins</code>	1
<code>\l@drd@ta</code>	1
<code>\l@dref@undefined</code>	1
<code>\l@drestorefills</code>	1
<code>\l@drestoreforedtext</code>	1
<code>\l@drp@rbox</code>	1
<code>\l@drsn@te</code>	1
<code>\l@drsnote</code>	1
<code>\l@dsetmaxcolwidth</code>	1
<code>\l@dskipnumberfalse</code>	1
<code>\l@dskipnumbertrue</code>	1
<code>\l@dtabaddcols</code>	1
<code>\l@dtabnoexpands</code>	1
<code>\l@dunboxmpfoot</code>	1
<code>\l@dunhbox@line</code>	1
<code>\l@dzeropenalties</code>	1
<code>\label</code>	57
<code>\labelpstartfalse</code>	1
<code>\labelpstarttrue</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
<code>\led@err@ManyRightnotes</code>	1
<code>\led@err@ManySidenotes</code>	1
<code>\led@err@NumberingNotStarted</code>	1
<code>\led@err@NumberingShouldHaveStarted</code>	1
<code>\led@err@NumberingStarted</code>	1
<code>\led@err@NumberingWithoutPstart</code>	1
<code>\led@err@PendNoPstart</code>	1
<code>\led@err@PendNotNumbered</code>	1
<code>\led@err@PstartInPstart</code>	1
<code>\led@err@PstartNotNumbered</code>	1

<code>\led@err@ReverseColumns</code>	1
<code>\led@err@toendnotes@outsidenumbering</code>	1
<code>\led@err@TooManyColumns</code>	1
<code>\led@err@UnequalColumns</code>	1
<code>\led@error@fail@patch@@doclearpage</code>	1
<code>\led@error@fail@patch@@iiiminipage</code>	1
<code>\led@error@fail@patch@@makecol</code>	1
<code>\led@error@fail@patch@@reinserts</code>	1
<code>\led@error@fail@patch@endminipage</code>	1
<code>\led@error@PackageAfterEledmac</code>	1
<code>\led@mess@NotesChanged</code>	1
<code>\led@mess@SectionContinued</code>	1
<code>\led@nopb</code>	1
<code>\led@nopbnum</code>	1
<code>\led@pb</code>	1
<code>\led@pb@setting</code>	1
<code>\led@pbnum</code>	1
<code>\led@reinit@index@fornote</code>	1
<code>\led@set@index@fornote</code>	1
<code>\led@toksa</code>	1
<code>\led@toksb</code>	1
<code>\led@warn@AppLabelOutSecondArgEdtext</code>	1
<code>\led@warn@BadAction</code>	1
<code>\led@warn@BadAdvancelineLine</code>	1
<code>\led@warn@BadAdvancelineSubline</code>	1
<code>\led@warn@BadLineation</code>	1
<code>\led@warn@BadLinenummargin</code>	1
<code>\led@warn@BadLockdisp</code>	1
<code>\led@warn@BadSetline</code>	1
<code>\led@warn@BadSetlinenum</code>	1
<code>\led@warn@BadSidenotemargin</code>	1
<code>\led@warn@BadSublockdisp</code>	1
<code>\led@warn@DuplicateLabel</code>	1
<code>\led@warn@edinde@outsidenumbering</code>	1
<code>\led@warn@LineFileObsolete</code>	1
<code>\led@warn@NoFile</code>	1
<code>\led@warn@NoIndexFile</code>	1
<code>\led@warn@NoMarginpars</code>	1
<code>\led@warn@RefUndefined</code>	1
<code>\led@warn@SeriesStillExist</code>	1
<code>\led@warning@hsizeX@deprecated</code>	1
<code>\led@warning@msdatawithoutstop</code>	1
<code>\led@warning@preXnotes@deprecated</code>	1
<code>\led@warning@Xhsize@deprecated</code>	1
<code>ledgroup (environment)</code>	55
<code>ledgroupsize (environment)</code>	55
<code>\ledinnernote</code>	1, 60
<code>\ledinnote</code>	1
<code>\ledinnotehyperpage</code>	1
<code>\ledinnotemark</code>	1

<code>\ledleftnote</code>	1, 60
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 61
<code>\ledlsnotesep</code>	1, 61
<code>\ledlsnotewidth</code>	1, 61
<code>\lednopb</code>	1, 69
<code>\lednopbinversetrue</code>	69
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	1, 60
<code>\ledpb</code>	1, 69
<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 69
<code>\ledrightnote</code>	1, 60
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 61
<code>\ledrsnotesep</code>	1, 61
<code>\ledrsnotewidth</code>	1, 61
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 60
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 22
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	61
<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
<code>Leibniz</code>	13
<code>\lemma</code>	1, 27
<code>\letsforverteilen</code>	1
<code>\line@list</code>	1
<code>\line@list@stuff</code>	1
<code>\line@list@version</code>	1
<code>\line@margin</code>	1
<code>\line@num</code>	1
<code>\line@set</code>	1
<code>\lineation</code>	1, 21
<code>\linenum</code>	1, 27
<code>\linenum@out</code>	1
<code>\linenumberlist</code>	1, 21
<code>\linenumberstyle</code>	1, 23
<code>\linenumincrement</code>	1, 21
<code>\linenummargin</code>	1, 21
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 22
<code>\linerrangesep@</code>	1
<code>\list@clear</code>	1

<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 22
Lorch, Richard	13
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	76

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>	49
Mayer, Gyula	13
<code>\measurembody</code>	1
<code>\measuremcell</code>	1
<code>\measuremrow</code>	1
<code>\measuretbody</code>	1
<code>\measuretcell</code>	1
<code>\measuretrow</code>	1
Middleton, Thomas	13, 99
<code>minipage</code> (environment)	55
Mittelbach, Frank	12, 13
<code>\morenoexpands</code>	1, 72
<code>\mp@append@notesX</code>	1
<code>\mp@append@Xnotes</code>	1
<code>\mpfnpos</code>	1, 36
<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, 33

\msdata@c	1
\msdata@cR	1
\multfootsep	1, 35
\multiplefootnotemarker	1

N

\n@num	1
\n@num@stanza	1
\new@line	1
\newhookarg@specific	1
\newhookcommand@series	1
\newhookcommand@series@reload	1
\newhooktoggle@series	1
\newhooktoggle@series@reload	1
\newhooktoggle@specific	1
\newseries@	1
\newverse	1
\NEXT	1
\no@expands	1
\noeledsec	68
\nomk@	1
\normal@footnotemarkX	1
\normal@page@break	1
\normal@pars	1
\normalbfnoteX	1
\normalbodyfootmarkX	1
\normalfootfmt	1
\normalfootfmtX	1
\normalfootfootmarkX	1
\normalfootgroup	1
\normalfootgroupX	1
\normalfootnoterule	1
\normalfootnoteruleX	1
\normalfootstart	1
\normalfootstartX	1
\normalvfootnote	1
\normalvfootnote@inserted	1
\normalvfootnoteX	1
\notefontsizeX	45
\notenumfontX	44
\noteschanged@false	1
\noteschanged@true	1
\noteswidthliketwocolumnsX	49
\nulledindex	1
\nullsetzen	1
\num@lines	1
\numberedpar@false	1
\numberedpar@true	1
\numberingfalse	1
\numberingtrue	1

<code>\numberlinefalse</code>	20
<code>\numberlinetrue</code>	20
<code>\numberpstartfalse</code>	1, 19
<code>\numberpstarttrue</code>	1, 19
<code>\numberstanzafalse</code>	54
<code>\numberstanzatrue</code>	54
<code>\numlabfont</code>	1, 50

O

<code>\old@hsize</code>	1
<code>\one@line</code>	1
<code>optionauxdir</code>	15, 432
<code>optioncontinuousnumberingwithcolumns</code>	431
<code>optioninnnote</code>	430
<code>optioninnote</code>	430
<code>optionlinrangesep</code>	257
<code>optionnocritical</code>	430
<code>optionnoeledsec</code>	349, 432
<code>optionnoend</code>	430
<code>optionnopenalties</code>	71
<code>optionnotenumber</code>	430
<code>optionswcaseinsensitive</code>	29, 435

P

<code>\page@action</code>	1
<code>\page@num</code>	1
<code>\pagelinesep</code>	1, 62
<code>\pageref</code>	57
<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>	47
<code>\parafootstart</code>	1
<code>\parafootstartX</code>	1
<code>\paravfootnote</code>	1
<code>\parindentX</code>	45
<code>\pausenumbering</code>	1, 20
<code>\pend</code>	1, 17
Plato of Tivoli	13
<code>\postbodyfootmark</code>	1
<code>\prebodyfootmark</code>	1
<code>\prenotesX</code>	48
<code>\prepare@edindex@fornote</code>	1
<code>\prepare@prenotesX</code>	1
<code>\prepare@Xgroupbyline</code>	1

<code>\prepare@Xprenotes</code>	1
<code>\prev@nopb</code>	1
<code>\prev@pb</code>	1
<code>\prevpage@num</code>	1
<code>\print@eledsection</code>	1
<code>\print@footnoteXrule</code>	1
<code>\print@leftmargin@eledsection</code>	1
<code>\print@lemma</code>	1
<code>\print@line</code>	1
<code>\print@notesX</code>	1
<code>\print@rightmargin@eledsection</code>	1
<code>\print@Xfootnoterule</code>	1
<code>\print@Xnotes</code>	1
<code>\printendlines</code>	1
<code>\printlineendnote</code>	1
<code>\printlineendnotearea</code>	1
<code>\printlinefootnote</code>	1
<code>\printlinefootnotearea</code>	1
<code>\printlinefootnotenumbers</code>	1
<code>\printlines</code>	1
<code>\printnpnum</code>	1
<code>\printpstart</code>	1
<code>\printsymlineendnotearea</code>	1
<code>\printsymlinefootnotearea</code>	1
<code>\printXafternumber</code>	1
<code>\printXbeforenumber</code>	1
<code>\pstart</code>	1, 17
<code>\pstarteref</code>	1
<code>\pstartnum</code>	1
<code>\pstartref</code>	55

Q

<code>\quotation</code>	1
<code>\quote</code>	1

R

<code>\raggedX</code>	47
<code>\raw@text</code>	1
<code>\rbracket</code>	1
<code>\read@linelist</code>	1
<code>\ref</code>	57
<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, 102
<code>\resetprevpage@</code>	1
<code>\resetprevpage@num</code>	103

<code>\restore@familiarnotes</code>	1
<code>\restore@notes</code>	1
<code>\restore@sidenotes</code>	1
<code>\resumenumbering</code>	1, 20
<code>\rightctab</code>	1
<code>\rightlinenum</code>	1, 22
<code>\rightltab</code>	1
<code>\rightnoteupfalse</code>	61
<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	13
<code>\section@num</code>	1
<code>\select@lemmafонт</code>	1
<code>\select@lemmafонт</code>	1, 50
<code>\SEref</code>	1, 58
<code>\SErefonlypage</code>	58
<code>\SErefwithpage</code>	1, 58
<code>\series</code>	1
<code>\seriesatbegin</code>	1, 36
<code>\seriesatend</code>	1, 36
<code>\set@continuousnumberingforL</code>	1
<code>\set@line</code>	1
<code>\set@line@action</code>	1
<code>\set@txtbeforenotesX</code>	1
<code>\set@Xtxtbeforenotes</code>	1
<code>\setapprefprefixmore</code>	58
<code>\setapprefprefixsingle</code>	58
<code>\setcommand@series</code>	1
<code>\sethangingsymbol</code>	1, 52
<code>\setistwofollowinglines</code>	1
<code>\setl@dlp@rbox</code>	1
<code>\setl@drpr@box</code>	1
<code>\setline</code>	1, 22
<code>\setlinenum</code>	1, 23
<code>\setmcellcenter</code>	1
<code>\setmcellleft</code>	1
<code>\setmcellright</code>	1
<code>\setmrowcenter</code>	1
<code>\setmrowleft</code>	1
<code>\setmrowright</code>	1
<code>\setmsdatalabel</code>	1, 34

\setmsdataposition	1, 34
\setmsdataseries	1, 34
\setnoteswidthliketwocolumnsX@	1
\setnotesXpositionliketwocolumns@	1
\setprintendlines	1
\setprintlines	1
\setSErefonlypageprefixmore	58
\setSErefonlypageprefixsingle	58
\setSErefprefixmore	58
\setSErefprefixsingle	58
\setsidenotesep	61
\setstanzaindents	1, 51
\setstanzapenalties	1, 52
\setstanzavalues	1
\settccllcenter	1
\settccllleft	1
\settccllright	1
\settoggle@series	1
\setthrowcenter	1
\setthrowleft	1
\setthrowright	1
\setXnotespositionliketwocolumns@	1
\setXnoteswidthliketwocolumns@	1
\showlemma	1, 70
\showwordrank	1, 32
\sidenote@margin	1
\sidenotemargin	1, 60
\sidepstartnumtrue	19
\skip@lockoff	1
\skipnumbering	1, 23
\splitoff	1
\spreadmath	1, 65
\spreadtext	1, 65
\stanza	1, 50
\stanza@count	1
\stanza@hang	1
\stanza@line	1
\stanzaindent	1, 52
\stanzaindent*	1, 52
\stanzaindentbase	1, 51
\stanzanumwrapper	1, 54
\startlock	1, 22
\startsub	1, 22
\stepl@dcollcount	1
\stopmsdata	1, 33
\strip@szacnt	1
\sub@action	1
\sub@lock	1
\sub@off	1
\sub@on	1

\subline@num	1
\sublinenumberstyle	1, 23
\sublinenumincrement	1, 21
\sublinenumr@p	1
\sublinenumrep	1
\sublineref	1, 55
\sublines@false	1
\sublines@true	1
\subblock@disp	1
\subblockdisp	1
Sullivan, Wayne	13, 71, 85, 90, 177, 178, 264, 303
\sza@penalty	1

T

\tabHilfbox	1
\tabhilfbox	1
\theadcolcount	1
\theadtext	1
\theendpageline	1
\thefootnoteA	34
Theodosius	13
\thepageline	1
\thepstart	1, 19
\thestanza	1, 54
\thestartpageline	1
\this@line@list@version	1
\threecol@begin@insert	1
\threecolfootfmt	1
\threecolfootfmtX	1
\threecolfootgroup	1
\threecolfootgroupX	1
\threecolfootsetup	1
\threecolfootsetupX	1
\threecolvfootnote	1
\threecolvfootnote@inserted	1
\threecolvfootnoteX	1
\toendnotes	1, 26
\toendnotes*	1
\twocolfootfmt	1
\twocolfootfmtX	1
\twocolfootgroup	1
\twocolfootgroupX	1
\twocolfootsetup	1
\twocolfootsetupX	1
\twocolvfootnote	1
\twocolvfootnote@inserted	1
\twocolvfootnoteX	1
\txtbeforenotesonlyonceX	48
\txtbeforenotesX	48

U

\unvxxX 1

V

Vamana 13
 \variab 1
 \vbfnoteX 1
 \vl@dbfnote 1
 \vl@dcsnote 1
 \vl@disnote 1
 \vl@dlsnote 1
 \vl@dosnote 1
 \vl@drsnote 1
 \vnumfootnoteX 1

W

Whitney, Ron 12
 \widthX 49
 \wrap@edcrossref 1
 \wrapcontentX 45
 \wrapped@bodyfootmarkX 1
 \wrapped@footfootmarkX 1
 Wujastyk, Dominik 12

X

\X@atbegininsertion 1
 \X@beforeinsertion 1
 \X@doreinfeet 1
 \xabslineref 1
 \Xafterlemmaseparator 43
 \Xafternote 47
 \Xafternumber 41
 \Xafterrule 48
 \Xaftersymmlinenumber 41
 \Xarrangement 1, 37
 \Xarrangement@normal 1
 \Xarrangement@paragraph 1
 \Xarrangement@threecol 1
 \Xarrangement@twocol 1
 \Xbeforeinserting 46
 \Xbeforelemmaseparator 43
 \Xbeforenotes 48
 \Xbeforenumber 39, 41
 \Xbeforemsymmlinenumber 41
 \Xbhookgroup 48
 \Xbhooknote 46
 \Xboxlinenum 42
 \Xboxlinenumalign 42
 \Xboxsymmlinenumber 42
 \Xcolalign 46

<code>\xedindex</code>	1
<code>\xedlabel</code>	1
<code>\xedtext</code>	1
<code>\Xendafterenumber</code>	41
<code>\Xendafterlemmaseparator</code>	44
<code>\Xendafternote</code>	50
<code>\Xendafterpagenumber</code>	43
<code>\Xendaftersymlinenum</code>	41
<code>\Xendahookinplaceofnumber</code>	43
<code>\Xendahooklinenumber</code>	43
<code>\Xendbeforelemmaseparator</code>	44
<code>\Xendbeforenumber</code>	41
<code>\Xendbeforepagenumber</code>	43
<code>\Xendbeforesymlinenum</code>	41
<code>\Xendbhookinplaceofnumber</code>	43
<code>\Xendbhooklinenumber</code>	43
<code>\Xendbhooknote</code>	46
<code>\Xendboxendlinenumalign</code>	42
<code>\Xendboxlinenum</code>	42
<code>\Xendboxlinenumalign</code>	42
<code>\Xendboxstartlinenumalign</code>	42
<code>\Xendboxsymlinenum</code>	42
<code>\Xendhangindent</code>	45
<code>\Xendinplaceofflemmaseparator</code>	44
<code>\Xendinplaceofnumber</code>	41
<code>\Xendinplaceofpagenumber</code>	39
<code>\Xendlemmadisablefontselection</code>	44
<code>\Xendlemmafont</code>	44
<code>\Xendlemmaseparator</code>	44
<code>\Xendlineprefixmore</code>	43
<code>\Xendlineprefixsingle</code>	43
<code>\Xendlinerrangeseparator</code>	39
<code>\Xendmorethantwolines</code>	40
<code>\Xendnonumber</code>	40
<code>\Xendnotefontsize</code>	45
<code>\Xendnotenumfont</code>	44
<code>\Xendnumberonlyfirstinline</code>	38
<code>\Xendnumberonlyfirstintwolines</code>	38
<code>\Xendpagenumberonlyfirst</code>	38
<code>\Xendparagraph</code>	50
<code>\Xendsep</code>	50
<code>\Xendsublinesep</code>	41
<code>\Xendsymlinenum</code>	38
<code>\Xendsympagenum</code>	39
<code>\Xendtwolines</code>	40
<code>\Xendtwolinesbutnotmore</code>	40
<code>\Xendwrapcontent</code>	45
<code>\xflagref</code>	1
<code>\Xgroupbyline</code>	47
<code>\Xgroupbylineseparetwolines</code>	47

\Xhangindent	45
\Xhsizethreecol	47
\Xhsizetwocol	47
\Xinplaceoflemmaseparator	43
\Xinplaceofnumber	41
\Xinsertparafootsep	1
\Xledsetnormalparstuff	1
\xleft@appenditem	1
\Xlemmadisablefontselection	44
\Xlemmafont	44
\Xlemmaseparator	43
\Xlinrangeseparator	39
\xlineref	1, 56
\Xmaxhnotes	49
\Xmorethantwolines	39
\Xnolemmaseparator	1, 43
\Xnonbreakableafternumber	41
\Xnonumber	40
\Xnotefontsize	45
\Xnotenumfont	44
\Xnoteswidthliketwocolumns	49
\Xnumberonlyfirstinline	38
\Xnumberonlyfirstintwolines	38
\Xonlypstart	40
\Xpagelinesep	41
\xpageref	1, 56
\Xparafootsep	47
\Xparindent	45
\Xprenotes	1, 48
\Xprenotes@	1
\Xpstart	40
\Xpstarteverytime	40
\xpstartref	1, 56
\XR@test	1
\XR@test@mac	1
\XR@test@mac@test	1
\Xragged	47
\xright@appenditem	1
\Xrigidbalance	1
\Xstanza	40
\Xstanzaseparator	40
\Xstorelineinfo	1
\xsublineref	1, 56
\Xsublinesep	41
\Xsymlinenum	38
\Xtoendnotes	26
\Xtwolines	39
\Xtwolinesonlyinsamepage	40
\Xtxtbeforenotes	48
\Xtxtbeforenotesonlyonce	48

<code>\Xunvxh</code>	<u>1</u>
<code>\Xwidth</code>	49
<code>\Xwrapcontent</code>	45
<code>\Xwrapendlemma</code>	45
<code>\Xwraplemma</code>	45
<code>\xxref</code>	<u>1</u> , 57

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	77
\morenoexpands: Added \l@dtabnoexpands to \no@expands	127
\reledmac@error: Added \eledmac@error and replaced error messages	79
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	267
\doxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	258
\edlabel: Tweaked \edlabel to get correct page numbers	264
\l@ddodoreinextrafeet: Renamed \dodoreinextrafeet to \l@ddodoreinextrafeet	259
\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	127
\zz@@@: Minor change to \zz@@@	264
General: Added text about normal labeling	57
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	261
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	175
\line@list@stuff: Added initial write of page number in \line@list@stuff	119
\para@footsetup: Added \footfudgefiddle to \para@footsetup	176
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	216
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	85
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	267
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	106
\ledlinenum: Added \linenumr@p and \sublinenum@rep to \leftlinenum and \rightlinenum	98
\linenumberlist: Added \linenumberlist mechanism	85
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	233
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	198
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	97
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	289
\Xarrangement@normal: Added minpage footnote setup to \footnormal	170
\edtext: Added \showlemma to \edtext	128
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	287

\mpnormalfootgroup: Added \mpnormalfootgroup	174
\mpnormalvfootnote: Added \mpnormalvfootnote	172
\showlemma: Added \showlemma	85
General: Added final/draft options	75
Added ledgroup environment	290
Added ledgroupsize environment	291
Added minipage, etc., support	1
v0.4.1.	
\do@Xfeet: Changed \do@Xfeet code for easier extensions	259
\edindex: Let eledmac take advantage of memoir's indexing	296
\print@Xnotes: Added \opXfeet	259
General: Added code for changing \doclearpage	262
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	201
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	202
\@xympar: Eliminated \marginpar disturbance	279
General: Added left and right side notes	280
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	155
\l@edgetsidenote@margin: Added \sidenotemargin and \sidenote@margin	280
General: Added moveable side note	280
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	291
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	108
\@nl@reg: Added \fix@page to \@nl	106
Extended \@nl to include the page number	106
\fix@page: Added \last@page@num and \fix@page	108
\get@thisfootnote: Changed \l@dbfnote and \vl@dbfnote as originals could give incorrect markers in the footnotes	202
\new@line: Extended \new@line to output page numbers	120
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	106
\@ref@reg: Added \@ref@reg	116
\affixline@num: Added skipnumering to \affixline@num	155
\do@actions@fixedcode: Added \do@actions@fixedcode	154
\do@actions@next: Added number skipping to \do@actions	153
\do@insidelinehook: Added \do@linehook for use in \do@line	150
\endnumbering: Changed \endnumbering for eledpar	89
\fix@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fix@l@cks	157
\footplitskips: Added \footplitskips for use in many footnote styles	168
\get@linelistfile: Added \get@linelistfile	105
\initnumbering@reg: Added \initnumbering@reg	87

\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	290
\l@dcsnotetext@r: Added \l@emptyd@ta	151
\l@dgetline@margin: Added \l@dgetline@margin	94
\l@dgetlock@disp: Added \l@dgetlock@disp	96
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	280
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	86
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	151
\l@dzeropenalties: Added \l@dzeropenalties	146
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	98
\line@list@stuff: Deleted \page@start from \line@list@stuff	119
\list@clearing@reg: Added \list@clearing@reg	105
\n@num: Added \n@num	113
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	208
\resumenumbering: Changed \resumenumbering for eledpar	90
\setprintendlines: Added \setprintendlines for use by \printendlines	231
\setprintlines: Added \setprintlines for use by \printlines	195
\skipnumbering: Added \skipnumbering and supports	124
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	95
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	97
Using \sublinenumrep instead of \sublinenumr@p	97
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	210
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	79
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	86
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	19

Possibility to number the pstart with the commands <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.	305
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every n verses. . .	51
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.	264
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.	155
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	303
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code>	198
General: Line numbering can be reset at each <code>pstart</code>	92
Possibility to print <code>\pstart</code> number inside.	19
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	303
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features	127
General: <code>\lemma</code> can contain commands.	27
Debug in lineation command	21
New generic commands to customize footnote display.	37, 249
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code>	25
Options of <code>\Xfootnotes</code>	166
Possibility to have commands in sidenotes.	60
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. . .	38
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code>	224
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value.	250
General: Add <code>\labelpstarttrue</code>	19
Add <code>\Xnumberonlyfirstintwolines</code>	38
Add <code>\Xpstart</code> and <code>\Xonlypstart</code>	40
New hook to add arbitrary code at the beginning of the notes	46
New options for block of notes.	48
New package option: <code>parapparatus</code>	1
New tools to change order of series	248
Sectioning commands.	67

v1.2.0.	
<code>\Xprenotes</code> : Debug in familiar footnotes (bug introduced by v1.1).	224
<code>\endquote</code> : Compatibility of <code>\ledchapter</code> with the <i>memoir</i> class.	342
v1.3.0.	
<code>\endquote</code> : <i>Quotation</i> and quote environment inside numbered sections.	342
v1.4.0.	
<code>\edtext</code> : Compatibility of <code>\edtext</code> with the right-to-left direction (with Polyglossia).	128
<code>\ledsetnormalparstuffX</code> : Direction of footnotes with polyglossia.	220
<code>\newseries@</code> : Remembers the language of the lemma, in order to create a correct direction for the footnote separator.	239
<code>\rbracket</code> : Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX).	189
General: Compatibility with LuaTeX of RTL notes.	1
v1.4.1.	
<code>\affixside@note</code> : Remove spurious spaces.	286
<code>\endquote</code> : New option <i>noquotation</i> .	342
<code>\get@thisfootnote</code> : Compatibility of standard footnotes with <code>eledmac</code> when these footnotes contain any commands.	202
<code>\labelrefsparsesubline</code> : Fix bug with <code>\edlabel</code> .	266
v1.4.2.	
General: Debug with some special classes.	1
v1.4.3.	
General: Add <code>\Xnonbreakableafternumber</code> .	41
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> .	342
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.	208
v1.5.0.	
<code>\do@insidelinehook</code> : Added <code>\do@insidelinehook</code> for use in <code>\do@line</code>	150
<code>\edindex</code> : Compatibility with <code>imakeidx</code> package, and possibility to use multiple index with <code>\edindex</code> .	296
General: Correct indexing when the call is made in critical notes.	292
v1.5.1.	
<code>\managestanza@modulo</code> : Correct <code>stanzaindentsrepetition</code> counter	305
<code>\normalvfootnoteX</code> : Fix bug with normal familiar footnotes when mixing RTL and LTR text.	205
v1.6.0.	
<code>\newverse</code> : Add <code>\falseverse</code> macro.	308
v1.6.1.	
<code>\AtStartEveryPstart</code> : Spurious space in <code>\pstart</code> .	141

\ifinserthangingsymbol: Hang verse is now not automatically flush right.	303
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill inside.	147
\pend: Spurious space in \pend.	143
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.	69
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts.	342
\get@index@command: Debug \get@index@command and compatibility with hyperref package.	295
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work.	252
\prevpage@num: Correct \parafootsep when using with ledgroup.	181
General: Compatibility with parledgroup option ofeledpar package.	1
If imakeidx and hyperref are loaded, adds hyperref in the index.	292
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined).	225
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.	292
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes.	258
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup.	287
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts.	342
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.	44
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.	106
\AtStartEveryPstart: New optional argument for \pstart, to execute code before it.	141
\edindex: Use correctly default index when imakeidx is loaded.	296
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands.	342
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar.	87
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	280
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in eledmac, including needs for eledpar case.	281
\l@dnumpstartsL: Add \ifledRcol@ for eledpar	86
\l@dunhbox@line: \do@line is split in more little commands.	148
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX.	252
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph.	252
\pend: New optional argument for \pend, to execute code after it.	143
\stanza: &can have an optional argument: content to be printed after.	308
\Stanza can have an optional argument: content to be printed before.	308
Add \newverse macro, \falseverse deprecated.	308
General: Add \ledinnernote and \ledouternote commands.	60
Add \Xendparagraph and related settings.	50
Add hyperlink to crossref (needs hyperref package).	55
Compatibility with musixtex.	1
Debug eledmac sectioning command after using \resumenumbering.	1
Ensure that imakeidx is loaded <i>before</i> eledmac	292
New hooks: \Xafterrule and \afterruleX	48
New options for ragged-paragraph notes	47
New sectioning commands.	67
Optional arguments for \pstart and \pend.	18

v1.12.1.

\wrap@edcrossref: Fix spurious spaces.	269
--	-----

v1.12.2.

\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	147
--	-----

v1.12.3.

\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar	120
\flag@start send a error message when a \edtext is done without insert (note)	120
\reledmac@error: Replaced error messages	79
General: Add macros for new messages since v0.7	79
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.	60
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	298
\@ssect: Debug \eledchapter in parallel with memoir	345
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	151
\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	89
\l@dgobblearg: \l@dgobblearg becomes \l@gobbeloptarg	325
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	326
\resumenumbering: Debug \resumenumbering	90
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	269
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	120
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	252
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	216
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	250
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	49
Added widthliketwocolumns option	75
v1.13.1.	
\ifat@every@pstart@star@: Add \l@dzeropenalties in \pstart	142
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	86
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	296
General: Debug spurious characters before endnotes.	225
Delete previous override of \l@d@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	300
Move gobbling command	85
Provide \@gobblefour	85
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref	
package.	348
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	345
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	128
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	215
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	122
\if@lemmacommand@: New boolean \iflemmacommand@.	133

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> .	19
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.	119
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.	128
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.	270
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph.	128
v1.18.0.	
<code>\@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools + hyperref</code> .	298
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code>	264
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes.	226
<code>\l@dnumpstartsL</code> : Add <code>\ifl@dprintingpages</code> and <code>\@dprintingcolumns</code> for <code>eledpar</code>	86
<code>\print@line</code> : Compatibility with <code>Lua\TeX</code> RTL languages.	148
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code>	191
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns (<code>eledpar</code>)	189
General: Add <code>\Xpstarteverytime</code>	40
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.	40
v1.19.0.	
<code>\footssplitskips</code> : <code>\footssplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages.	168
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty.	272
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>	198
<code>\stanza</code> : <code>&</code> and <code>&</code> can be preceded by spaces.	308
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (fix bug added in 1.19.0).	272
General: Add <code>\Xendboxlinenum</code>	42
Add <code>\Xtwolines</code> and <code>\Xmorethantwolines</code> hooks	39
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>	25
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	298
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments.	226
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code>	295
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code>	220
<code>\n@num</code> : <code>\n@num@ref</code> deleted	113
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	113
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument.	251
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument.	252
<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>	223
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code>	249
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code>	124
Correct <code>\skipnumbering</code> for stanza.	124
Delete <code>\skipnumbering@reg</code>	124
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	248
Add <code>\applabel</code> and related	58
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column.	1
Add <code>\hidenumbering</code>	23
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks	46
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code>	40
Add <code>\Xparindent</code> and <code>\hangindentX</code>	45
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>	85
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks.	295
General: Add <code>\doendnotesbysection</code> command.	26
Add option for lemma separator inside endnotes	44
Adds hyperlink for references to notes in indices.	1
Fix conflict between <code>noend</code> package option and <code>edtabularx</code> environments	1
Provides support for <code>xindy</code>	1
Standardize endnotes handbook.	26
When using <code>hyperref</code> package, internal links in index or with <code>\edlineref</code> are now targeted to the top and not longer to the bottom of the lines they refer to.	1
v1.22.1.	
<code>\prevpage@num</code> : Correct double symbol when using both <code>\parafootsep</code> and <code>\Xsymlinenum</code>	181
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>	128
<code>\SErefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code>	276
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command	133
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0).	239
<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.	137
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code>	42
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code>	42
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.	289
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it	279
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it.	289
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code>	342
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some conde of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code>	290
<code>\newseries@</code> : One endnotes file by series.	245
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined	261
<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>	261
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	202
Suppress indexing command specific to <code>memoir</code>	296
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.	220
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.	166
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>	39

Option <code>linrangesep</code> for critical footnotes and endnotes.	39
v2.4.1.	
General: Fix bug with <code>\appref</code> and <code>\apprefwithpage</code> (introduced in v2.4.0).	1
Fix bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override \LaTeX <code>\roman</code> command, like Greek language.	1
Fix bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override \LaTeX <code>\roman</code> command, like Greek.	1
v2.5.0.	
<code>\SErefwithpage</code> : Debug <code>\setapprefprefixsingle</code>	275
<code>\edlabel</code> : Fix bug when calling <code>\edlabel</code> in a footnotes of the rightside	264
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments.	226
<code>\printlines</code> : <code>\printlines</code> takes an eighth argument: the line flag	198
<code>\xlineref</code> : <code>\xlineref</code> does not include anymore the side flag. Use <code>\xflagref</code> to get it. Not that <code>\edlineref</code> still contains the flag.	270
General: <code>\apprefwithpage</code> and <code>\appref</code> print double quotation mark when the label was not defined.	1
<code>\apprefwithpage</code> and <code>\appref</code> work with right side crossref.	1
<code>\apprefwithpage</code> works also when <code>noend</code> option is enabled.	1
<code>\appref</code> and <code>\apprefwithpage</code> can take <code>linrangesep</code> optional argument.	1
<code>\edlabel</code> works now in <code>\Xfootnote</code>	1
<code>\lemma</code> can be used even when the <code>nocritical</code> is enabled.	1
Compatibility with new hook and tools of <code>reledpar</code> 2.6.0.	1
Fix spurious vertical space in <code>astanza</code> environment (<code>reledpar</code>)	1
Log now states ‘There were undefined references’ when using wrong references in <code>\edlineref</code> or <code>edpageref</code>	1
New hooks to customize page and line number appearance in endnotes.	1
New hooks: <code>\Xhookgroup</code> and <code>\bhookgroupX</code>	1
New tools to easily make cross-reference to a passage defined by a start and an end line	57
v2.6.0.	
General: Adds compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code> package.	1
Fix bug with footnote counter in <code>ledgroup</code> (added in v2.5.0).	1
Fix bug, introduced in v2.5.0, with footnote numbering in parallel typesetting when using <code>perpage</code> package.	1
v2.7.0.	
<code>\@k</code> : <code>\rigidbalance</code> is split in <code>\Xrigidbalance</code> and <code>\rigidbalanceX</code>	182
<code>\l@d@section</code> : <code>\endnotes</code> take seven arguments.	226
General: Add dash as default page range separator for <code>\SEonlypage</code>	1
Debug <code>\SErefonlypage</code> when referring to only one page.	1
Delete parenthesis after <code>\SErefonlypage</code>	1
Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.5.0).	1
Fix bug with <code>\SErefwithpage</code>	1
Fix bugs in compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code> package, when indexing outside of a <code>ledgroup</code>	1
New commands to make glossaries connected to page and linenumber with the <code>glossaries</code> package	1
New hooks: <code>\Xhsize</code> and <code>\hsizeX</code>	49
New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</code>	44

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>	49
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.	226
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> .	190

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> .	45
New hooks: <code>\Xwraplemma</code> and <code>\Xendwraplemma</code> .	45
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	154
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>	261
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks.	1
Add <code>\Xpagelinesep</code> hook.	41
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.	261
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 the <code>hyperref</code> package. . . .	1
v2.19.0.	
General: Add <code>\footnoteXmark</code> and <code>\footnoteXtext</code> commands.	1
Add better compatibility with the <code>csquotes</code> package when using familiar footnotes. .	1
Fix bug with paragraph indent after sectioning command.	1
v2.20.0.	
General: Add <code>\AtStartEveryStanza</code> , <code>\BeforeEveryStopStanza</code> ,	
<code>\AtEndEveryPend</code> , <code>\AtStartEveryPstart</code>	1
Add second optional argument of <code>\pstart</code> , <code>\pend</code> and <code>\stanza</code>	1
Add starred version of <code>\AtEveryPstart</code> , <code>\AtEveryPend</code> , <code>\AtEveryStanza</code> and	
<code>\AtEveryStopStanza</code>	1
Add third and fourth optional argument of <code>\newverse</code>	1
Fix bug when using familiar footnotes in <code>\eledsection</code> and related.	1
Reset font specification at the beginning of familiar footnotes.	1
v2.21.0.	
General: Add the possibility of nested <code>\sameword</code>	1

Fix bug when using formatting command in the argument of <code>\edindex</code> inside <code>\edtext</code>	1
Now, as explained in the handbook, an <code>\edindex</code> inside <code>\edtext</code> only creates index reference to main text, and not to the critical footnote.	1
v2.22.0.	
General: Add <code>\txtbeforenotesonlyonceX</code> and <code>\Xtxtbeforenotesonlyonce</code> hooks. .	1
Add <code>\txtbeforenotesX</code> hook.	1
Fix bug added in v2.16.0 when using <code>\Xtxtbeforenotes</code> with paragraphed or normal footnotes.	1
Fix bug with three and two columns critical footnotes, broken in v. 2.17.6.	1
v2.22.1.	
General: Compatibility with new version of <code>reledpar</code>	1
Fix bug with some commands inside <code>\sameword</code>	1
v2.22.2.	
General: Do not print footnotes at the first run.	1
v2.23.0.	
General: Add <code>swcaseinsensitive</code> option	1
Optimisation of the code added on v2.22.2 to not print footnotes at the first run.	1