

ledmac

A presumptuous attempt to port EDMAC, TABMAC and EDSTANZA to LaTeX*

Peter Wilson
Herries Press[†]
Maïeul Rouquette[‡]

based on the original work by
John Lavagnino, Dominik Wujastyk, Herbert Breger and Wayne Sullivan

Abstract

For over ten years EDMAC, a set of PLAIN T_EX macros, has been available for typesetting critical editions in the traditional way, i.e., similar to the Oxford Classical Texts, Teubner, Arden Shakespeare and other series. A separate set of PLAIN T_EX macros, TABMAC, provides for tabular material. Another set of PLAIN T_EX macros, EDSTANZA, assists in typesetting verse.

The ledmac package makes the EDMAC, TABMAC and EDSTANZA facilities available to authors who would prefer to use LaTeX. The principal functions provided by the package are marginal line numbering and multiple series of footnotes and endnotes keyed to line numbers.

In addition to the EDMAC, TABMAC and EDSTANZA functions the package also provides for index entries keyed to both page and line numbers. Multiple series of the familiar numbered footnotes are also available.

Other LaTeX packages for critical editions include EDNOTES, and poemscol for poetical works.

To report bugs, please go to ledmac's GitHub page and click "New Issue": <https://github.com/maieul/ledmac/issues/>. You must open an account with github.com to access my page (maieul/ledmac). GitHub accounts are free for open-source users.

Contents

1	Introduction	5
1.1	Overview	5
1.2	History	7

*This file (ledmac.dtx) has version number v0.18.2, last revised 2012/09/05.

[†]herries dot press at earthlink dot net

[‡]maieul at maieul dot net

1.2.1	EDMAC	7
1.2.2	ledmac	8
2	The ledmac package	8
3	Numbering text lines and paragraphs	9
3.1	Lineation commands	11
3.2	Changing the line numbers	12
4	The apparatus	14
4.1	Alternate footnote formatting	16
4.2	Display options	17
4.3	Creating a new series	18
5	Fonts	19
6	Verse	21
7	Grouping	23
8	Crop marks	24
9	Endnotes	24
10	Cross referencing	24
11	Side notes	26
12	Familiar footnotes	27
13	Indexing	28
14	Tabular material	29
15	Miscellaneous	32
15.1	Hints	33
15.2	Known and suspected limitations	33
15.3	Use with other packages	34
15.4	Parallel typesetting	35
15.5	Notes for EDMAC users	36
16	Implementation overview	39
17	Preliminaries	39
17.1	Messages	41
18	Sectioning commands	43

19 Line counting	46
19.1 Choosing the system of lineation	46
19.2 List macros	50
19.3 Line-number counters and lists	51
19.4 Reading the line-list file	55
19.5 Commands within the line-list file	56
19.6 Writing to the line-list file	63
20 Marking text for notes	66
20.1 <code>\edtext</code> and <code>\critext</code> themselves	67
20.2 Substitute lemma	72
20.3 Substitute line numbers	72
21 Paragraph decomposition and reassembly	73
21.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	73
21.2 Processing one line	76
21.3 Line and page number computation	77
21.4 Line number printing	80
21.5 Pstart number printing in side	84
21.6 Add insertions to the vertical list	85
21.7 Penalties	86
21.8 Printing leftover notes	87
22 Footnotes	88
22.1 Fonts	88
22.2 Outer-level footnote commands	89
22.3 Normal footnote formatting	92
22.4 Standard footnote definitions	97
22.5 Paragraphed footnotes	99
22.5.1 Insertion of footnotes separator	104
22.6 Columnar footnotes	104
22.7 Display	108
22.7.1 Options	108
23 Output routine	117
24 Cross referencing	124
25 Endnotes	128
26 Side notes	131
27 Familiar footnotes	135
27.1 The A series footnotes	137
27.2 Footnote formats	137
27.2.1 Two column footnotes	140
27.2.2 Three column footnotes	142

27.2.3 Paraphrased footnotes	143
27.3 Other series footnotes	145
28 Minipages and such	147
29 Indexing	150
30 Macro as environment	152
31 Verse	155
32 Arrays and tables	158
33 The End	178
A Historic compatibility	179
A.1 Version 0.18	179
A.2 Version 0.18.1	179
B Examples	179
B.1 Simple example	188
B.2 General example of features	189
B.3 Gascoigne	192
B.4 Shakespeare	195
B.5 Classical text edition	198
B.6 Nijmegen	204
B.7 Irish verse	208
References	213
Index	213
Change History	232

List of Figures

1	Output from <code>ledeasy.tex</code>	181
2	Output from <code>ledfeat.tex</code>	182
3	Output from <code>ledioc.tex</code>	183
4	Output from <code>ledarden.tex</code>	184
5	Output from <code>ledmixed.tex</code>	185
6	Output from <code>ledekker.tex</code>	186
7	Output from <code>ledbraonain.tex</code>	187

1 Introduction

The EDMAC macros [LW90] for typesetting critical editions of texts have been available for use with TeX for some years. Since EDMAC was introduced there has been a small but constant demand for a version of EDMAC that could be used with LaTeX. The ledmac package is an attempt to satisfy that request.

ledmac would not have been possible without the amazing work by John Lavagnino and Dominik Wujastyk, the original authors of EDMAC. I am very grateful for their encouragement and permission to use EDMAC as a base. The majority of both the code and this manual are by these two. The tabular material is based on the TABMAC code [Bre96], by permission of its author, Herbert Breger. The verse-related code is by courtesy of Wayne Sullivan, the author of EDSTANZA [Sul92], who has kindly supplied more than his original macros.

I have altered their code and documentation as little as possible. In order to more easily show the debt that I owe, my few contributions are in the font you are now reading. I have not noted minor editorial changes such as replacing ‘TeX’ with ‘LaTeX’ or replacing ‘EDMAC’ with ‘ledmac’ or ‘package’. The original work is in the normal roman font.

There are places where I have not supplied some of the original EDMAC facilities, either because they are natively provided by LaTeX (such as font handling), or are available from other LaTeX packages (such as crop marks).

1.1 Overview

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

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

While ledmac can be used ‘out of the box’, with little or no customization, you may also go to the other extreme and view it as a collection of tools. Critical editions are amongst the most idiosyncratic of books (like their authors), so we have made ledmac deliberately bland in some ways, while also trying to document it reasonably well so that you can find out how to make it do what you want.

The original EDMAC can be used as a ‘stand alone’ processor or as part of a process. One example is its use as the formatting engine or ‘back end’ for the output of an automatic manuscript collation program. COLLATE, written by Peter Robinson, runs on the Apple Macintosh, can collate simultaneously up to a hundred manuscripts of any length, and provides facilities for the scholar to tailor the collation interactively. For further details of this and other related work, visit the EDMAC home page at <http://www.homepages.ucl.ac.uk/~ucgadkw/edmac/index.html>.

Apart from ledmac there are some other LaTeX packages for critical edition typesetting. As I am not an author, or even a prospective one, of any critical edition work I cannot provide any opinions on what authors in this area might feel comfortable with or how well any of the packages meet their needs.

EDNOTES [Lüc03], by Uwe Lück and Christian Tapp, is another LaTeX package being developed for critical editions. Unlike ledmac which is based on EDMAC, EDNOTES takes a different (internal) approach and provides a different set of features. For example it provides additional facilities for overlapping lemmas and for handling tables. For more information there is a web site at <http://ednotes.sty.de.vu> or email to ednotes.sty@web.de.

The poemscol package [Bur01] by John Burt is designed for typesetting critical editions of collections of poems. I do not know how, or whether, poemscol and ledmac will work together.

Critical authors may find it useful to look at EDMAC, EDNOTES, ledmac, and poemscol to see which best meets their needs.

At the time of writing I know of two web sites, apart from the EDMAC home page, that have information on ledmac, and other programs.

- Jerónimo Leal pointed me to <http://www.guit.sssup.it/latex/critical.html>. This also mentions another package for critical editions called MauroTeX (<http://www.maurolico.unipi.it/mtex/mtex.htm>). These sites are both in Italian.
- Dirk-Jan Dekker maintains <http://www.djdekker.net/ledmac> which is a FAQ for typesetting critical editions and ledmac.

This manual contains a general description of how to use the LaTeX version of EDMAC, namely ledmac, (in sections 2 through 15.5); the complete source code for the package, with extensive documentation (in sections 16 through 33); a series of examples (in Appendix B); and an Index to the source code. We do not suggest that you 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 the earlier sections. But no documentation, however thorough, can cover every question that comes up, and many can be answered quickly by consultation of the code. On a first reading, we suggest that you should skip from the general documentation in sections 2 through 15.5 to the examples in Appendix B, unless you are particularly interested in the innards of ledmac.

1.2 History

1.2.1 EDMAC

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

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

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

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

At the time of writing (1994), we are pleased to be able to say that EDMAC is being used for 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

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

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

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

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

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, as well as the editions illustrated in Appendix B.

1.2.2 ledmac

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

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

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

Since July 2011, ledmac is maintained by Maeul Rouquette.

Important changes were put in version 0.18 (see 4.2 p. 17). They can make some little troubles with old customization. To correct it, please read A.1 (p. 179).

2 The ledmac package

ledmac is a three-pass package like LaTeX itself. Although your textual apparatus and line numbers will be printed even on the first run, it takes two more passes through LaTeX to be sure that everything gets to its right place. Any changes you make to the input file may similarly require three passes to get everything to the right place, if the changes alter the number of lines or notes. ledmac will tell you that you need to make more runs, when it notices, but it does not expend the labor

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: Schiphoewer en Brinkman, 1993).

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

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

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

¹²Being prepared at Poona and Lausanne Universities.

to 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.

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'll want to print the text that you're editing. Unnumbered text is not printed with line numbers, and you can't use `ledmac`'s note commands with it: this is appropriate for introductions and other material added by the editor around the edited text.

3 Numbering text lines and paragraphs

`\beginnumbering` Each section of numbered text must be preceded by `\beginnumbering` and followed by `\endnumbering`, like:

```
\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>.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. `ledmac` 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).

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

```
\pstart
<paragraph of text>
\pend
```

Text that appears within a numbered section but isn't 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

```

1 This is a sample paragraph
2 with lines numbered
3 automatically.

```

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

```

4 This paragraph too
5 has its lines automatically
6 numbered.

The lines of this paragraph are
not numbered.

The lines of this paragraph
are not numbered.

```

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

```

7 And here the numbering
8 begins again.

`\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.

1 A paragraph of numbered
2 text.

Another paragraph of numbered
text.

3 Another paragraph of
4 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

`\firstlinenum`
`\linenumincrement`

By default, `ledmac` numbers every 5th line. There are two counters, `firstlinenum` and `linenumincrement`, that control this behaviour; 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>}`

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

for controlling sub-line numbering.

<code>\pausenumbering</code> <code>\resumenumbering</code>	<p>ledmac 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 LaTeX may reach its memory limit. There are two solutions to this. The first is to get a larger LaTeX with increased memory. The second solution is to split your long section into several smaller ones. The trouble with this is that your line numbering will start again at zero with each new section. To avoid this problem, we provide <code>\pausenumbering</code> and <code>\resumenumbering</code> which are just like <code>\endnumbering ... \beginnumbering</code>, except that they arrange for your line numbering to continue across the break. Use <code>\pausenumbering</code> only between numbered paragraphs:</p> <pre> \beginnumbering \pstart Paragraph of text. \pend \pausenumbering \resumenumbering \pstart Another paragraph. \pend \endnumbering </pre>
---	---

```

1 Paragraph of
2 text.
3 Another paragraph.

```

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 say

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

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

<code>\numberpstarttrue</code> <code>\numberpstartfalse</code> <code>\thepstart</code>	<p>It's possible to insert a number at every <code>\pstart</code> command. You must use the <code>\numberpstarttrue</code> command to have it. You can stop the numbering with <code>\numberpstartfalse</code>. You can redefine the command <code>\thepstart</code> to change style. On each <code>\beginnumbering</code> the numbering restarts. With the <code>\sidepstartnumtrue</code> command, the number of <code>\pstart</code> will be printed in side. In this case, the line number will be not printed.</p>
--	---

3.1 Lineation commands

<code>\numberlinefalse</code> <code>\numberlinetrue</code> <code>\lineation</code>	<p>Line numbering can be disabled with <code>\numberlinefalse</code>. It can be enabled again with <code>\numberlinetrue</code>. Lines can be numbered either by page, by <code>pstart</code> or by section; you specify this using the <code>\lineation{<arg>}</code> macro, where <code><arg></code> is either <code>page</code>, <code>pstart</code> or <code>section</code>. 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</p>
--	--

	system. The package's standard setting is <code>\lineation{section}</code> . If the lineation is by <code>pstart</code> , the <code>pstart</code> number will be printed before the line number in the notes.
<code>\linenummargin</code>	The command <code>\linenummargin{location}</code> specifies the margin where the line numbers will be printed. The permissible value for <code>location</code> is one out of the list <code>left</code> , <code>right</code> , <code>inner</code> , or <code>outer</code> , for example <code>\linenummargin{inner}</code> . The package's default setting is <code>\linenummargin{left}</code> 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 <code>\linenummargin</code> used is that in effect at the <code>\pend</code> of a numbered paragraph. Apart from an initial setting for <code>\linenummargin</code> , only change it after a <code>\pend</code> , whereupon it will apply to all following numbered paragraphs, until changed again (changing it between a <code>\pstart</code> and <code>\pend</code> pair will apply the change to all the current paragraph).
<code>\firstlinenum</code> <code>\linenumincrement</code> <code>\firstsublinenum</code> <code>\sublinenumincrement</code>	In most cases, you will not want a number printed for every single line of the text. Four LaTeX counters control the printing of marginal numbers and they can be set by the macros <code>\firstlinenum{num}</code> , etc. <code>\firstlinenum</code> specifies the number of the first line in a section to number, and <code>\linenumincrement</code> is the increment between numbered lines. <code>\firstsublinenum</code> and <code>\sublinenumincrement</code> do the same for sub-lines. Initially, all these are set to 5 (e.g., <code>\firstlinenum{5}</code>).
<code>\linenumberlist</code>	You can define <code>\linenumberlist</code> to specify a non-uniform distribution of printed line numbers. For example: <code>\def\linenumberlist{1,2,3,5,7,11,13,17,19,23,29}</code> to have numbers printed on prime-numbered lines only. There must be no spaces within the definition which consists of comma-separated decimal 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 <code>\linenumberlist</code> or following the vacuous definition <code>\def\linenumberlist{}</code> the standard numbering sequence is applied. The standard sequence is that specified by the combination of the <code>firstlinenum</code> , <code>linenumincrement</code> , <code>firstsublinenum</code> and <code>linenumincrement</code> counter values.
<code>\leftlinenum</code> <code>\rightlinenum</code> <code>\linenumsep</code>	When a marginal line number is to be printed, there are a lot of ways to display it. You can redefine <code>\leftlinenum</code> and <code>\rightlinenum</code> to change the way marginal line numbers are printed in the left and right margins respectively; the initial versions print the number in font <code>\numlabfont</code> (described below) at a distance <code>\linenumsep</code> (initially set to one pica) from the text.

3.2 Changing the line numbers

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

<code>\startsub</code> <code>\endsub</code>	You insert the <code>\startsub</code> and <code>\endsub</code> commands in your text to turn sublineation on and off. In plays, for example, stage directions 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 that changes in the middle.

\startlock The **\startlock** command, used in running text, locks the line number at its current value, until you say **\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.

\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. (This assumes that, on the basis of the settings of the previous parameters, it is necessary to display 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}**.

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

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

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

Alph Uppercase letters (A...Z).

alph Lowercase letters (a...z).

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

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

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

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

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

`\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.

4 The apparatus

`\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:

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

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

<pre>\Afootnote \Bfootnote \Cfootnote \Dfootnote \Efootnote</pre>	<p>Five separate series of footnotes are maintained; each macro taking one argument like <code>\Afootnote{<text>}</code>. When all five are used, the A notes appear in a layer just below the main text, followed by the rest in turn, down to the E notes at the bottom. These are the main macros that you will use to construct the critical apparatus of your text. The package provides five layers of notes in the belief that</p>
---	---

this will be adequate for the most demanding editions. But it is not hard to add further layers of notes should they be required. A optional argument can be add before the texte of footnote. His value is a comma separated list of options. The options are :

- `nonum` to disable line numbering for this note.
- `nosep` to disable lemma separator for this note.

`\Aendnote` Example : `\Afootnote[nonum]{texte}`. The package also maintains five separate series of endnotes. Like footnotes each macro takes a single argument
`\Bendnote` like `\Aendnote{<text>}`. Normally, none of them is printed: you must use the
`\Cendnote` `\doendnotes` macro described below (p.24) to call for their output at the appropriate point in your document.
`\Dendnote`
`\Eendnote`

`\lemma` Sometimes 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`, before the note commands. The most common use of this command is to abbreviate the lemma that's printed in the notes. For example:

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

`\linenum` You can use `\linenum{<arg>}` to change the line numbers passed to the notes. The notes are actually given 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). However, you can retain the value computed by `ledmac` 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 one number, the ending page number of the current lemma.

This command doesn't change the marginal line numbers in any way; it just changes the numbers passed to the footnotes. 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 (p.24) 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.

Changing the names of these commands 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 doesn't mean you have to type `\Afootnote` when you'd rather say 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:

```
\let\variant=\Afootnote
\let\explanatory=\Bfootnote
\let\trivial=\Aendnote
\let\testimonia=\Cfootnote
```

4.1 Alternate footnote formatting

If you just launch into `ledmac` using the commands outlined above, you will get a standard layout for your text and notes. You may be happy to accept this at the very beginning, while you get the hang of things, but the standard layout is not particularly pretty, and you will certainly want to modify it in due course. The package provides ways of changing the fonts and layout of your text, but these are not aimed at being totally comprehensive. They are enough to deal with simple variations from the norm, and to exemplify how you might go on to make more swingeing changes.

`\footparagraph` All footnotes will normally be formatted as a series of separate paragraphs in one column. But there are three other formats available for notes, and using these macros you can select a different format for a series of notes.

- `\footparagraph` formats all the footnotes of a series as a single paragraph (see figs. 3 and 5, pp.183 and 185);
- `\foottwocol` formats them as separate paragraphs, but in two columns (see bottom notes in fig. 4, p.184);
- `\footthreecol`, in three columns (see second layer of notes in fig.2, p.182).

Each of these macros takes one argument: a letter (between A and E) for the series of notes you want changed. So a text with three layers of notes might begin thus:

```
\footnormal{A}
\footthreecol{B}
\footparagraph{C}
```


This would make the A-notes ordinary, B-notes would be in three columns, and the bottom layer of notes would be formed into a paragraph on each page.

`\interparanoteglue` If you use paragraphed footnotes, the macro `\interparanoteglue` defines the glue appearing in between footnotes in the paragraph. It is a macro whose argument is the glue you want, and its initial setting is (see p. ??):

```
\interparanoteglue{1em plus .4em minus .4em}
```

4.2 Display options

Since version 0.18, some commands can be used to change display of footnote. All can have a optional argument, which is the letter of the series which option is applied.

When a length is used, it can be stretchable : `a minus b minus c`. The final length `m` is calculated by L^AT_EX to have : $b - a \leq m \leq b + c$. If you use relative unity¹⁴, it will be relative to fontsize of the footnote.

`\numberonlyfirstinline` By default, line number is printed everytime in notes. If you want to print it only the first time for a value (i.e one time for line 1, one time for line 2 etc.), you can use `\numberonlyfirstinline[⟨series⟩]`. To cancel it, use `\numberonlyfirstinline[⟨series⟩][⟨false⟩]` (`⟨series⟩` can be empty).

`\symlinenum` You can use `\symlinenum[⟨series⟩]{⟨symbol⟩}` in combination with `\numberonlyfirstinline[⟨series⟩]`. From the second lemma of the same line, the symbol will be used instead of line number.

`\nonumberinfootnote` If you don't want to have line number in footnote, you can use `\nonumberinfootnote[⟨series⟩]`. To cancel it, use `\nonumberinfootnote[⟨series⟩][⟨false⟩]` (`⟨series⟩` can be empty).

`\beforenumberinfootnote` You can add some space before line number in footnote with `\beforenumberinfootnote[⟨series⟩]{⟨length⟩}`. If the line number is not printed, the space is not too. Default value is 0 pt.

`\afternumberinfootnote` You can add some space after line number in footnote with `\afternumberinfootnote[⟨series⟩]{⟨length⟩}`. If the line number is not printed, the space is not too. Default value is 0.5 em.

`\beforesymlinenum` You can add some space before line symbol in footnote, with `\beforesymlinenum[⟨series⟩]{⟨length⟩}`.
`\aftersymlinenum` Default value is value set by `\beforenumberinfootnote`. You can add some space before line symbol in footnote, with `\aftersymlinenum[⟨series⟩]{⟨length⟩}`. Default value is value set by `\afternumberinfootnote`.

`\inplaceofnumber` If no number or symbolic line number is printed, you can add a space, with `\inplaceofnumber[⟨series⟩]{⟨length⟩}`. Default value is 1 em.

`\lemmaseparator` By default, in footnote, separator between lemma and note is a right bracket (`\rbracket`). You can use `\lemmaseparator[⟨series⟩]{⟨lemmaseparator⟩}` to change it. The optional argument can be used to precise in which series it is applied. Note that there is a no breakable space between lemma and separator, but **breakable** space between separator and lemma.

`\beforelemmaseparator` You can add some space between lemma and separator, by using `\beforelemmaseparator[⟨series⟩]{⟨length⟩}`. If your lemma separator is empty, this space won't be printed. Default value is 0 em.

¹⁴Like `em` which is the width of a M.

<code>\afterlemmaseparator</code>	You can add some space between separator and note, by using <code>\afterlemmaseparator[⟨series⟩]{}</code> . If your lemma separator is empty, this space won't be printed. Default value is 0.5 em.
<code>\nolemmaseparator</code>	You can suppress lemma separator, using <code>\nolemmaseparator[⟨series⟩]</code> , which is only a alias of <code>\lemmaseparator[⟨series⟩]{}</code> .
<code>\inplaceoflemmaseparator</code>	If no lemma separator is printed, you can add a space, with <code>\inplaceoflemmaseparator[⟨series⟩]</code> . Default value is 1 em.
<code>\afternote</code>	You can add some space after note, by using <code>\afternote[⟨series⟩]{⟨length⟩}</code> . The default value is 1em plus.4em minus.4em.
<code>\parafootsep</code>	For paragraphed footnotes (see below), you can choice the separator between each not by <code>\parafootsep[⟨series⟩]{⟨length⟩}</code> . A common separator is double pipe (<code>.</code>). So you can set by <code>\parafootsep\$ \$</code> . You should set up the page layout parameters, and in particular the <code>\baselineskip</code> of the footnotes (this is done for you if you use the standard <code>\notefontsetup</code>), before you call any of these macros because their 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. ¹⁵

4.3 Creating a new series

If you need more than 5 series of critical footnotes you can readily create extra series. For example to create a G series you have to put the following code into either a .sty package file, or into the preamble sandwiched between `\makeatletter` and `\makeatother` declarations.

```
\newcommand*{\Gfootnote}[1]{%
  \ifnumberedpar@
    \xright@appenditem{\noexpand\Gfootnote{G}%
      {\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}\to\inserts@1
  \global\advance\insert@count by \@ne
  \else
    \vGfootnote{G}{\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}{\#1}}%
  \fi\ignorespaces}
\newinsert\Gfootins

\newcommand*{\mpGfootnote}[1]{%
  \ifnumberedpar@
    \xright@appenditem{\noexpand\mpvGfootnote{G}%
      {\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}\to\inserts@1
  \global\advance\insert@count by \@ne
  \else
    \mpvGfootnote{G}{\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}{\#1}}%
  \fi\ignorespaces}
\newinsert\mpGfootins
```

¹⁵There is one tiny proviso about using paragraphed notes: you shouldn't force any explicit line-breaks inside such notes: do not use `\par`, `\break`, or `\penalty=-10000`. If you must have a line-break for some obscure reason, just suggest the break very strongly: `\penalty=-9999` will do the trick. Page 101 explains why this restriction is necessary.

```
\addfootins{G}
\footnormal{G}
```

5 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.

For those who are setting up for a large job, here is a list of the complete set of `ledmac` macros relating to fonts that are intended for manipulation by the user: `\endashchar`, `\fullstop`, `\notefontsetup`, `\notenumfont`, `\numlabfont`, and `\rbracket`.

`\notefontsetup` The `\notefontsetup` macro defines the standard size of the fonts for all your footnotes; `ledmac` initially defines this as:

```
\newcommand*{\notefontsetup}{\footnotesize}
```

`\notenumfont` The `\notenumfont` macro specifies the font used for the line numbers printed in notes. This will typically be a command like `\bfseries` that selects a distinctive style for the note numbers, but leaves the choice of a size up to `\notefontsetup`. `ledmac` initially defines:

```
\newcommand{\notenumfont}{\normalfont}
```

thus using the main document font.

`\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.

Here are some examples of how you might redefine some of the font macros.

```
\renewcommand*{\notefontsetup}{\small}
\renewcommand*{\notenumfont}{\sffamily}
```

These commands select `\small` fonts for the notes, and choose a sans font for the line numbers within notes.

`\endashchar` A relatively trivial matter relates to punctuation. In your footnotes, there will sometimes be spans of line numbers like this: 12–34, or lines with sub-line numbers like this: 55.6. The en-dash and the full stop are taken from the same font as the numbers, and it all works nicely. But what if you wanted to use old-style numbers, like 12 and 34? These look nice in an edition, but when you use the fonts provided by PLAIN T_EX they are taken from a math font which does not have the en-dash or full stop in the same places as a text font. If you (or your macros) just typed

`\fullstop`

`\rbracket`

`\oldstyle 12--34$` or `\oldstyle 55.6$` you would get ‘12”34’ and ‘55▷6’. So we define `\endashchar` and `\fullstop`, which produce an en-dash and a full stop respectively from the normal document font, whatever font you are using for the numbers. These two macros are used in the macros which format the line numbers in the margins and footnotes, instead of explicit punctuation. We also define an `\rbracket` macro for the right square bracket printed at the end of the lemma in many styles of textual notes (including `ledmac`’s standard style).

`\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 `ledmac`’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.

`ledmac` 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 footnotes are formatted in the same way.

But it is also likely that you might want to have different fonts for just, say, the note numbers in layers A and B of your apparatus. To do this, make two copies of the `\normalfootfmt` macro (see p. 93)—or `\twocolfootfmt`, or the other appropriate macro ending in `-footfmt`, depending on what footnote format you have selected—and give these macros the names `\Afootfmt` and `\Bfootfmt`. Then, in these new macros, change the font specifications (and spacing, or whatever) to your liking.

As an example, in some texts the lemma in a footnote ends with a right bracket except where the lemma is an abbreviation (often typeset in italics). This requirement can be met as follows, assuming that the ‘A’ series footnote will be used.

First, define `\Afootfmt` as a modified version of the original `\normalfootfmt` (all the following should be enclosed in `\makeatletter` and `\makeatother` if it is in the preamble). The change is modifying `...#2\rbracket\enskip...` to read `...#2\rbracket}\enskip...`, so that `\rbracket` is inside the group that includes the lemma argument.

```
\renewcommand{\Afootfmt}[3]{%
  \ledsetnormalparstuff
  {\notenumfont\printlines#1|}\strut\enspace
  {\select@lemmafont#1|#2\rbracket}\enskip#3\strut\par}
```

Define an ‘abbreviation’ macro that kills the definition of `\rbracket`.

```
\newcommand*{\nobrak}{}
\newcommand{\abb}[1]{\textit{#1}\let\rbracket\nobrak\relax}
```

Finally, make sure that `\abb` is not expanded during the first processing of a line.

```
\newcommand{\morenoexpands}{%
  \let\abb=0%
}
```

Now code like the following can be used, and ‘lemma’ will be footnoted with a ‘`]’` and ‘abbrv’ will have no ‘`]’`.

```
A sentence with a \edtext{lemma}{\Afootnote{ordinary}} in it.
A sentence with an \edtext{\abb{abbrv}}{\Afootnote{abbreviated}} in it.
```

6 Verse

In 1992 Wayne Sullivan¹⁶ wrote the EDSTANZA macros [Sul92] for typesetting verse in a critical edition. More specifically they were for handling poetry stanzas which use indentation to indicate rhyme or metre.

With Wayne Sullivan’s permission the majority of this section has been taken from [Sul92]. I have made a few changes to enable his macros to be used in the LaTeX ledmac package.

`\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.

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

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

To specify these multiples one invokes, for example

```
\setstanzaindent{3,1,2,1,2}.
```

The numerical entries must be whole numbers, 0 or greater, separated by commas without embedded spaces. The first entry gives the hanging indentation to be used if the stanza line requires more than one print line. If it is known that each stanza line will fit on a single 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. 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,

¹⁶Department of Mathematics, University College, Dublin 4, Ireland

with 3 units of hanging indentation in case a stanza line is too long to fit on one print line.

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

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

is like

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

If you don't use the `stanzaindentrepetition` counter, make sure you have at least one more numerical entry in `\setstanzavalues` than the number of lines in the stanza. The macros make no restriction on the number of lines in a stanza. Stanza indentation values (and penalty values) obey T_EX'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.

`\setstanzapenalties`

When the stanzas run over several pages, often it is 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_EX, 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. The control sequence `\endstanzaextra` can be defined to include a penalty. 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.

`\ampersand`

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

`\endstanzaextra`

The macro `\endstanzaextra`, if it is defined, is called at the end of a stanza. You could define this, for example, to add extra space between stanzas (by default there is no extra space between stanzas); if you are using the memoir class, it provides a length `\stanzaskip` which may come in handy.

`\startstanzahook` Similarly, if `\startstanzahook` is defined, it is called by `\stanza` at the start. This can be defined to do something.

`\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`.

For example, to put a verse number before the first line of a stanza you could proceed along the lines:

```
\newcounter{stanzanum}
\setcounter{stanzanum}{0}
\newcommand*{\startstanzahook}{\refstepcounter{stanzanum}}
\newcommand{\numberit}{\flagstanza{\thestanzanum}}
...
\stanza
\numberit First line...&
    rest of stanza&

\stanza
\numberit First line, second stanza...
```

`\hangingsymbol` It's possible to insert a symbol on each line of verse's hanging, as in French typography for '['. To insert in ledmac, redefine macro `\hangingsymbol` with this code :

```
\renewcommand{\hangingsymbol}{[,]}
```

7 Grouping

In a minipage environment 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 12) in a minipage but unlike with `\footnote` the numbering scheme is unaltered.

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

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

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

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

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

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

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

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

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

8 Crop marks

The `ledmac` package does not provide crop marks. These are available with either the `memoir` class [Wil02] or the `crop` package.

9 Endnotes

`\doendnotes` `\doendnotes{<letter>}` closes the `.end` file that contains the text of the endnotes, if it's open, and prints one series of endnotes, as specified by a series-letter argument, e.g., `\doendnotes{A}`. `\endprint` is the macro that's called to print each note. It uses `\notenumfont`, `\select@lemmfont`, and `\notefontsetup` to select fonts, just as the footnote macros do (see p. 19 above).

As endnotes may be printed at any point in the document they always start with the page number of where they were specified. The macro `\printnpnum{<num>}` is used to print these numbers. Its default definition is:

`\newcommand*{\printnpnum}[1]{p.#1}`

`\noendnotes` If you aren't going to have any endnotes, you can say `\noendnotes` in your file, before the first `\beginnumbering`, to suppress the generation of an unneeded `.end` file.

10 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.

`\edlabel` First you place a label in the text using the command `\edlabel{<lab>}`. `<lab>` can be almost anything you like, including letters, numbers, punctuation, or a combination—anything but spaces; you might say `\edlabel{toves-3}`, for example.¹⁷

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

An `\edlabel` command may appear in the main text, or in the first argument of `\edtext`, but not in the apparatus itself. But `\edpageref`, `\lineref` and

¹⁷More precisely, you should stick to characters in the T_EX categories of 'letter' and 'other'.

`\sublineref` commands can also be used in the apparatus to refer to `\edlabel`'s in the text.

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

You will be warned if you say `\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.)

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

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

`\xpageref` However, there are situations in which you'll want `ledmac` to return a number
`\xlineref` without displaying any warning messages about undefined labels or the like: if
`\xsublineref` you want to use the reference in a context where LaTeX 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. For this
situation, three variants of the reference commands, with the `x` prefix, are supplied:
`\xpageref`, `\xlineref`, and `\xsublineref`. They have these limitations: they
will not tell you if the label is undefined, and 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.

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

The `\xxref{<lab1>}{<lab2>}` command generates a reference to a sequence of
lines, for use in the second argument of `\edtext`. It takes two arguments, both
of which are labels: e.g., `\xxref{mouse}{elephant}`. It calls `\linenum` (q.v.,
p.15 above) and sets the beginning page, line, and sub-line numbers to those of
the place where `\edlabel{mouse}` was placed, and the ending numbers to those
where `\edlabel{elephant}` occurs.

`\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 say '`\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 docu-
ment, so that you can see them at a glance.

`\label` The normal `\label`, `\ref` and `\pageref` macros may be used within numbered
`\ref`
`\pageref`

text, and operate in the familiar fashion. As an example, here is one way of numbering paragraphs in numbered text, and then being able to refer to the paragraph numbers, in addition to line and page numbers.

```
\newcounter{para} \setcounter{para}{0}
\newcommand{\newpara}{%
  \refstepcounter{para}%
  \noindent\llap{\thepar. }\quad}
\newcommand{\oldpara}[1]{%
  \noindent\llap{\ref{#1}. }\quad}
```

The definitions of `\newpara` and `\oldpara` put the numbers in the left margin and the first line of the paragraph is indented. You can now write things like:

```
\linenummargin{right}
\beginnumbering
\pstart
\newpara\label{P1} A paragraph about \ldots
\pend
  In paragraph~\ref{P1} the author \ldots
\pstart
\oldpara{P1} This has the same
  \edtext{number}{\Afootnote{\ref{P1} is the paragraph, not line}}
  as the first paragraph.
\pend
\endnumbering
```

11 Side notes

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

`\ledleftnote` `\ledleftnote{<text>}` will put `<text>` into the left margin level with where the command was issued. Similarly, `\ledrightnote{<text>}` puts `<text>` in the right margin. `\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}`. The package's default setting is

```
\sidenotemargin{right}
```

to typeset `\ledsidenotes` in the right hand margin. This is the opposite to 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, say, `\ledleftnote`, commands are called in the same line the second `<text>` will obliterate the first. There is no problem though with having both a left and a right sidenote on the same line.

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

text into a box of width `\ledrsnotewidth`. These are initially set to the value of `\marginparwidth`.

`\ledlsnotesep` The texts are put a distance `\ledlsnotesep` (or `\ledrsnotesep`) into the left
`\ledrsnotesep` (or right) margin. These lengths are initially set to the value of `\linenumsep`.
`\ledlsnotefontsetup` These macros specify how the sidenote texts are to be typeset. The initial defini-
`\ledrsnotefontsetup` tions are:

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

These can of course be changed to suit.

12 Familiar 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 `ledmac` provides this automatically.

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

```
\providecommand*{\multfootsep}{\textsuperscript{\normalfont,}}
and can be changed if necessary.
```

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

`\footnormalX` Each of the `\foot...X` macros takes one argument which is the series letter (e.g.,
`\footparagraphX` B). `\footnormalX` is the typical footnote format. With `\footparagraphX` the series
`\foottwocolX` is typeset a one paragraph, with `\foottwocolX` the notes are in two columns, and
`\foothreecolX` are in three columns with `\foothreecolX`.

`\thefootnoteA` As well as using the `\foot...X` macros to specify the general footnote arrangement
for a series, each series uses a set of macros for styling the marks. The mark numbering
scheme is defined by the `\thefootnoteA` macro; the default is:

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

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

```
\newcommand*{\bodyfootmarkA}{%
\hbox{\textsuperscript{\normalfont\thefootnoteA}}}
```

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

```
\newcommand*{\footfootmarkA}{\textsuperscript{\thefootnoteA}}
```

There are similar command triples for the other series.

Additional footnote series can be easily defined. For example, to specify a D series you have to specify the following code, either in a .sty package file or in the preamble sandwiched between `\makeatletter` and `\makeatother` commands.

```
\newcommand{\footnoteD}[1]{%
  \refstepcounter{footnoteD}%
  \@footnotemarkD
  \vfootnoteD{D}{#1}\m@mmf@prepare}
\newcounter{footnoteD}
\renewcommand{\thefootnoteD}{\arabic{footnoteD}}
\newinsert\footinsD

\newcommand{\mpfootnoteD}[1]{%
  \refstepcounter{footnoteD}%
  \@footnotemarkD
  \mpvfootnoteD{D}{#1}\m@mmf@prepare}
\newinsert\footins\mpfootinsD

\addfootinsX{D}
\footnormalX{D}
```

The above creates the D series with the default layout, and perhaps that is all that is required. If not, then you can now start to specialise it. For instance, to have the marks in the main text as lowercase roman numerals in parentheses, the marks in the foot on the baseline with a single closing parenthesis, and using the paragraph style:

```
\renewcommand*{\thefootnoteD}{\roman{footnoteD}}
\renewcommand*{\bodyfootmarkD}{\hbox{\textsuperscript{(\thefootnoteD)}}}
\renewcommand*{\footfootmarkD}{\thefootnoteD} }
\footparagraphX{D}
```

13 Indexing

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

If the memoir class 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`.

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

`\edindexlab` The `\edindex` process uses a `\label/\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 Tabular material

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

<code>edarrayl</code> <code>edarrayc</code> <code>edarrayr</code> <code>edtabularl</code> <code>edtabularc</code> <code>edtabularr</code>	<p>There are six environments; the <code>edarray*</code> environments are for math and <code>edtabular*</code> for text entries. The final <code>l</code>, <code>c</code>, or <code>r</code> in the environment names indicate that the entries will be flushleft (<code>l</code>), centered (<code>c</code>) or flushright (<code>r</code>). 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.</p> <pre>\begin{edtabularc} 1 & 2 & 3 \\ a & bb & ccc \\ AAA & BB & C \end{edtabularc}</pre> <table border="0" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> </tr> <tr> <td style="text-align: center;">a</td> <td style="text-align: center;">bb</td> <td style="text-align: center;">ccc</td> </tr> <tr> <td style="text-align: center;">AAA</td> <td style="text-align: center;">BB</td> <td style="text-align: center;">C</td> </tr> </table>	1	2	3	a	bb	ccc	AAA	BB	C
1	2	3								
a	bb	ccc								
AAA	BB	C								

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

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

For example:

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

produces the following parallel pair of verses.

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

`\edtabcolsep` The distance between the columns is controlled by the length `\edtabcolsep`.
`\spreadmath` `\spreadmath{<math>}` typesets `{<math>}` but the `{<math>}` has no effect on the
`\spreadtext` calculation of column widths. `\spreadtext{<text>}` is the analagous 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	a bb ccc dddd
<code>\end{edarrayl}</code>	

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

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

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

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

1	2	3	4	5
Q		fd	h	qwertziohg
v	wptz	x	y	vb
g	nnn	\upbracefill		
\downbracefill		pq	dgh	
k		l	co	ghweropjklmnbvcxys
1	2	3	<hr/>	

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:

[illegible]

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<symbol>` 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<symbol>` with $\langle math \rangle$ centered after it.

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

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

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

For example:

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

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

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

`\edvertdots`

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

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

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

15 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
```


15.1 Hints

One user has wanted a big space between the text and footnotes but a smaller space between each series. That is, the first printed series on a page must have a big skip and all later ones a small skip. Of course, there is no telling which will be the first on any given page; on one page there might be A, C and E series and on the next D and E.

Here is the start of a solution.

```
\newskip\prefootsskip % the big initial skip
\prefootsskip=3.3em plus .6em minus .6em
\newif\ifskipped \skippedfalse
\renewcommand*{\normalfootstart}[1]{%
  \ifskipped
    \vskip\skip\csname #1footins\endcsname% normal skip
  \else
    \skip\prefootsskip%      first note so big skip
    \skippedtrue
  \fi
  \leftskip0pt\rightskip0pt
  \csname #1footnoterule\endcsname}
\footnormal{A}% make sure the new \normalfootstart is used
\footnormal{B}
...
```

In addition similar changes would be required for paragraphed footnotes, footnotes in minipages, and the familiar footnotes.

15.2 Known and suspected limitations

In general, ledmac'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.

`\parshape` cannot be used within numbered text, except in a very restricted way (see p. ??).

`\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, ledmac 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, say

```
\setcounter{ballast}{100}
```

or some such figure, and with any luck the page breaks will settle down. Luckily, this problem doesn't crop up at all often.

The restriction on explicit line-breaking in paragraphed footnotes, mentioned in footnote 15, p. 18, and described in more detail on p. 101, really is a nuisance if that's something you need to do. There are some possible solutions, described by Michael Downes, but this area remains unsatisfactory.

LaTeX has a reputation for putting things in the wrong margin after a page break. The ledmac package does nothing to improve the situation — in fact it just makes it more obvious if numbered text crosses a page (or column) boundary and the numbers are meant to flip from side to side. Try and keep the numbers in the same margin all the time. Another aspect of TeX's page breaking mechanism is that when numbering lines by the page, the first few numbers after a page break may continue as though the lines were still on the previous page.

`\pageparbreak`

If you can't resist flipping the numbers or numbering by the page, then you might find that judicious use of `\pageparbreak` may help if numbering goes awry across a page (or column) break. It tries to force TeX into partitioning the current paragraph into two invisibly joined paragraphs with a page break between them. Insert the command between the last word on one page and the first word on the next page. If later you change something earlier in the document the natural page break may be in a different place, and you will have to adjust the location of `\pageparbreak` accordingly.

`\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}
```

Help, suggestions and corrections will be gratefully received.

15.3 Use with other packages

Because of ledmac's complexity it may not play well with other packages. In particular ledmac is sensitive to commands in the arguments to the `\edtext` and `*footnote` macros (this is discussed in more detail in section 20, 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.

It is possible that ledmac and the hyperref package may work together. I have not tried this combination but past experience with hyperref suggests that cooperation is unlikely; hyperref changes many LaTeX internals and ledmac does things that are not normally seen in LaTeX.

`\morenoexpands`

You can define the macro `\morenoexpands` to modify macros that you call within `\edtext`. Because of the way ledmac 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 this¹⁸ 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
\newcommand{\morenoexpands}{\let\textcolor\@secondoftwo}
\makeatother
```

It took me 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.

15.4 Parallel typesetting

ledmac and the parallel package [Eck03] do not work together — they have very different ideas about footnoting — and I do not have the skills to try and get them to cooperate. If you are trying to typeset short pieces in parallel on the same page you can try using the edtabular environment.

More likely you are wanting to typeset in parallel on opposite pages (e.g., original on the left (even numbered) pages and a translation on the right (odd numbered) pages). Essentially you will have to do all the page breaking yourself. Here’s some example code that might help, though.

```
\makeatletter
\providecommand{\cleartoevenpage}{% defined in the memoir class
```

¹⁸Reported by Dirk-Jan Dekker in the CTT thread ‘Incompatibility of “color” package’ on 2003/08/28.

```

\clearpage%
\ifodd\c@page\hbox{}\clearpage\fi}
\providecommand{\cleartooddpage}{% defined in the memoir class
\clearpage%
\ifodd\c@page\else\hbox{}\clearpage\fi}
\makeatother
\newenvironment{parallelpages}{\cleartoevenpage}{}
\newcommand{\leftpage}{\cleartoevenpage}
\newcommand{\rightpage}{\cleartooddpage}
...
\begin{parallelpages}
\leftpage{first left page text}
\rightpage{first right page text}
\leftpage{second left page text}
...
\end{parallelpages}

```

Notes:

- The `\(left|right)page` declarations are guaranteed to start a new page of the specified kind.
- You are responsible for ensuring that each text (plus any footnotes) is not more than a page long.
- I used braces above so that would be possible to do, say,
`\renewcommand{\rightpage}[1]{}`
to comment out all the texts on the righthand pages.
- However, in general it's probably not a good idea for these macros to take the text as an argument as that would prohibit the use of any verbatim text.
- You could do things like
`\renewcommand{\rightpage}{\cleartooddpage\normalfont\itshape}`
`\renewcommand{\leftpage}{\cleartoevenpage\normalfont\sffamily}`
to have different fonts for the two texts.

I realise that the above does not eliminate the need for hand massaging but it might help in other ways.

Since the above was written I have developed the `ledpar` package [Wil04] as an adjunct to `ledmac` specifically for parallel typesetting of critical texts. This also co-operates with the `babel` package for typesetting in multiple languages. An even more recent extension is the `ledarab` package [Wil05] for handling parallel arabic text in critical editions.

15.5 Notes for EDMAC users

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 changed¹⁹ 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
\let<=\critext
```

¹⁹A name like `\text` is likely to be defined by other LaTeX packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.

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 section 20 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.

16 Implementation overview

We present the `ledmac` code in roughly the order in which it's used during a run of `TEX`. The order is *exactly* that in which it's read when you load the `ledmac` 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.

We begin with the commands you use to start and stop line numbering in a section of text (Section 17). 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 19); 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 20), followed by the macros that take each paragraph apart, attach the line numbers and insertions, and send the result to the vertical list (Section 21). The footnote commands (Section 22) and output routine (Section 23) finish the main part of the processing; cross-referencing (Section 24) and endnotes (Section 25) complete the story.

In what follows, macros with an `@` in their name are more internal to the workings of `ledmac` 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.

17 Preliminaries

I'll try and use `l@d` in macro names to help avoid name clashes, but this is not a hard and fast rule. For example, if an original EDMAC macro includes `edmac` I'll simply change that to `ledmac`.

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

```

1 (*code)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{ledmac}[2012/09/05 v0.18.2 LaTeX port of EDMAC]
4
```

In general I have made the following 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.

I'm adding final/draft options which I hope may be useful.

`\ifledfinal` Use this to remember which option is used, set and execute the options with final as the default.

```
5 \newif\ifledfinal
6 \DeclareOption{final}{\ledfinaltrue}
7 \DeclareOption{draft}{\ledfinalfalse}
8 \ExecuteOptions{final}
```

Use the starred form of `\ProcessOptions` 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.

```
9 \ProcessOptions*\relax
10
```

Loading package *xargs* to declare commands with optional arguments. *Etoolbox* is also used for more clear code, for example in dynamic command names (it could replace `\csname` etc.)

```
11 \RequirePackage{xargs}
12 \RequirePackage{etoolbox}
```

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

```
13 \ifledfinal
14   \newcommand*{\showlemma}[1]{#1}
15 \else
16   \newcommand*{\showlemma}[1]{\underline{#1}}
17 \fi
18
```

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

Initialize it as `\empty`

```
19 \let\linenumberlist=\empty
20
```

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

`\@l@tempcntb` LaTeX already defines `\@tempcnta` and `\@tempcntb` but I have 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).

```
21 \newcount\@l@tempcnta \newcount\@l@tempcntb
```

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

```
22 \newif\ifl@dmemoir
23 \ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
24
```


17.1 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.

```

\ledmac@warning Write a warning message. Changed to use LaTeX capabilities.
25 \newcommand{\ledmac@warning}[1]{\PackageWarning{ledmac}{#1}}

\ledmac@error Write an error message.
26 \newcommand{\ledmac@error}[2]{\PackageError{ledmac}{#1}{#2}}

\led@err@NumberingStarted
\led@err@NumberingNotStarted
\led@err@NumberingShouldHaveStarted
27 \newcommand*{\led@err@NumberingStarted}{%
28   \ledmac@error{Numbering has already been started}{\@ehc}}
29 \newcommand*{\led@err@NumberingNotStarted}{%
30   \ledmac@error{Numbering was not started}{\@ehc}}
31 \newcommand*{\led@err@NumberingShouldHaveStarted}{%
32   \ledmac@error{Numbering should already have been started}{\@ehc}}

\led@mess@NotesChanged
33 \newcommand*{\led@mess@NotesChanged}{%
34   \typeout{ledmac reminder: }%
35   \typeout{ The number of footnotes in this section
36     has changed since the last run.}%
37   \typeout{ You will need to run LaTeX two more times
38     before the footnote placement}%
39   \typeout{ and line numbering in this section are
40     correct.}}

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

\led@err@LineationInNumbered
43 \newcommand*{\led@err@LineationInNumbered}{%
44   \ledmac@error{You can't use \string\lineation\space within
45     a numbered section}{\@ehc}}

\led@warn@BadLineation
\led@warn@BadLinenummargin
\led@warn@BadLockdisp
\led@warn@BadSublockdisp
46 \newcommand*{\led@warn@BadLineation}{%
47   \ledmac@warning{Bad \string\lineation\space argument}}
48 \newcommand*{\led@warn@BadLinenummargin}{%
49   \ledmac@warning{Bad \string\linenummargin\space argument}}
50 \newcommand*{\led@warn@BadLockdisp}{%
51   \ledmac@warning{Bad \string\lockdisp\space argument}}
52 \newcommand*{\led@warn@BadSublockdisp}{%
53   \ledmac@warning{Bad \string\sublockdisp\space argument}}

```

```

\led@warn@NoLineFile
54 \newcommand*{\led@warn@NoLineFile}[1]{%
55   \ledmac@warning{Can't find line-list file #1}}

\led@warn@BadAdvancelineSubline
\led@warn@BadAdvancelineLine 56 \newcommand*{\led@warn@BadAdvancelineSubline}{%
57   \ledmac@warning{\string\advanceline\space produced a sub-line
58     number less than zero.}}
59 \newcommand*{\led@warn@BadAdvancelineLine}{%
60   \ledmac@warning{\string\advanceline\space produced a line
61     number less than zero.}}

\led@warn@BadSetline
\led@warn@BadSetlinenum 62 \newcommand*{\led@warn@BadSetline}{%
63   \ledmac@warning{Bad \string\setline\space argument}}
64 \newcommand*{\led@warn@BadSetlinenum}{%
65   \ledmac@warning{Bad \string\setlinenum\space argument}}

\led@err@PstartNotNumbered
\led@err@PstartInPstart 66 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PendNotNumbered 67   \ledmac@error{\string\pstart\space must be used within a
\led@err@PendNoPstart 68     numbered section}{\@ehc}}
\led@err@AutoparNotNumbered 69 \newcommand*{\led@err@PstartInPstart}{%
70   \ledmac@error{\string\pstart\space encountered while another
71     \string\pstart\space was in effect}{\@ehc}}
72 \newcommand*{\led@err@PendNotNumbered}{%
73   \ledmac@error{\string\pend\space must be used within a
74     numbered section}{\@ehc}}
75 \newcommand*{\led@err@PendNoPstart}{%
76   \ledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
77 \newcommand*{\led@err@AutoparNotNumbered}{%
78   \ledmac@error{\string\autopar\space must be used within a
79     numbered section}{\@ehc}}

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

\led@warn@DuplicateLabel
\led@warn@RefUndefined 82 \newcommand*{\led@warn@DuplicateLabel}[1]{%
83   \ledmac@warning{Duplicate definition of label '#1' on page \the\pageno.}}
84 \newcommand*{\led@warn@RefUndefined}[1]{%
85   \ledmac@warning{Reference '#1' on page \the\pageno\space undefined.
86     Using '000'.}}

\led@warn@NoMarginpars
87 \newcommand*{\led@warn@NoMarginpars}{%
88   \ledmac@warning{You can't use \string\marginpar\space in numbered text}}

```

```

\led@warn@BadSidenotemargin
89 \newcommand*{\led@warn@BadSidenotemargin}{%
90 \ledmac@warning{Bad \string\sidenotemmargin\space argument}}

\led@warn@NoIndexFile
91 \newcommand*{\led@warn@NoIndexFile}[1]{%
92 \ledmac@warning{Undefined index file #1}}

\led@err@TooManyColumns
\led@err@UnequalColumns 93 \newcommand*{\led@err@TooManyColumns}{%
\led@err@LowStartColumn 94 \ledmac@error{Too many columns}{\@ehc}}
\led@err@HighEndColumn 95 \newcommand*{\led@err@UnequalColumns}{%
\led@err@ReverseColumns 96 \ledmac@error{Number of columns is not equal to the number
97 in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
98 \newcommand*{\led@err@LowStartColumn}{%
99 \ledmac@error{Start column is too low}{\@ehc}}
100 \newcommand*{\led@err@HighEndColumn}{%
101 \ledmac@error{End column is too high}{\@ehc}}
102 \newcommand*{\led@err@ReverseColumns}{%
103 \ledmac@error{Start column is greater than end column}{\@ehc}}

```

18 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. LaTeX will maintain and display a ‘section number’ as a count named `\section@num` that counts how many `\beginnumbering` and `\resumenummering` commands have appeared; it needn’t 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.
- ```

104 \newcount\section@num
105 \section@num=0
106 \let\extensionchars=\empty

```
- \ifnumbering** The `\ifnumbering` flag is set to `true` if we’re within a numbered section (that is, between `\beginnumbering` and `\endnumbering`). You can use `\ifnumbering` in your own code to check whether you’re in a numbered section, but don’t change the flag’s value.
- \numberingtrue**
- \numberingfalse**

```

107 \newif\ifnumbering

\ifnumberingR In preparation for the ledpar package, these are related to the ‘left’ text of parallel
\ifl@dpairing texts (when \ifl@dpairing is TRUE). They are explained in the ledpar manual.
\l@dpairingtrue 108 \newif\ifl@dpairing
\l@dpairingfalse 109 \l@dpairingfalse
\ifpst@rtedL 110 \newif\ifpst@rtedL
\pst@rtedLtrue 111 \pst@rtedLfalse
\pst@rtedLfalse 112 \newcount\l@dnumpstartsL
\l@dnumpstartsL 113 \newif\ifledRcol
\ifledRcol The \ifnumberingR flag is set to true if we’re within a right text numbered
section.
114 \newif\ifnumberingR

\beginnumbering \beginnumbering begins a section of numbered text. When it’s executed we
\initnumbering@reg 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’s 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.

115 \newcommand*{\beginnumbering}{%
116 \ifnumbering
117 \led@err@NumberingStarted
118 \endnumbering
119 \fi
120 \global\numberingtrue
121 \global\advance\section@num \@ne
122 \initnumbering@reg
123 \message{Section \the\section@num }%
124 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
125 \l@dend@stuff
126 \setcounter{pstart}{1}
127 }
128 \newcommand*{\initnumbering@reg}{%
129 \global\pst@rtedLfalse
130 \global\l@dnumpstartsL \z@
131 \global\absline@num \z@
132 \global\line@num \z@
133 \global\subline@num \z@
134 \global\@lock \z@
135 \global\sub@lock \z@
136 \global\sublines@false
137 \global\let\next@page@num=\relax

```

```

138 \global\let\sub@change=\relax}
139

```

`\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.

```

140 \def\endnumbering{%
141 \ifnumbering
142 \global\numberingfalse
143 \normal@pars
144 \ifl@dpairing
145 \global\pst@rtedLfalse
146 \else
147 \ifx\insertlines@list\empty\else
148 \global\noteschanged@true
149 \fi
150 \ifx\line@list\empty\else
151 \global\noteschanged@true
152 \fi
153 \fi
154 \ifnoteschanged@
155 \led@mess@NotesChanged
156 \fi
157 \else
158 \led@err@NumberingNotStarted
159 \fi
160 \autoparfalse}

```

`\pausenumbering` The `\pausenumbering` macro is just the same as `\endnumbering`, but with the `\resumenummering` `\ifnumbering` flag set to true, to show that numbering continues across the gap.<sup>20</sup>

```

161 \newcommand{\pausenumbering}{%
162 \endnumbering\global\numberingtrue}

```

The `\resumenummering` 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 `\resumenummering` to ensure that `\pausenumbering` was actually invoked.

```

163 \newcommand*{\resumenummering}{%
164 \ifnumbering
165 \global\pst@rtedLtrue
166 \global\advance\section@num \@ne
167 \led@mess@SectionContinued{\the\section@num}%
168 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
169 \l@dend@stuff
170 \else
171 \led@err@NumberingShouldHaveStarted
172 \endnumbering
173 \beginnumbering

```

---

<sup>20</sup>Our thanks to Wayne Sullivan, who suggested the idea behind these macros.

```

174 \fi}
175

```

## 19 Line counting

### 19.1 Choosing the system of lineation

Sometimes you want line numbers that start at 1 at the top of each page; sometimes you want line numbers that start at 1 at each `\pstart`; other times you want line numbers that start at 1 at the start of each section and increase regardless of page breaks. `ledmac` can do it either way, and you can switch from one to the other within one work. But you have to choose one or the other for all line numbers and line references within each section. Here we will define internal codes for these systems and the macros you use to select them.

```

\ifbypstart@ The \ifbypage@ and \ifbypstart@ flag specifie the current lineation system:
\byppstart@true
\byppstart@false
 \ifbypage@
 \bypage@true
 \bypage@false

```

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

```

 ledmac will use the line-of-section system unless instructed otherwise.
176 \newif\ifbypage@
177 \newif\ifbypstart@

\lineation \lineation{<word>} is the macro you use to select the lineation system. Its
argument is a string: either page or section or pstart.
178 \newcommand*{\lineation}[1]{%
179 \ifnumbering
180 \led@err@LineationInNumbered
181 \else
182 \def\@tempa{#1}\def\@tempb{page}%
183 \ifx\@tempa\@tempb
184 \global\bypage@true
185 \global\bypstart@false
186 \else
187 \def\@tempb{pstart}%
188 \ifx\@tempa\@tempb
189 \global\bypage@false
190 \global\bypstart@true
191 \else
192 \def\@tempb{section}
193 \ifx\@tempa\@tempb
194 \global\bypage@false
195 \global\bypstart@false
196 \else
197 \led@warn@BadLineation
198 \fi
199 \fi

```

```
200 \fi
201 \fi}}
```

`\linenummargin` You call `\linenummargin{<word>}` to specify which margin you want your line numbers in; it takes one argument, a string. You can put the line numbers in the same margin on every page using `left` or `right`; or you can use `inner` or `outer` to get them in the inner or outer margins. (These last two options assume that even-numbered pages will be on the left-hand side of every opening in your book.) You can change this within a numbered section, but the change may not take effect just when you'd like; if it's done between paragraphs nothing surprising should happen.

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

```
202 \newcount\line@margin
203 \newcommand*\linenummargin[1]{%
204 \l@getline@margin{#1}%
205 \ifnum\@l@tempcntb>\m@ne
206 \global\line@margin=\@l@tempcntb
207 \fi}}
208 \newcommand*\l@getline@margin[1]{%
209 \def\@tempa{#1}\def\@tempb{left}%
210 \ifx\@tempa\@tempb
211 \@l@tempcntb \z@
212 \else
213 \def\@tempb{right}%
214 \ifx\@tempa\@tempb
215 \@l@tempcntb \@ne
216 \else
217 \def\@tempb{outer}%
218 \ifx\@tempa\@tempb
219 \@l@tempcntb \tw@
220 \else
221 \def\@tempb{inner}%
222 \ifx\@tempa\@tempb
223 \@l@tempcntb \thr@@
224 \else
225 \led@warn@BadLinenummargin
226 \@l@tempcntb \m@ne
227 \fi
228 \fi
229 \fi
230 \fi}
231
```

`\c@firstlinenum` The following counters tell `ledmac` which lines should be printed with line numbers. `firstlinenum` is the number of the first line in each section that gets a number; `\c@linenumincrement` `linenumincrement` is the difference between successive numbered lines. The initial values of these counters produce labels on lines 5, 10, 15, etc. `linenumincrement` must be at least 1.

```

232 \newcounter{firstlinenum}
233 \setcounter{firstlinenum}{5}
234 \newcounter{linenumincrement}
235 \setcounter{linenumincrement}{5}

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

236 \newcounter{firstsublinenum}
237 \setcounter{firstsublinenum}{5}
238 \newcounter{sublinenumincrement}
239 \setcounter{sublinenumincrement}{5}
240

\firstlinenum These macros can be used to set the corresponding counters.
\linenumincrement 241 \newcommand*{\firstlinenum}[1]{\setcounter{firstlinenum}{#1}}
\firstsublinenum 242 \newcommand*{\linenumincrement}[1]{\setcounter{linenumincrement}{#1}}
\sublinenumincrement 243 \newcommand*{\firstsublinenum}[1]{\setcounter{firstsublinenum}{#1}}
244 \newcommand*{\sublinenumincrement}[1]{\setcounter{sublinenumincrement}{#1}}
245

\lockdisp When line locking is being used, the \lockdisp{<word>} macro specifies whether
\lock@disp a line number—if one is due to appear—should be printed on the first printed line
\l@dgetlock@disp 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.

246 \newcount\lock@disp
247 \newcommand{\lockdisp}[1]{%
248 \l@dgetlock@disp{#1}%
249 \ifnum\l@dtempcntb>\m@ne
250 \global\lock@disp=\l@dtempcntb
251 \else
252 \led@warn@BadLockdisp
253 \fi}}
254 \newcommand*{\l@dgetlock@disp}[1]{
255 \def\@tempa{#1}\def\@tempb{first}%
256 \ifx\@tempa\@tempb
257 \l@dtempcntb \z@
258 \else
259 \def\@tempb{last}%
260 \ifx\@tempa\@tempb
261 \l@dtempcntb \@ne
262 \else
263 \def\@tempb{all}%
264 \ifx\@tempa\@tempb
265 \l@dtempcntb \tw@
266 \else
267 \l@dtempcntb \m@ne
268 \fi
269 \fi

```



```
270 \fi}
271
```

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

```
272 \newcount\sublock@disp
273 \newcommand{\sublockdisp}[1]{%
274 \l@getlock@disp{#1}%
275 \ifnum\@l@tempcntb>\m@ne
276 \global\sublock@disp=\@l@tempcntb
277 \else
278 \led@warn@BadSublockdisp
279 \fi}}
280
```

`\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.

```
281 \newcommand*{\linenumberstyle}[1]{%
282 \def\linenumrep##1{\@nameuse{##1}}
283 \newcommand*{\sublinenumberstyle}[1]{%
284 \def\sublinenumrep##1{\@nameuse{##1}}}
 Initialise the number styles to arabic.
285 \linenumberstyle{arabic}
286 \let\linenumr@p\linenumrep
287 \sublinenumberstyle{arabic}
288 \let\sublinenumr@p\sublinenumrep
289
```

`\leftlinenum` `\leftlinenum` and `\rightlinenum` are the macros that are called to print  
`\rightlinenum` marginal line numbers on a page, for left- and right-hand margins respectively.  
`\linenumsep` They're made easy to access and change, since you may often want to change the  
`\numlabfont` styling in some way. These standard versions illustrate the general sort of thing  
`\ledlinenum` that will be needed; they're 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'll generally want a kern between a line number and the text, and `\linenumsep` is provided as a standard way of storing its size. Line numbers are usually printed in a smaller font, and `\numlabfont` is provided as a standard name for that font. When called, these macros will be executed within a group, so font changes and the like will remain local.

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

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

```

290 \newlength{\linenumsep}
291 \setlength{\linenumsep}{1pc}
292 \newcommand*{\numlabfont}{\normalfont\scriptsize}
293 \newcommand*{\ledlinenum}{%
294 \numlabfont\linenumrep{\line@num}%
295 \ifsublines@
296 \ifnum\subline@num>0\relax
297 \unskip\fullstop\sublinenumrep{\subline@num}%
298 \fi
299 \fi}
300 \newcommand*{\leftlinenum}{%
301 \ledlinenum
302 \kern\linenumsep}
303 \newcommand*{\rightlinenum}{%
304 \kern\linenumsep
305 \ledlinenum}
306

```

## 19.2 List macros

Reminder: compare these with the LaTeX list macros in case they would be suitable instead.

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.

`\list@create` The `\list@create` macro creates a new list. In this version of `ledmac` this macro doesn't do anything beyond initializing an empty list macro, but in future versions it may do more.

```

307 \newcommand*{\list@create}[1]{\global\let#1=\empty}

```

`\list@clear` The `\list@clear` macro just initializes a list to the empty list; in this version of `ledmac` it is no different from `\list@create`.

```

308 \newcommand*{\list@clear}[1]{\global\let#1=\empty}

```

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

```

309 \newtoks\@toksa \newtoks\@toksb
310 \global\@toksa={\}
311 \long\def\xright@appenditem#1\to#2{%

```

```

312 \global\@toksb=\expandafter{#2}%
313 \xdef#2{\the\@toksb\the\@toksa\expandafter{#1}}%
314 \global\@toksb={}}

```

`\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`.

```

315 \long\def\xleft@appenditem#1\to#2{%
316 \global\@toksb=\expandafter{#2}%
317 \xdef#2{\the\@toksa\expandafter{#1}\the\@toksb}%
318 \global\@toksb={}}

```

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

```

319 \def\glp#1\to#2{\expandafter\glpoff#1\glpoff#1#2}
320 \long\def\glpoff\#1#2\glpoff#3#4{\gdef#4{#1}\gdef#3{#2}}
321

```

### 19.3 Line-number counters and lists

Footnote references using line numbers rather than symbols can't be generated in one pass, because we don't 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's used in marginal line numbering and in notes: counting either from the start of the page or from the start of the section, depending on your choice for this section. This may be qualified by `\subline@num`.

```

322 \newcount\line@num

```

`\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.

```

323 \newcount\subline@num

```

|                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\ifsublines@</code>              | We maintain an associated flag, <code>\ifsublines@</code> , to tell us whether we're within a                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>\sublines@true</code>            | sub-line range or not.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\sublines@false</code>           | You may wonder why we don't just use the value of <code>\subline@num</code> 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-lined 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. |
| <code>324 \newif\ifsublines@</code>    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>\absline@num</code>              | The count <code>\absline@num</code> stores the absolute number of lines since the start of the section: that is, the number we've actually printed, no matter what numbers we attached to them. This value is never printed on an output page, though <code>\line@num</code> 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 <code>\line@num</code> is a lot simpler, because it doesn't depend on the lineation system in use.                                                                                                                                           |
| <code>325 \newcount\absline@num</code> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|                                        | We'll be calling <code>\absline@num</code> numbers 'absolute' numbers, and <code>\line@num</code> and <code>\subline@num</code> numbers 'visible' numbers.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\@lock</code>                    | The counts <code>\@lock</code> and <code>\sub@lock</code> tell us the state of line-number and sub-line-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <code>\sub@lock</code>                 | number locking. 0 means we're not within a locked set of lines; 1 means we're at the first line in the set; 2, at some intermediate line; and 3, at the last line.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>326 \newcount\@lock</code>       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>327 \newcount\sub@lock</code>    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>\line@list</code>                | Now we can define the list macros that will be created from the line-list file. We                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>\insertlines@list</code>         | will maintain the following lists:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>\actionlines@list</code>         | • <code>\line@list</code> : the page and line numbers for every lemma marked by                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>\actions@list</code>             | <code>\edtext</code> . There are seven pieces of information, separated by vertical bars:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|                                        | <ol style="list-style-type: none"> <li>1. the starting page,</li> <li>2. line, and</li> <li>3. sub-line numbers, followed by the</li> <li>4. ending page,</li> <li>5. line, and</li> <li>6. sub-line numbers, and then the</li> <li>7. font specifier for the lemma.</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                               |
|                                        | These line numbers are all visible numbers. The font specifier is a set of four codes for font encoding, family, series, and shape, separated by / characters. Thus a lemma that started on page 23, line 35 and went on until page 24, line 3 (with no sub-line numbering), and was typeset in a normal roman font                                                                                                                                                                                                                                                                                                                                                           |

would have a line list entry like this:

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

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

- `\insertlines@list`: the line numbers of lines that have footnotes or other insertions. These are the absolute numbers where the corresponding lemmas begin. This list contains one entry for every footnote in the section; one lemma may contribute no footnotes or many footnotes. This list is used by `\add@inserts` within `\do@line`, to tell it where to insert notes.
- `\actionlines@list`: a list of absolute line numbers at which we are to perform special actions; these actions are specified by the `\actions@list` list defined below.
- `\actions@list`: action codes corresponding to the line numbers in `\actionlines@list`. These codes tell `ledmac` what action it's supposed to take at each of these lines. One action, the page-start action, is generated behind the scenes by `ledmac` 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. `ledmac` 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  $\geq 0$ ) 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 `ledmac` computes the visible line numbers from the absolute line numbers with reference to the other action codes and the settings they invoke; it doesn't require an entry in the action-code list for every line.

Here are the commands to create these lists:

```

328 \list@create{\line@list}
329 \list@create{\insertlines@list}
330 \list@create{\actionlines@list}
331 \list@create{\actions@list}
332
 \page@num We'll need some counts while we read the line-list, for the page number and the
 \endpage@num ending page, line, and sub-line numbers. Some of these will be used again later
 \endline@num on, when we are acting on the data in our list macros.
 \endsubline@num
333 \newcount\page@num
334 \newcount\endpage@num
335 \newcount\endline@num
336 \newcount\endsubline@num

 \ifnoteschanged@ If the number of footnotes in a section is different from what it was during the last
 \noteschanged@true run, or if this is the very first time you've run LaTeX, on this file, the information
 \noteschanged@false 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 don't really know where in the section notes were added or
 removed, and the solution in any case is simply to run LaTeX two more times;

```

there's 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.

```
337 \newif\ifnoteschanged@
```

## 19.4 Reading the line-list file

`\read@linelist` `\read@linelist{<file>}` is the control sequence that's called by `\beginnumbering` (via `\line@list@stuff`) to open and process a line-list file; its argument is the name of the file.

```
338 \newread\@inputcheck
339 \newcommand*{\read@linelist}[1]{%
340 \list@clearing@reg
```

When the file is there we start a new group and make some special definitions we'll need to process it: it's a sequence of `TEX` commands, but they require a few special settings. We make `[` and `]` become grouping characters: they're 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's 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're in horizontal mode they can get interpreted as spaces to be printed.

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

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

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

```
341 \get@linelistfile{#1}%
342 \endgroup
343
```

When the reading is done, we're 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.

```
344 \global\page@num=\m@ne
345 \ifx\actionlines@list\empty
346 \gdef\next@actionline{1000000}%
347 \else
348 \gl@p\actionlines@list\to\next@actionline
349 \gl@p\actions@list\to\next@action
350 \fi}
351
```

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

```
352 \newcommand*{\list@clearing@reg}{%
```

```

353 \list@clear{\line@list}%
354 \list@clear{\insertlines@list}%
355 \list@clear{\actionlines@list}%
356 \list@clear{\actions@list}}

```

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

```

357 \newcommand*{\get@linelistfile}[1]{%
358 \InputIfFileExists{#1}{%
359 \global\noteschanged@false
360 \begingroup
361 \catcode'\[=1 \catcode'\]=2
362 \makeatletter \catcode'\^M=9}{%
363 \led@warn@NoLineFile{#1}%
364 \global\noteschanged@true
365 \begingroup}%
366 }
367

```

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’d have to do some file renaming outside of LaTeX for that to work. We’ve 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 p. 11 above).

## 19.5 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, `\@l`, 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 say `\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`.

```

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

```



In order to get the `\setlinenum` to work I 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 my 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 I added these to the macro. It is now:

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

I don't (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.

```
368 \newcommand{\@l}[2]{%
369 \fix@page{#1}%
370 \@l@reg}
371 \newcommand*{\@l@reg}{%
372 \ifx\l@dchset@num\relax \else
373 \advance\absline@num \@ne
374 \set@line@action
375 \let\l@dchset@num=\relax
376 \advance\absline@num \m@ne
377 \advance\line@num \m@ne
378 \fi
```

Now we are back to the original code.

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

```
379 \advance\absline@num \@ne
380 \ifx\next@page@num\relax \else
381 \page@action
382 \let\next@page@num=\relax
383 \fi
384 \ifx\sub@change\relax \else
385 \ifnum\sub@change>\z@
386 \sublines@true
387 \else
388 \sublines@false
389 \fi
390 \sub@action
391 \let\sub@change=\relax
392 \fi
```

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

```
393 \ifcase\@lock
394 \or
395 \@lock \tw@
396 \or \or
397 \@lock \z@
398 \fi
```

```

399 \ifcase\sub@lock
400 \or
401 \sub@lock \tw@
402 \or \or
403 \sub@lock \z@
404 \fi
 Now advance the visible line number, unless it's been locked.
405 \ifsublines@
406 \ifnum\sub@lock<\tw@
407 \advance\subline@num \@ne
408 \fi
409 \else
410 \ifnum\@lock<\tw@
411 \advance\line@num \@ne \subline@num \z@
412 \fi
413 \fi}
414

```

`\@page` `\@page{<num>}` marks the start of a new output page; its argument is the number of that page.

First we reset the visible line numbers, if we're numbering by page, and store the page number itself in a count.

```

415 \newcommand*{\@page}[1]{%
416 \ifbypage@
417 \line@num \z@ \subline@num \z@
418 \fi
419 \page@num=#1\relax

```

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

```

420 \def\next@page@num{#1}}
421

```

`\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 `\@l`.

```

422 \newcount\last@page@num
423 \last@page@num=-10000
424 \newcommand*{\fix@page}[1]{%
425 \ifnum #1=\last@page@num
426 \else
427 \ifbypage@
428 \line@num=\z@ \subline@num=\z@
429 \fi
430 \page@num=#1\relax
431 \last@page@num=#1\relax
432 \def\next@page@num{#1}%
433 \fi}
434

```

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

```
\@lopR 435 \newcommand*{\@pend}[1]{}
 436 \newcommand*{\@pendR}[1]{}
 437 \newcommand*{\@lopL}[1]{}
 438 \newcommand*{\@lopR}[1]{}
 439
```

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

```
440 \newcommand*{\sub@on}{\ifsublines@
441 \let\sub@change=\relax
442 \else
443 \def\sub@change{1}%
444 \fi}
445 \newcommand*{\sub@off}{\ifsublines@
446 \def\sub@change{-1}%
447 \else
448 \let\sub@change=\relax
449 \fi}
450
```

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

```
451 \newcommand*{\@adv}[1]{\ifsublines@
452 \advance\subline@num by #1\relax
453 \ifnum\subline@num<\z@
454 \led@warn@BadAdvancelineSubline
455 \subline@num \z@
456 \fi
457 \else
458 \advance\line@num by #1\relax
459 \ifnum\line@num<\z@
460 \led@warn@BadAdvancelineLine
461 \line@num \z@
462 \fi
463 \fi
464 \set@line@action}
465
```

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

```
466 \newcommand*{\@set}[1]{\ifsublines@
467 \subline@num=#1\relax
468 \else
469 \line@num=#1\relax
470 \fi
```

```

471 \set@line@action}
472
\l@d@set The \l@d@set{<num>} macro sets the line number for the next \pstart... to the
\l@dchset@num value specified as its argument. This is used to implement \setlinenum.
 \l@dchset@num is a flag to the \@l macro. If it is not \relax then a linenumber
 change is to be done.
473 \newcommand*{\l@d@set}[1]{%
474 \line@num=#1\relax
475 \advance\line@num \@ne
476 \def\l@dchset@num{#1}}
477 \let\l@dchset@num\relax
478

\page@action \page@action adds an entry to the action-code list to change the page number.
479 \newcommand*{\page@action}{%
480 \xright@appenditem{\the\absline@num}\to\actionlines@list
481 \xright@appenditem{\next@page@num}\to\actions@list}

\set@line@action \set@line@action adds an entry to the action-code list to change the visible line
number.
482 \newcommand*{\set@line@action}{%
483 \xright@appenditem{\the\absline@num}\to\actionlines@list
484 \ifsublines@
485 \@l@dttempcnta=-\subline@num
486 \else
487 \@l@dttempcnta=-\line@num
488 \fi
489 \advance\@l@dttempcnta by -5000
490 \xright@appenditem{\the\@l@dttempcnta}\to\actions@list}

\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.
491 \newcommand*{\sub@action}{%
492 \xright@appenditem{\the\absline@num}\to\actionlines@list
493 \ifsublines@
494 \xright@appenditem{-1001}\to\actions@list
495 \else
496 \xright@appenditem{-1002}\to\actions@list
497 \fi}

\lock@on \lock@on adds an entry to the action-code list to turn line number locking on.
\do@lockon The current setting of the sub-lineation flag tells us whether this applies to line
\do@lockonL numbers or sub-line numbers.
 Adding commands to the action list is slow, and it's 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.

```

```

498 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
499 \newcommand*{\do@lockon}{%
500 \ifx\next\lock@off
501 \global\let\lock@off=\skip@lockoff
502 \else
503 \do@lockonL
504 \fi}
505 \newcommand*{\do@lockonL}{%
506 \xright@appenditem{\the\absline@num}\to\actionlines@list
507 \ifsublines@
508 \xright@appenditem{-1005}\to\actions@list
509 \ifnum\sub@lock=z@
510 \sub@lock \@ne
511 \else
512 \ifnum\sub@lock=\thr@@
513 \sub@lock \@ne
514 \fi
515 \fi
516 \else
517 \xright@appenditem{-1003}\to\actions@list
518 \ifnum\@lock=z@
519 \@lock \@ne
520 \else
521 \ifnum\@lock=\thr@@
522 \@lock \@ne
523 \fi
524 \fi
525 \fi}
526

```

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

```

\do@lockoff 527 \newcommand*{\do@lockoffL}{%
\do@lockoffL 528 \xright@appenditem{\the\absline@num}\to\actionlines@list
\skip@lockoff 529 \ifsublines@
530 \xright@appenditem{-1006}\to\actions@list
531 \ifnum\sub@lock=tw@
532 \sub@lock \thr@@
533 \else
534 \sub@lock \z@
535 \fi
536 \else
537 \xright@appenditem{-1004}\to\actions@list
538 \ifnum\@lock=tw@
539 \@lock \thr@@
540 \else
541 \@lock \z@
542 \fi
543 \fi}
544 \newcommand*{\do@lockoff}{\do@lockoffL}
545 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}

```

```
546 \global\let\lock@off=\do@lockoff
547
```

`\n@num` This macro implements the `\skipnumbering` command. It uses a new action code, `\n@num@reg` namely 1007.

```
548 \newcommand*{\n@num}{\n@num@reg}
549 \newcommand*{\n@num@reg}{%
550 \xright@appenditem{\the\absline@num}\to\actionlines@list
551 \xright@appenditem{-1007}\to\actions@list}
552
```

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

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

```
553 \newcount\insert@count
```

- #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's necessary to temporarily redefine `\@ref` within `\@ref`, so that we're only doing one of these at a time.

```
554 \newcommand*{\dummy@ref}[2]{#2}
```

`\@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.

```
555 \newcommand*{\@ref}[2]{%
556 \@ref@reg{#1}{#2}}
557 \newcommand*{\@ref@reg}[2]{%
558 \global\insert@count=#1\relax
559 \loop\ifnum\insert@count>\z@
560 \xright@appenditem{\the\absline@num}\to\insertlines@list
561 \global\advance\insert@count \m@ne
562 \repeat
```

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.

```
563 \begingroup
564 \let\@ref=\dummy@ref
565 \let\page@action=\relax
566 \let\sub@action=\relax
567 \let\set@line@action=\relax
```

```

568 \let\@lab=\relax
569 #2
570 \global\endpage@num=\page@num
571 \global\endline@num=\line@num
572 \global\endsubline@num=\subline@num
573 \endgroup

```

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

```

574 \xright@appenditem%
575 {\the\page@num|\the\line@num|%
576 \ifsublines@ \the\subline@num \else 0\fi}%
577 \the\endpage@num|\the\endline@num|%
578 \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list

```

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

```

579 #2}
580

```

## 19.6 Writing to the line-list file

We've now defined all the counters, lists, and commands involved in reading the line-list file at the start of a section. Now we'll cover the commands that `ledmac` 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`.

```

581 \newwrite\linenum@out

```

`\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'd have to write it at the start of every line. But it's not very easy for the output routine to tell whether an output stream is open or not. There's 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's set to be `true` before any `\linenum@out` file is opened. When such a file is opened for the first time, it's 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`.

```

582 \newif\iffirst@linenum@out@
583 \iffirst@linenum@out@true

```

`\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.

```

584 \newcommand*{\line@list@stuff}[1]{%

```

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

```
585 \read@linelist{#1}%
```

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.

```
586 \iffirst@linenum@out@
587 \immediate\closeout\linenum@out
588 \global\first@linenum@out@false
589 \immediate\openout\linenum@out=#1\relax
590 \else
```

If we get here, then this is not the first line-list we've seen, so we don't open or close the files immediately.

```
591 \closeout\linenum@out
592 \openout\linenum@out=#1\relax
593 \fi}
594
```

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

```
595 \newcommand*{\new@line}{\write\linenum@out{\string\@l[\the\c@page][\thepage]}}
```

`\flag@start` We enclose a lemma marked by `\edtext` in `\flag@start` and `\flag@end`: these  
`\flag@end` 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.

```
596 \newcommand*{\flag@start}{%
597 \edef\next{\write\linenum@out{%
598 \string\@ref[\the\insert@count] []}%
599 \next}
600 \newcommand*{\flag@end}{\write\linenum@out{[]}}
```

`\page@start` Originally the commentary was: `\page@start` writes a command to the line-list file noting the current page number; when used within an output routine, this should be called so as to place its `\write` within the box that gets shipped out, and as close to the top of that box as possible.

However, in October 2004 Alexej Krukov discovered that when processing long paragraphs that included Russian, Greek and Latin texts `ledmac` would go into an infinite loop, emitting thousands of blank pages. This was caused by being unable to find an appropriate place in the output routine. A different algorithm is now used for getting page numbers.

```
601 \newcommand*{\page@start}{%
602
```

`\startsub` `\startsub` and `\endsub` turn sub-lineation on and off, by writing appropriate in-  
`\endsub` structions 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 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 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.

```
603 \newcommand*{\startsub}{\dimen0\lastskip
604 \ifdim\dimen0>0pt \unskip \fi
605 \write\linenum@out{\string\sub@on}%
606 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
607 \def\endsub{\dimen0\lastskip
608 \ifdim\dimen0>0pt \unskip \fi
609 \write\linenum@out{\string\sub@off}%
610 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
611
```

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

```
612 \newcommand*{\advanceline}[1]{\write\linenum@out{\string\@adv[#1]}}
```

`\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.

```
613 \newcommand*{\setline}[1]{%
614 \ifnum#1<\z@
615 \led@warn@BadSetline
616 \else
617 \write\linenum@out{\string\@set[#1]}%
618 \fi}
619
```

`\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.

```
620 \newcommand*{\setlinenum}[1]{%
621 \ifnum#1<\z@
622 \led@warn@BadSetlinenum
623 \else
624 \write\linenum@out{\string\l@d@set[#1]}%
625 \fi}
626
```

`\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.

```

627 \newcommand*{\startlock}{\write\linenum@out{\string\lock@on}}
628 \def\endlock{\write\linenum@out{\string\lock@off}}
629
\ifl@dskipnumber In numbered text \skipnumbering will suspend the numbering for that particular
\l@dskipnumbertrue line.
\l@dskipnumberfalse 630 \newif\ifl@dskipnumber
\skipnumbering 631 \l@dskipnumberfalse
\skipnumbering@reg 632 \newcommand*{\skipnumbering}{\skipnumbering@reg}
633 \newcommand*{\skipnumbering@reg}{%
634 \write\linenum@out{\string\n@num}%
635 \advance\line{-1}}
636

```

## 20 Marking text for notes

The `\edtext` (or `\critext`) 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.

For convenience, I will use `\*text` when I do not need to distinguish between `\edtext` and `\critext`. The `\*text` macros take two arguments, the only difference between `\edtext` and `\critext` is how the second argument is delineated.

`\critext` requires two arguments. At any point within numbered text, you use it by saying:

```
\critext{#1}#2/
```

Similarly `\edtext` requires the same two arguments but you use it by saying:

```
\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. With `\critext` the `/` after **#2** *must* appear: it marks the end of the macro. (*The TeXbook*, p. 204, points out that when additional text to be matched follows the arguments like this, spaces following the macro are not skipped, which is very desirable since this macro will never be used except within text. Having an explicit terminator also helps keep things straight when nested calls to `\critext` are used.) Braces around **#2** are optional with `\critext` and required for `\edtext`.

The `\*text` macro may be used (somewhat) recursively; that is, `\*text` 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's quite likely that we'll 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 aren't 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 `\*text` will fail if you try to use a copy that is called something other than `\*text`. In order to handle recursion, `\*text` needs to redefine its own definition temporarily at one point, and that doesn't work if the macro you are calling is not actually named `\*text`. There's no problem as long as `\*text` is not invoked in the first argument. If you want to call `\*text` something else, it is best to create instead a macro that expands to an invocation of `\*text`, rather than copying `\*text` 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`, p.??). 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 don't provide previous-note information, although it's often wanted; your own macros must handle that. We can't do it correctly without keeping track of what kind of notes have gone past: it's not just a matter of remembering the line numbers associated with the previous invocation of `\*text`, 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.

An example where some 'memory' of line numbers might be required is where there are several variant readings per line of text, and you do not wish the line number to be repeated for each lemma in the notes. After the first occurrence of the line number, you might want the symbol '||' instead of further occurrences, for instance. This can easily be done by a macro like `\printlines`, if it saves the last value of `\l@d@nums` that *it* saw, and then performs a simple conditional test to see whether to print a number or a '||'.

## 20.1 `\edtext` and `\critext` themselves

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 `\critext`'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 `\critext` 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.

```
637 \list@create{\end@lemmas}
```

`\dummy@text` We now need to define a number of macros that allow us to weed out nested instances of `\critext`, 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's because nested `\critext` macros create nested `\@ref` entries in the line-list file.

Here's a macro that takes the same arguments as `\critext` but merely returns the first argument and ignores the second.

```
638 \long\def\dummy@text#1#2/{#1}
```

`\dummy@edtext` LaTeX users are not used to delimited arguments, so I provide a `\edtext` macro as well.

```
639 \newcommand{\dummy@edtext}[2]{#1}
```

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

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

The first class is font-changing macros. We suppress expansion for them by letting them become equal to zero.<sup>21</sup> 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's 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

---

<sup>21</sup>Since 'control sequences equivalent to characters are not expandable'—*The TeXbook*, answer to Exercise 20.14.

alignments— $\text{\TeX}$  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  $\text{\PLAIN\TeX}$  has this sort of problem as well, but isn't 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 `ledmac` macros like `\edlabel` and `\setline` that write things to auxiliary files: that writing should be done only once. And we make `\critext` 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 `ledmac` 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 `\critext` is used.

(A related problem, not addressed by these two macros, is that of characters whose category code is changed by any the macros used in the arguments to `\critext`. 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.)

```

640 \newcommand*{\noexpands}{%
641 \let\select@lemmafnt=0%
642 \let\startsub=\relax \let\endsub=\relax
643 \let\startlock=\relax \let\endlock=\relax
644 \let\edlabel=\@gobble
645 \let\setline=\@gobble \let\advanceline=\@gobble
646 \let\critext=dummy@text
647 \let\edtext=dummy@edtext
648 \l@dtabnoexpands
649 \morenoexpands}
650 \let\morenoexpands=\relax
651
```

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

```

652 \newcommand{\@tag}{}
653 % \end{macrocode}
654 % \end{macro}
655 % \begin{macro}{\critext}
656 % Now we begin \cs{critext} itself. The definition requires a \verb"/" after
```

```

657 % the arguments: this eliminates the possibility of problems about
658 % knowing where \verb"#2" ends. This also changes the handling of spaces
659 % following an invocation of the macro: normally such spaces are
660 % skipped, but in this case they're significant because \verb"#2" is
661 % a 'delimited parameter'. Since \cs{critext} is always used in running
662 % text, it seems more appropriate to pay attention to spaces than to
663 % skip them.
664 %
665 % When executed, \cs{critext} first ensures that we're in
666 % horizontal mode.
667 % \begin{macrocode}
668 \long\def\critext#1#2/{\leavevmode

```

`\csexpandonce@tag` Our normal lemma is just argument #1; but that argument could have further invocations of `\critext` within it. We get a copy of the lemma without any `\critext` macros within it by temporarily redefining `\critext` 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 `\critext` restored; within this group we've also turned off the expansion of those control sequences commonly found within text that can cause trouble for us.

```

669 \begingroup
670 \global\renewcommand{\@tag}{\no@expands #1}%%

```

`\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`.

```

671 \set@line

```

`\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 `\critext`.

```

672 \global\insert@count=0

```

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.

```

673 \ignorespaces #2\relax

```

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

It's 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.

```

674 \flag@start
675 \endgroup
676 \showlemma{#1}%

```

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.

```

677 \ifx\end@lemmas\empty \else
678 \gl@p\end@lemmas\to\x@lemma
679 \x@lemma
680 \global\let\x@lemma=\relax
681 \fi
682 \flag@end}

```

Here's the promised undelimited LaTeX version of `\critext`.

`\edtext`

```

683 \newcommand{\edtext}[2]{\leavevmode
684 \begingroup
685 \global\renewcommand{\@tag}{\no@expands #1}%
686 \set@line
687 \global\insert@count=0
688 \ignorespaces #2\relax
689 \flag@start
690 \endgroup
691 \showlemma{#1}%
692 \ifx\end@lemmas\empty \else
693 \gl@p\end@lemmas\to\x@lemma
694 \x@lemma
695 \global\let\x@lemma=\relax
696 \fi
697 \flag@end}
698

```

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

```

699 \newif\ifnumberline
700 \numberlinetrue

```

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

One instance of `\critext` may generate several notes, or it may generate none—it's legitimate for argument `#2` to `\critext` 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's vital to also remove one and only one `\line@list` entry here.

```

701 \newcommand*{\set@line}{%

```

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

```

702 \ifx\line@list\empty
703 \global\noteschanged@true
704 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
705 \else

```

```

706 \gl@p\line@list\to\@tempb
707 \xdef\l@d@nums{\@tempb|\edfont@info}%
708 \global\let\@tempb=\undefined
709 \fi}
710

```

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

```

711 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
712

```

## 20.2 Substitute lemma

**\lemma** The `\lemma{<text>}` macro allows you to change the lemma that's passed on to the notes.

```

713 \newcommand*{\lemma}[1]{\global\renewcommand{\@tag}{\no@expands #1}}

```

## 20.3 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 p. 52): 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 don't 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.

```

714 \newcommand*{\linenum}[1]{%
715 \xdef\@tempa{#1|||||\\noexpand\\l@d@nums}%
716 \global\let\l@d@nums=\empty
717 \expandafter\line@set\@tempa|\\ignorespaces}

```

**\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.

```

718 \def\line@set#1|#2\\#3|#4\\{%
719 \gdef\@tempb{#1}%
720 \ifx\@tempb\empty
721 \l@d@add{#3}%
722 \else
723 \l@d@add{#1}%
724 \fi
725 \gdef\@tempb{#4}%
726 \ifx\@tempb\empty\else

```



```

727 \l@d@add{|}\line@set#2\\#4\\%
728 \fi}

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

729 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
730

```

## 21 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.

### 21.1 Boxes, counters, \pstart and \pend

```

\raw@text Here are numbers and flags that are used internally in the course of the paragraph
\ifnumberedpar@ decomposition.
\numberedpar@true When we first form the paragraph, it goes into a box register, \raw@text,
\numberedpar@false instead of onto the current vertical list. The \ifnumberedpar@ flag will be true
\num@lines while a paragraph is being processed in that way. \num@lines will store the
\one@line number of lines in the paragraph when it's complete. When we chop it up into
\par@line lines, each line in turn goes into the \one@line register, and \par@line will be
the number of that line within the paragraph.

731 \newbox\raw@text
732 \newif\ifnumberedpar@
733 \newcount\num@lines
734 \newbox\one@line
735 \newcount\par@line

\pstart \pstart starts the paragraph by clearing the \inserts@list list and other rele-
\numberpstarttrue vant variables, and then arranges for the subsequent text to go into the \raw@text
\numberpstartfalse box. \pstart needs to appear at the start of every paragraph that's to be num-
thepstart bered; the \autopar command below may be used to insert these commands
automatically.

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.

You can use the command \numberpstarttrue to insert a number on every
\pstart. To stop the numbering, you must use \numberpstartfalse. To reset
the numebering of \pstarts, insert

\setcounter{pstart}{0}

```

```

736
737 \newcounter{pstart}
738 \renewcommand{\thepstart}{\bfseries\@arabic{c@pstart}. }
739 \newif\ifnumberpstart
740 \numberpstartfalse
741 \newcommand*{\pstart}{
742 \if@nobreak
743 \let\@oldnobreak\@nobreaktrue
744 \else
745 \let\@oldnobreak\@nobreakfalse
746 \fi
747 \@nobreaktrue
748 \ifnumbering \else
749 \led@err@PstartNotNumbered
750 \beginnumbering
751 \fi
752 \ifnumberedpar@
753 \led@err@PstartInPstart
754 \pend
755 \fi
756 \list@clear{\inserts@list}%
757 \global\let\next@insert=\empty
758 \begingroup\normal@pars
759 \global\setbox\raw@text=\vbox\bgroup\ifautopar\else\ifnumberpstart\ifinstanza\else\ifsi
760 \numberedpar@true}

```

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

```

761 \newcommand*{\pend}{\ifnumbering \else
762 \led@err@PendNotNumbered
763 \fi
764 \ifnumberedpar@ \else
765 \led@err@PendNoPstart
766 \fi

```

We set all the usual interline penalties to zero and then immediately call `\endgraf` to end the paragraph; this ensures that there'll 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 aren't any more lines left.

```

767 \l@dzeroopenalties
768 \endgraf\global\num@lines=\prevgraf\egroup
769 \global\par@line=0

```

We check if lineation is by `pstart` : in this case, we reset line number, but only in the second line of the `pstart`, to prevent some trouble. We can't reset line number at the beginning of `\pstart` `\setline` is parsed at the end of previous `\pend`, and so, we must do it at the end of first line of `pstart`.

```

770 \csnumdef{pstartline}{0}

```

```

771 \loop\ifvbox\raw@text
772 \cnumdef{pstartline}{\pstartline+1}%
773 \do@line
774 \ifbypstart@%
775 \ifnumequal{pstartline}{1}{\setline{1}}{}%
776 \fi
777 \repeat

```

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

```

778 \flush@notes
779 \endgroup
780 \ignorespaces
781 \ifnumberpstart
782 \pstartnumtrue
783 \fi
784 \@oldnobreak
785 \addtocounter{pstart}{1}}
786

```

`\l@dzzeropenalties` A macro to zero penalties for `\pend`.

```

787 \newcommand*{\l@dzzeropenalties}{%
788 \brokenpenalty \z@ \clubpenalty \z@
789 \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
790 \postdisplaypenalty \z@ \widowpenalty \z@}
791

```

`\autopar` In most cases it's 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 don't 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's been added for this paragraph.

Then we start again with `\pstart`, restoring the indentation that we saved, and locally change `\par` so that it'll do our `\pend` for us.

```

792 \newif\ifautopar
793 \autoparfalse
794 \newcommand*{\autopar}{
795 \ifledRcol
796 \ifnumberingR \else
797 \led@err@AutoparNotNumbered
798 \beginnumberingR
799 \fi
800 \else
801 \ifnumbering \else
802 \led@err@AutoparNotNumbered
803 \beginnumbering
804 \fi
805 \fi
806 \autopartrue
807 \everypar={\setbox0=\lastbox
808 \endgraf \vskip-\parskip
809 \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\fi\fi
810 \let\par=\pend}%
811 \ignorespaces}

```

`\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'll want to do this within footnotes, for example.

```

812 \newcommand*{\normal@pars}{\everypar={}\let\par\endgraf}
813

```

## 21.2 Processing one line

`\do@line` The `\do@line` macro is called by `\pend` to do all the processing for a single line of text.

```

\l@dunhbox@line
814 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
815 \newcommand*{\do@line}{%
816 {\vbadness=10000
817 \splittopskip=z@
818 \do@linehook
819 \l@emptyd@ta
820 \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
821 \unvbox\one@line \global\setbox\one@line=\lastbox
822 \getline@num
823 \ifnum\@lock>\@ne
824 \inserthangingsymboltrue
825 \else
826 \inserthangingsymbolfalse
827 \fi
828 \affixline@num
829 \affixpstart@num

```

```

830 \hb@xt@ \linewidth{\inserthangingsymbol\l@dld@ta\add@inserts\affixside@note
831 \l@dlsn@te
832 {\ledllfill\hb@xt@ \wd\one@line{\new@line\l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta%
833 \l@drsn@te
834 }}}}

```

`\do@linehook` A hook into `\do@line`.

```
835 \newcommand*{\do@linehook}{}
```

`\l@emptyd@ta` Nulls the `\...d@ta`, which may later hold line numbers. Similarly for `\l@dcsnotetext`  
`\l@dld@ta` for the text of a sidenote.

```
\l@drd@ta 836 \newcommand*{\l@emptyd@ta}{%
```

```
\l@dcsnotetext 837 \gdef\l@dld@ta{}%
```

```
838 \gdef\l@drd@ta{}%
```

```
839 \gdef\l@dcsnotetext{}}
```

```
840
```

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

```
\l@drsn@te 841 \newcommand{\l@dlsn@te}{%
```

```
842 \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep}}
```

```
843 \newcommand{\l@drsn@te}{%
```

```
844 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
```

```
845
```

`\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`.

```
846 \newcommand*{\ledllfill}{\hfil}
```

```
847 \newcommand*{\ledrlfill}{}
```

```
848
```

## 21.3 Line and page number computation

`\getline@num` The `\getline@num` macro determines the page and line numbers for the line we're about to send to the vertical list.

```
849 \newcommand*{\getline@num}{%
```

```
850 \ifnumberline
```

```
851 \global\advance\absline@num \@ne
```

```
852 \fi
```

```
853 \do@actions
```

```
854 \do@ballast
```

```
855 \ifnumberline
```

```
856 \ifsublines@
```

```
857 \ifnum\sub@lock<\tw@
```

```
858 \global\advance\subline@num \@ne
```

```
859 \fi
```

```
860 \else
```

```
861 \ifnum\@lock<\tw@
```

```
862 \global\advance\line@num \@ne
```

```

863 \global\subline@num \z@
864 \fi
865 \fi
866 \fi
867 }

```

`\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,  $\text{\TeX}$  will be given extra encouragement to break the page here (see p. 86).

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

```

868 \newcount\ballast@count
869 \newcounter{ballast}
870 \setcounter{ballast}{0}

```

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.

```

871 \newcommand*{\do@ballast}{\global\ballast@count \z@
872 \begingroup
873 \advance\absline@num \@ne
874 \ifnum\next@actionline=\absline@num
875 \ifnum\next@action>-1001\relax
876 \global\advance\ballast@count by -\c@ballast
877 \fi
878 \fi
879 \endgroup}

```

`\do@actions` The `\do@actions` macro looks at the list of actions to take at particular absolute  
`\do@actions@next` line numbers, and does everything that's specified for the current line.

It may call itself recursively, and to do this efficiently (using  $\text{\TeX}$ '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's just `\relax`.

```

880 \newcommand*{\do@actions}{%
881 \global\let\do@actions@next=\relax
882 \ifnum\absline@num<\next@actionline\else

```

First, page number changes, which will generally be the most common actions. If we're restarting lineation on each page, this is where it happens.

```

883 \ifnum\next@action>-1001
884 \global\page@num=\next@action
885 \ifbypage@
886 \global\line@num=\z@ \global\subline@num=\z@

```

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`.)

It's one of the fixed codes. We rescale the value in `@l@dttempcnta` so that we can use a case statement.

Now we get information about the next action off the list, and then set `\do@actions@next` so that we'll call ourself recursively: the next action might also be for this line.

There's no warning if we find `\actionlines@list` empty, since that will always happen near the end of the section.

Make the recursive call, if necessary.

This macro handles the fixed codes for `\do@actions`. It is one big case statement.

[illegible]

```

919 \or% % 1003
920 \global\@lock=\@ne
921 \or% % 1004
922 \ifnum\@lock=\tw@
923 \global\@lock=\thr@@
924 \else
925 \global\@lock=\z@
926 \fi
927 \or% % 1005
928 \global\sub@lock=\@ne
929 \or% % 1006
930 \ifnum\sub@lock=\tw@
931 \global\sub@lock=\thr@@
932 \else
933 \global\sub@lock=\z@
934 \fi
935 \or% % 1007
936 \l@dskipnumbertrue
937 \else
938 \led@warn@BadAction
939 \fi}
940
941

```

## 21.4 Line number printing

`\affixline@num` `\affixline@num` originally took a single argument, a series of commands for printing the line just split off by `\do@line`; it put that line back on the vertical list, and added a line number if necessary. It now just puts a left line number into `\l@dld@ta` or a right line number into `\l@drd@ta` if required.

To determine whether we need to affix a line number to this line, we compute the following:

$$\begin{aligned}
 n &= \text{int}((\text{linenum} - \text{firstlinenum}) / \text{linenumincrement}) \\
 m &= \text{firstlinenum} + (n \times \text{linenumincrement})
 \end{aligned}$$

(where *int* truncates a real number to an integer). *m* will be equal to *linenum* only if we're 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@dtmpcnta`, 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@dtmpcntb` for comparison.

Remember that some counts are now counters!

First, the case when we're within a sub-line range.

```

942 \newcommand*{\affixline@num}{%

```



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).

```

943 \ifnumberline
944 \ifl@dskipnumber
945 \global\l@dskipnumberfalse
946 \else
947 \ifsublines@
948 \@l@tempcntb=\subline@num
949 \ifnum\subline@num>\c@firstsublinenum
950 \@l@tempcnta=\subline@num
951 \advance\@l@tempcnta by-\c@firstsublinenum
952 \divide\@l@tempcnta by\c@sublinenumincrement
953 \multiply\@l@tempcnta by\c@sublinenumincrement
954 \advance\@l@tempcnta by\c@firstsublinenum
955 \else
956 \@l@tempcnta=\c@firstsublinenum
957 \fi

```

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.

```

958 \ch@cksub@l@ck

```

Now the line number case, which works the same way.

```

959 \else
960 \@l@tempcntb=\line@num

```

Check on the `\linenumberlist` If it's `\empty` use the standard algorithm.

```

961 \ifx\linenumberlist\empty
962 \ifnum\line@num>\c@firstlinenum
963 \@l@tempcnta=\line@num
964 \advance\@l@tempcnta by-\c@firstlinenum
965 \divide\@l@tempcnta by\c@linenumincrement
966 \multiply\@l@tempcnta by\c@linenumincrement
967 \advance\@l@tempcnta by\c@firstlinenum
968 \else
969 \@l@tempcnta=\c@firstlinenum
970 \fi
971 \else

```

The `\linenumberlist` wasn't `\empty`, so here's Wayne's numbering mechanism. This takes place in TeX's mouth.

```

972 \@l@tempcnta=\line@num
973 \edef\rem@inder{\linenumberlist,\number\line@num,}%
974 \edef\sc@n@list{\def\noexpand\sc@n@list
975 ###1,\number\@l@tempcnta,###2|{\def\noexpand\rem@inder{###2}}}%
976 \sc@n@list\expandafter\sc@n@list\rem@inder|%
977 \ifx\rem@inder\empty\advance\@l@tempcnta\@ne\fi
978 \fi

```

A locking check for lines, just like the version for sub-line numbers above.

```
979 \ch@ck@l@ck
980 \fi
```

The following test is true if we need to print a line number.

```
981 \ifnum\@l@tempcnta=\@l@tempcntb
```

If we got here, we're 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's 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's 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 I think 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 982 \if@twocolumn
983 \if@firstcolumn
984 \gdef\l@dld@ta{\llap{\leftlinenum}}%
985 \else
986 \gdef\l@drd@ta{\rlap{\rightlinenum}}%
987 \fi
988 \else
```

Continuing the original code ...

```
989 \@l@tempcntb=\line@margin
990 \ifnum\@l@tempcntb>\@ne
991 \advance\@l@tempcntb \page@num
992 \fi
```

Now print the line (#1) with its page number.

```
993 \ifodd\@l@tempcntb
994 \gdef\l@drd@ta{\rlap{\rightlinenum}}%
995 \else
996 \gdef\l@dld@ta{\llap{\leftlinenum}}%
997 \fi
998 \fi
999 \else
```

As no line number is to be appended, we just print the line as is.

```
1000 %% #1%
1001 \fi
```

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

```
1002 \f@x@l@cks
```

```

1003 \fi
1004 \fi
1005 }
1006

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck` checks subline locking. If it fails, then we disable the line-number display by setting the counters to arbitrary but unequal values.

```

1007 \newcommand*{\ch@cksub@l@ck}{%
1008 \ifcase\sub@lock
1009 \or
1010 \ifnum\sublock@disp=\@ne
1011 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1012 \fi
1013 \or
1014 \ifnum\sublock@disp=\tw@ \else
1015 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1016 \fi
1017 \or
1018 \ifnum\sublock@disp=\z@
1019 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1020 \fi
1021 \fi}

```

Similarly for line numbers.

```

1022 \newcommand*{\ch@ck@l@ck}{%
1023 \ifcase\@lock
1024 \or
1025 \ifnum\lock@disp=\@ne
1026 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1027 \fi
1028 \or
1029 \ifnum\lock@disp=\tw@ \else
1030 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1031 \fi
1032 \or
1033 \ifnum\lock@disp=\z@
1034 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1035 \fi
1036 \fi}

```

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

```

1037 \newcommand*{\f@x@l@cks}{%
1038 \ifcase\@lock
1039 \or
1040 \global\@lock=\tw@
1041 \or \or
1042 \global\@lock=\z@
1043 \fi

```

```

1044 \ifcase\sub@lock
1045 \or
1046 \global\sub@lock=\tw@
1047 \or \or
1048 \global\sub@lock=\z@
1049 \fi}
1050

```

`\pageparbreak` Because of TeX's asynchronous page breaking mechanism we can never be sure juust where it will make a break and, naturally, it has already decided exactly how it will typeset any remainder of a paragraph that crosses the break. This is disconcerting when trying to number lines by the page or put line numbers in different margins. This macro tries to force an invisible paragraph break and a page break.

```

1051 \newcommand{\pageparbreak}{\pend\newpage\pstart\noindent}
1052

```

## 21.5 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 begining of a pstart, and not in every line, a boolean test is made. The `\pstartnum` boolean is set to TRUE at every `\pend`. It's tried in the `\leftpstartnum` and `\rightpstartnum` commands. After the try, it is set to FALSE.

```

\leftpstartnum
\rightpstartnum 1053
\ifsidepstartnum 1054 \newif\ifsidepstartnum
1055 \newcommand*{\affixpstart@num}{%
1056 \ifsidepstartnum
1057 \if@twocolumn
1058 \if@firstcolumn
1059 \gdef\l@dld@ta{\llap{{\leftpstartnum}}}%
1060 \else
1061 \gdef\l@drd@ta{\rlap{{\rightpstartnum}}}%
1062 \fi
1063 \else
1064 \l@dtmpcntb=\line@margin
1065 \ifnum\l@dtmpcntb>\@ne
1066 \advance\l@dtmpcntb \page@num
1067 \fi
1068 \ifodd\l@dtmpcntb
1069 \gdef\l@drd@ta{\rlap{{\rightpstartnum}}}%
1070 \else

```

```

1071 \gdef\l@dld@ta{\llap{{\leftpstartnum}}}%
1072 \fi
1073 \fi
1074 \fi
1075
1076 }
1077 %
1078
1079 \newif\ifpstartnum
1080 \pstartnumtrue
1081 \newcommand*{\leftpstartnum}{
1082 \ifpstartnum\thepstart
1083 \kern\linenumsep\fi
1084 \global\pstartnumfalse
1085 }
1086 \newcommand*{\rightpstartnum}{
1087 \ifpstartnum
1088 \kern\linenumsep
1089 \thepstart
1090 \fi
1091 \global\pstartnumfalse
1092 }

```

## 21.6 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.

```

1093 \list@create{\inserts@list}

```

`\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 T<sub>E</sub>X'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's just `\relax`.

```

1094 \newcommand*{\add@inserts}{%
1095 \global\let\add@inserts@next=\relax

```

If `\inserts@list` is empty, there aren't any more notes or insertions for this paragraph, and we needn't waste our time.

```

1096 \ifx\inserts@list\empty \else

```

The `\next@insert` macro records the number of the line that receives the next footnote or other insert; it's empty when we start out, and just after we've affixed a note or insert.

```

1097 \ifx\next@insert\empty
1098 \ifx\insertlines@list\empty
1099 \global\noteschanged@true

```

```

1100 \gdef\next@insert{100000}%
1101 \else
1102 \gl@p\insertlines@list\to\next@insert
1103 \fi
1104 \fi

```

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'll call ourself recursively: there might be another insert for this same line.

```

1105 \ifnum\next@insert=\absline@num
1106 \gl@p\inserts@list\to\@insert
1107 \@insert
1108 \global\let\@insert=\undefined
1109 \global\let\next@insert=\empty
1110 \global\let\add@inserts@next=\add@inserts
1111 \fi
1112 \fi

```

Make the recursive call, if necessary.

```

1113 \add@inserts@next}
1114

```

## 21.7 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're 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 (p.78). Finally, the penalty is checked to see that it doesn't go below  $-10000$ .

```

1115 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
1116 \ifnum\num@lines>\@ne
1117 \global\advance\par@line \@ne
1118 \ifnum\par@line=\@ne
1119 \advance\@l@tempcnta \clubpenalty
1120 \fi
1121 \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
1122 \ifnum\@l@tempcntb=\num@lines
1123 \advance\@l@tempcnta \widowpenalty
1124 \fi
1125 \ifnum\par@line<\num@lines
1126 \advance\@l@tempcnta \interlinepenalty
1127 \fi

```

```

1128 \fi
1129 \ifnum\@l@dttempcnta=\z@
1130 \relax
1131 \else
1132 \ifnum\@l@dttempcnta>-10000
1133 \penalty\@l@dttempcnta
1134 \else
1135 \penalty -10000
1136 \fi
1137 \fi}
1138

```

## 21.8 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 last run of **T<sub>E</sub>X**, then there can be leftover notes that haven't yet been printed. An appropriate error message will be printed elsewhere; but it's best to go ahead and print these notes somewhere, even if it's 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's not too far from the proper location, to which they'll move on the next run.

```

1139 \newcommand*{\flush@notes}{%
1140 \xloop
1141 \ifx\inserts@list\empty \else
1142 \gl@p\inserts@list\to\@insert
1143 \@insert
1144 \global\let\@insert=\undefined
1145 \repeat}
1146

```

**\xloop** **\xloop** is a variant of the PLAIN **T<sub>E</sub>X** **\loop** macro, useful when it's hard to construct a positive test using the **T<sub>E</sub>X** **\if** commands—as in **\flush@notes** above. One says **\xloop ... \if ... \else ... \repeat**, and the action following **\else** is repeated as long as the **\if** test fails. (This macro will work wherever the PLAIN **T<sub>E</sub>X** **\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.

```

1147 \def\xloop#1\repeat{%
1148 \def\body{#1\expandafter\body\fi}%
1149 \body}
1150

```

## 22 Footnotes

The footnote macros are adapted from those in PLAIN T<sub>E</sub>X, 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 five separate levels of footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

### 22.1 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.

I have deleted all Plain Font-related code and just kept the code for NFSS font handling.

`\notefontsetup` The font setup defined in `\notefontsetup` defines the standard fonts for the text of the footnotes. Parts of the footnote, such as the line number references and the lemma, are enclosed in groups, with their own font macros, so a note in plain roman can still have line numbers in bold, say, and the lemma in the same font encoding, family, series, and shape of font as in the main text. Typically this definition should specify only a size.

The original font for `\notefontsetup` effectively maps to LaTeX `\footnotesize` for a 10pt document.

```
1151 \newcommand*{\notefontsetup}{\footnotesize}
```

`\notenumfont` The line numbers will be printed using the font selected by executing `\notenumfont`.

The original font for `\notenumfont` maps to LaTeX `\scriptsize` for a 10pt document. However, the description in the user guide does not seem to match the definition (the usage guide says that the size is `\notefontsetup`).

```
1152 \newcommand*{\notenumfont}{\normalfont}
```

`\select@lemmafонт` `\select@lemmafонт` is provided to set the right font for the lemma in a note.

`\select@@lemmafонт` 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.

```
1153 \def\select@lemmafонт#1|#2|#3|#4|#5|#6|#7|{\select@@lemmafонт#7|}
1154 \def\select@@lemmafонт#1/#2/#3/#4|{%
1155 {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
1156 \selectfont}
1157
```



## 22.2 Outer-level footnote commands

`\footnoteoptions@` The `\footnoteoption@[<side>]{<options>}{<value>}` change the value of on options of Xfootnote, to switch between true and false.

```

1158 \newcommand*{\footnoteoptions@}[3][1=L,usedefault]{%
1159 \renewcommand{\do}[1]{%
1160 \ifstrequal{#1}{L}{% In Leftside
1161 \xright@appenditem{\global\noexpand\settoggle{##1@}{#3}}\to\inserts@list% Switch toogle, in
1162 \global\advance\insert@count \@ne% Increment the left insert counter.
1163 }%
1164 {%
1165 \xright@appenditem{\global\noexpand\settoggle{##1@}{#3}}\to\inserts@listR% Switch toogle, in
1166 \global\advance\insert@countR \@ne% Increment the right insert counter insert.
1167 }%
1168 }%
1169 \notblank{#2}{\docsvlist{#2}}{}}% Parsing all options
1170 }
```

`\Afootnote` The outer-level footnote commands will look familiar: they're just called `\Afootnote`, `\Bfootnote`, etc., instead of plain `\footnote`. What they do, however, is quite different, since they have to operate in conjunction with `\critext` when numbering is in effect.

If we're within a line-numbered paragraph, then, we tack this note onto the `\inserts@list` list, and increment the deferred-page-bottom-note counter.

```

1171 \newcommand*{\Afootnote}[2][]{%
1172 \newcommand{\content}{#2}%
1173 \footnoteoptions@{#1}{true}%
1174 \ifnumberedpar@
1175 \xright@appenditem{\noexpand\vAfootnote{A}%
1176 {{\l@ednums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}\to\inserts@list
1177 \global\advance\insert@count \@ne
```

Within free text, there's no need to put off making the insertion for this note. No line numbers are available, so this isn't generally that useful; but you might want to use it to get around some limitation of `ledmac`.

```

1178 \else
1179 \vAfootnote{A}{{0|0|0|0|0|0|0|0}}{\csexpandonce{content}}}%
1180 \fi%
1181 \footnoteoptions@{#1}{false}%
1182 \ignorespaces}
```

`\Bfootnote` We need similar commands for the other footnote series.

```

\Bfootnote 1183 \newcommand*{\Bfootnote}[2][]{%
\Bfootnote 1184 \newcommand{\content}{#2}%
\Bfootnote 1185 \footnoteoptions@{#1}{true}%
\Bfootnote 1186 \ifnumberedpar@
\Bfootnote 1187 \xright@appenditem{\noexpand\vBfootnote{B}%
\Bfootnote 1188 {{\l@ednums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}\to\inserts@list
\Bfootnote 1189 \global\advance\insert@count \@ne%
```

`\mpAfootins` For footnotes in minipages and the like, we need a new set of inserts.  
`\mpBfootins`  
`\mpCfootins`  
`\mpDfootins`  
`\mpEfootins`

```

1236 \newinsert\mpAfootins
1237 \newinsert\mpBfootins
1238 \newinsert\mpCfootins
1239 \newinsert\mpDfootins
1240 \newinsert\mpEfootins
1241

```

`\mpAfootnote` For footnotes in minipages and the like, we need a similar series of commands.

```

\mpBfootnote 1242 \newcommand*\mpAfootnote}[2][]{%
\mpCfootnote 1243 \newcommand{\content}{#2}%
\mpDfootnote 1244 \footnoteoptions@{#1}{true}%
\mpEfootnote 1245 \ifnumberedpar@
1246 \xright@appenditem{\noexpand\mpvAfootnote{A}%
1247 {\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}\to\inserts@list
1248 \global\advance\insert@count \one%
1249 \footnoteoptions@{#1}{false}%
1250 \else
1251 \mpvAfootnote{A}{0|0|0|0|0|0|0|0}{#1}%
1252 \fi%
1253 \footnoteoptions@{#1}{false}%
1254 \ignorespaces}

1255 \newcommand*\mpBfootnote}[2][]{%
1256 \newcommand{\content}{#2}%
1257 \footnoteoptions@{#1}{true}%
1258 \ifnumberedpar@
1259 \xright@appenditem{\noexpand\mpvBfootnote{B}%
1260 {\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}\to\inserts@list
1261 \global\advance\insert@count \one%
1262 \footnoteoptions@{#1}{false}%
1263 \else
1264 \mpvBfootnote{B}{0|0|0|0|0|0|0|0}{#1}%
1265 \fi%
1266 \footnoteoptions@{#1}{false}%
1267 \ignorespaces}

1268 \newcommand*\mpCfootnote}[2][]{%
1269 \newcommand{\content}{#2}%
1270 \footnoteoptions@{#1}{true}%
1271 \ifnumberedpar@
1272 \xright@appenditem{\noexpand\mpvCfootnote{C}%
1273 {\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}\to\inserts@list
1274 \global\advance\insert@count \one%
1275 \footnoteoptions@{#1}{false}%
1276 \else
1277 \mpvCfootnote{C}{0|0|0|0|0|0|0|0}{#1}%
1278 \fi%
1279 \footnoteoptions@{#1}{false}%
1280 \ignorespaces}

1281 \newcommand*\mpDfootnote}[2][]{%

```

```

1282 \newcommand{\content}{#2}%
1283 \footnoteoptions@{#1}{true}%
1284 \ifnumberedpar@
1285 \xright@appenditem{\noexpand\mpvDfootnote{D}%
1286 {\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}\to\inserts
1287 \global\advance\insert@count \@ne%
1288 \footnoteoptions@{#1}{false}%
1289 \else
1290 \mpvDfootnote{D}{\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}%
1291 \fi%
1292 \footnoteoptions@{#1}{false}%
1293 \ignorespaces}

1294 \newcommand*{\mpEfootnote}[2][]{%
1295 \newcommand{\content}{#2}%
1296 \footnoteoptions@{#1}{true}%
1297 \ifnumberedpar@
1298 \xright@appenditem{\noexpand\mpvEfootnote{E}%
1299 {\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}\to\inserts
1300 \global\advance\insert@count \@ne%
1301 \footnoteoptions@{#1}{false}%
1302 \else
1303 \mpvEfootnote{E}{\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}%
1304 \fi%
1305 \footnoteoptions@{#1}{false}%
1306 \ignorespaces}

```

## 22.3 Normal footnote formatting

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 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 footnotes we’re 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`.

`\normalvfootnote` We now begin a series of commands that do ‘normal’ footnote formatting: a format much like that implemented in PLAIN  $\text{\TeX}$ , 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.

```

1307 \newcommand*{\normalvfootnote}[2]{%
1308 \insert\csname #1footins\endcsname\bgroup
1309 \notefontsetup
1310 \footssplitskips
1311 \spaceskip=\z@skip \xspaceskip=\z@skip
1312 \csname #1footfmt\endcsname #2[#1]\egroup}

```

`\footssplitskips` Some setup code that is common for a variety of footnotes.

```

1313 \newcommand*{\footssplitskips}{%
1314 \interlinepenalty=\interfootnotelinepenalty
1315 \floatingpenalty=\@MM
1316 \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
1317 \leftskip=\z@skip \rightskip=\z@skip}
1318

```

`\mpnormalvfootnote` And a somewhat different version for minipages.

```

1319 \newcommand*{\mpnormalvfootnote}[2]{%
1320 \global\setbox\@nameuse{mp#1footins}\vbox{%
1321 \unvbox\@nameuse{mp#1footins}
1322 \notefontsetup
1323 \hsize\columnwidth
1324 \@parboxrestore
1325 \color@begingroup
1326 \csname #1footfmt\endcsname #2[#1]\color@endgroup}}
1327

```

`\ledsetnormalparstuff` `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see p.52), 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; it’s intended to be copied and modified as necessary.

`\par` should always be redefined to `\endgraf` within the format macro (this is what `\normal@pars` does), to override any tricky stuff which might be done in the main text to get the lines numbered automatically (as set up by `\autopar`, for example).

```

1328 \newcommand*{\ledsetnormalparstuff}{%
1329 \normal@pars
1330 \parindent \z@ \parfillskip \z@ \@plus 1fil}
1331 \newcommandx*{\normalfootfmt}[4][4=Z]{% 4th arg is optional, for ascendant compatibility
1332 \ledsetnormalparstuff%
1333 \strut{\printlinefootnote{#1}{#4}}%
1334 {\select@lemmafnt#1|#2}%
1335 \iftoggle{nosep@}{\hskip\csuse{inplaceoflemmaseparator@#4}}{\ifcsemt{lemmaseparator@#4}%
1336 {\hskip\csuse{inplaceoflemmaseparator@#4}}%
1337 {\nobreak\hskip\csuse{beforelemmaseparator@#4}\csuse{lemmaseparator@#4}\hskip\csuse{afterlemmasep
1338 }}%
1339 #3\strut\par}

```

`\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 footnotes.

```
1340 \def\endashchar{\textnormal{--}}
1341 \newcommand*{\fullstop}{\textnormal{.}}
1342 \newcommand*{\rbracket}{\textnormal{\thinspace]]}}
1343
```

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 page 52: 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.

The original EDMAC code used several counters at this point, saying:

To simplify the logic, we use a lot of counters to tell us which numbers need to get printed (using 1 for yes, 0 for no, so that `\ifodd` tests for ‘yes’). The counter assignments are:

- `\@pnum` for page numbers;
- `\@ssub` for starting sub-line;
- `\@elin` for ending line;
- `\@esl` for ending sub-line; and
- `\@dash` for the dash between the starting and ending groups.

There’s no counter for the line number because it’s always printed.

LaTeX tends to use a lot of counters and packages should try and minimise the number of new ones they create. In line with this I have reverted to traditional booleans.

```
\ifl@d@pnum
\ifl@d@ssub 1344 \newif\ifl@d@pnum
\ifl@d@elin 1345 \l@d@pnumfalse
\ifl@d@esl 1346 \newif\ifl@d@ssub
\ifl@d@dash 1347 \l@d@ssubfalse
1348 \newif\ifl@d@elin
1349 \l@d@elinfalse
1350 \newif\ifl@d@esl
1351 \l@d@eslfalse
1352 \newif\ifl@d@dash
1353 \l@d@dashfalse
```

```

\l@dparsfootspec \l@dparsfootspec{<spec>}{<lemma>}{<text>} parses a footnote specification.
\l@dp@rsefootspec <lemma> and <text> are the lemma and text respectively. <spec> is the line and
\l@dparsedstartpage page number and lemma font specifier in \l@d@nums style format. The real work
\l@dparsedstartline is done by \l@dp@rsefootspec which defines macros holding the numeric values.
\l@dparsedstartsub 1354 \newcommand*{\l@dparsfootspec}[3]{\l@dp@rsefootspec#1|}
\l@dparsedendpage 1355 \def\l@dp@rsefootspec#1|#2|#3|#4|#5|#6|#7|{%
\l@dparsedendline 1356 \gdef\l@dparsedstartpage{#1}%
\l@dparsedendsub 1357 \gdef\l@dparsedstartline{#2}%
1358 \gdef\l@dparsedstartsub{#3}%
1359 \gdef\l@dparsedendpage{#4}%
1360 \gdef\l@dparsedendline{#5}%
1361 \gdef\l@dparsedendsub{#6}%
1362 }

```

Initialise the several number value macros.

```

1363 \def\l@dparsedstartpage{0}%
1364 \def\l@dparsedstartline{0}%
1365 \def\l@dparsedstartsub{0}%
1366 \def\l@dparsedendpage{0}%
1367 \def\l@dparsedendline{0}%
1368 \def\l@dparsedendsub{0}%
1369

```

`\setprintlines` First of all, we print the page numbers only if: 1) we're doing the lineation by page, and 2) the ending page number is different from the starting page number.

Just a reminder of the arguments:

```

\printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\printlines start-page | line | subline | end-page | line | subline | font

```

The macro `\setprintlines` does the work of deciding what numbers should be printed. Its arguments are the same as the first 6 of `\printlines`.

```

1370 \newcommand*{\setprintlines}[6]{%
1371 \l@d@pnumfalse \l@d@dashfalse
1372 \ifbypage@
1373 \ifnum#4=#1 \else
1374 \l@d@pnumtrue
1375 \l@d@dashtrue
1376 \fi
1377 \fi

```

We print the ending line number if: (1) we're printing the ending page number, or (2) it's different from the starting line number.

```

1378 \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
1379 \ifnum#2=#5 \else
1380 \l@d@elintrue
1381 \l@d@dashtrue
1382 \fi

```

We print the starting sub-line if it's nonzero.

```

1383 \l@d@ssubfalse
1384 \ifnum#3=0 \else

```

```

1385 \l@d@ssubtrue
1386 \fi

```

We print the ending sub-line if it's nonzero and: (1) it's different from the starting sub-line number, or (2) the ending line number is being printed.

```

1387 \l@d@eslfalse
1388 \ifnum#6=0 \else
1389 \ifnum#6=#3
1390 \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
1391 \else
1392 \l@d@esltrue
1393 \l@d@dashtrue
1394 \fi
1395 \fi}

```

`\printlines` Now we're ready to print it all. If the lineation is by `pstart`, we print the `pstart`.

```

1396 \def\printlines#1|#2|#3|#4|#5|#6|#7|{\begingroup
1397 \ifbypstart%
1398 \ifl@d@pairing%
1399 \ifledRcol%
1400 \thepstartR%
1401 \else%
1402 \thepstartL%
1403 \fi%
1404 \else%
1405 \thepstart%
1406 \fi%
1407 \fi%
1408 \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%

```

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 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).

```

1409 \ifl@d@pnum #1\fullstop\fi
1410 \linenumrep{#2}

1411 \ifl@d@ssub \fullstop \sublinenumrep{#3}\fi
1412 \ifl@d@dash \endashchar\fi
1413 \ifl@d@pnum #4\fullstop\fi
1414 \ifl@d@elin \linenumrep{#5}\fi
1415 \ifl@d@esl \ifl@d@elin \fullstop\fi \sublinenumrep{#6}\fi
1416 \endgroup}
1417 % \begin{macro}{\normalfootstart}
1418 % \phantomsection\label{normalvalues}\relax
1419 % \cs{normalfootstart}
1420 % is a standard footnote-starting macro, called in the output
1421 % routine whenever there are footnotes of this series to be
1422 % printed: it skips a bit and then draws a rule.
1423 %

```



```

1424 % Any \verb"footstart" macro must put onto the page something that
1425 % takes up space exactly equal to the \verb"\skip\footins" value for
1426 % the associated series of notes. \TeX\ makes page computations based
1427 % on that \cs{skip} value, and the output pages will suffer from
1428 % spacing problems if what you add takes up a different amount of
1429 % space.
1430 %
1431 % The \cs{leftskip} and \cs{rightskip} values
1432 % are both zeroed here.\phantomsection\label{footskips}
1433 % Similarly, these skips are cancelled in the \verb"vfootnote" macros
1434 % for the various types of notes. Strictly speaking,
1435 % this is necessary only
1436 % if you are using paragraphed footnotes, but we have put it here
1437 % and in the other \verb"vfootnote" macros too so that the behavior of
1438 % \Ledmac{} in this respect is general across all footnote types
1439 % (you can change this). What this means is that any \cs{leftskip}
1440 % and \cs{rightskip} you specify applies to the main text, but not
1441 % the footnotes. The footnotes continue to be of width \cs{hsize}.
1442 % \begin{macrocode}
1443 \newcommand*{\normalfootstart}[1]{%
1444 \vskip\skip\csname #1footins\endcsname
1445 \leftskip0pt \rightskip0pt
1446 \csname #1footnoterule\endcsname}

```

`\normalfootnoterule` `\normmalfootnoterule` is a standard footnote-rule macro, for use by a `footstart` macro: just the same as the PLAIN T<sub>E</sub>X footnote rule.

```
1447 \let\normalfootnoterule=\footnoterule
```

`\normalfootgroup` `\normalfootgroup` is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

1448 \newcommand*{\normalfootgroup}[1]{\unvbox\csname #1footins\endcsname}
1449

```

`\mpnormalfootgroup` A somewhat different version for minipages.

```

1450 \newcommand*{\mpnormalfootgroup}[1]{%
1451 \vskip\skip\@nameuse{mp#1footins}
1452 \normalcolor
1453 \@nameuse{#1footnoterule}
1454 \unvbox\csname mp#1footins\endcsname}}
1455

```

## 22.4 Standard footnote definitions

`\footnormal` We can now define all the parameters for the five series of footnotes; initially they use the ‘normal’ footnote formatting, which is set up by calling `\footnormal`. You can switch to another type of formatting by using `\footparagraph`, `\foottwocol`, or `\footthreecol`.

Switching to a variation of ‘normal’ formatting requires changing the quantities defined in `\footnormal`. The best way to proceed would be to make a copy of

this macro, with a different name, make your desired changes in that copy, and then invoke it, giving it the letter of the footnote series you wish to control.

(We have not defined baseline skip values like `\abaselineskip`, since this is one of the quantities set in `\notefontsetup`.)

What we want to do here is to say something like the following for each footnote series. (This is an example, not part of the actual `ledmac` code.)

```
\newinsert\Afootins
\skip\Afootins=12pt plus5pt minus5pt
\count\Afootins=1000
\dimen\Afootins=0.8\vsiz
\let\vAfootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule
```

Instead of repeating ourselves, we define a `\footnormal` macro that makes all these assignments for us, for any given series letter. This also makes it easy to change from any different system of formatting back to the `normal` setting.

```
\ledfootinsdim Have a constant value for the \dimen\footins
1456 \newcommand*{\ledfootinsdim}{0.8\vsiz}
1457
```

We begin by defining the five new insertion classes, and some `count` registers; these are `\outer` operations that can't be done inside `\footnormal`.

```
1458 \newinsert\Afootins \newinsert\Bfootins
1459 \newinsert\Cfootins \newinsert\Dfootins
1460 \newinsert\Efootins
```

Now we set up the `\footnormal` macro itself. It takes one argument: the footnote series letter.

```
1461 \newcommand*{\footnormal}[1]{%
1462 \expandafter\let\csname #1footstart\endcsname=\normalfootstart
1463 \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
1464 \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
1465 \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
1466 \expandafter\let\csname #1footnoterule\endcsname=%
1467 \normalfootnoterule
1468 \count\csname #1footins\endcsname=1000
1469 \dimen\csname #1footins\endcsname=\ledfootinsdim
1470 \skip\csname #1footins\endcsname=1.2em \@plus .6em \@minus .6em
```

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).

```
1471 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
1472 \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
1473 \count\csname mp#1footins\endcsname=1000
1474 \dimen\csname mp#1footins\endcsname=\ledfootinsdim
1475 \skip\csname mp#1footins\endcsname=1.2em \@plus .6em \@minus .6em
```

1476 }  
1477

Some of these values deserve comment: the `\dimen` setting allows 80% of the page to be occupied by notes; the `\skip` setting is deliberately flexible, since pages with lots of notes attached to many of the lines can be a bit hard for T<sub>E</sub>X to make.

And finally, we initialize the formatting for all the footnote series to be normal.

1478 \footnormal{A}  
1479 \footnormal{B}  
1480 \footnormal{C}  
1481 \footnormal{D}  
1482 \footnormal{E}  
1483

## 22.5 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 T<sub>E</sub>X of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

`\footparagraph` The `\footparagraph` macro sets up everything for one series of footnotes so that they'll 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 you are switching to paraphrased footnotes after having columnar ones, since they change this value (see below).

It is important to call `\footparagraph` only after `\hsize` has been set for the pages that use this series of notes; otherwise T<sub>E</sub>X 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 `\footparagraph` again afterwards to take account of the new value. The argument of `\footparagraph` is the letter (A–E) denoting the series of notes to be paraphrased.

1484 \newcommand\*{\footparagraph}[1]{%  
1485 \expandafter\newcount\csname prevpage#1@num\endcsname  
1486 \expandafter\let\csname #1footstart\endcsname=\parafootstart  
1487 \expandafter\let\csname v#1footnote\endcsname=\para@vfootnote  
1488 \expandafter\let\csname #1footfmt\endcsname=\parafootfmt  
1489 \expandafter\let\csname #1footgroup\endcsname=\para@footgroup  
1490 \count\csname #1footins\endcsname=1000  
1491 \para@footsetup{#1}

And the extra setup for minipages.

1492 \expandafter\let\csname mpv#1footnote\endcsname=\mppara@vfootnote  
1493 \expandafter\let\csname mp#1footgroup\endcsname=\mppara@footgroup  
1494 \count\csname mp#1footins\endcsname=1000  
1495 }

`\footfudgefiddle` For paragraphed footnotes  $\TeX$  has to estimate the amount of space required. If it underestimates this then the notes may get too long and run off the bottom of the text block. `\footfudgefiddle` can be increased from its default 64 (say to 70) to increase the estimate.

```
1496 \providecommand{\footfudgefiddle}{64}
```

`\para@footsetup` `\footparagraph` calls the `\para@footsetup` macro to calculate a special fudge factor, which is the ratio of the `\baselineskip` to the `\hsize`. We assume that the proper value of `\baselineskip` for the footnotes (normally 9 pt) has been set already, in `\notefontsetup`. The argument of the macro is again the note series letter.

I think that `\columnwidth` should be used here for LaTeX, not `\hsize`. I've also included `\footfudgefiddle`.

```
1497 \newcommand*{\para@footsetup}[1]{\notefontsetup
1498 \dimen0=\baselineskip
1499 \multiply\dimen0 by 1024
1500 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
1501 \expandafter
1502 \xdef\csname #1footfudgefactor\endcsname{%
1503 \expandafter\strip@pt\dimen0 }}
1504
```

EDMAC defines `\en@number` which does the same as the LaTeX kernel `\strip@pt`, namely strip the characters pt from a dimen value. I'll use `\strip@pt`.

`\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). You might have decided to change this for other kinds of note, but here it should stay as it is. 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.

```
1505 \newcommand*{\parafootstart}[1]{%
1506 \rightskip=0pt \leftskip=0pt \parindent=0pt
1507 \vskip\skip\csname #1footins\endcsname
1508 \csname #1footnoterule\endcsname}
```

`\para@vfootnote` `\para@vfootnote` is a version of the `\vfootnote` command that's 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  $\text{\TeX}$  does not expect to have to break lines, it does not insert certain items like  $\text{\discretionary}$ s. If you later unbox these  $\text{\hboxes}$  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  $\text{\hboxes}$  when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.<sup>22</sup>

Wayne Sullivan pointed out to us another subtle problem that arises from the same cause:  $\text{\TeX}$  also leaves the  $\text{\language}$  whatsit nodes out of the horizontal list.<sup>23</sup> 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 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  $\text{\hbox}$  in the first place, but instead to collect it in a  $\text{\vbox}$  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  $\text{\vbox}$ , as well as the  $\text{\hboxes}$  inside it, but that's not too hard. For details, we refer you to Michael's article, where the issues are clearly explained.<sup>24</sup> Michael's unboxing macro is called  $\text{\unvxh}$ :  $\text{\unvbox}$ , extract the last line, and  $\text{\unhbox}$  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  $\text{\vbox}$  the way we are doing.<sup>25</sup> In other words, be very careful not to say  $\text{\break}$ , or  $\text{\penalty-10000}$ , 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  $\text{\penalty-9999}$  will be quite okay. Just don't make the break mandatory. We haven't applied any of Michael's solutions here, since we feel that the problem is exiguous, and  $\text{\ledmac}$  is quite baroque enough already. If you think you are having this problem, look up Michael's solutions.

One more thing; we set  $\text{\leftskip}$  and  $\text{\rightskip}$  to zero. This has the effect of neutralizing any such skips which may apply to the main text (cf. p. ?? above). We need to do this, since  $\text{\footfudgefactor}$  is calculated on the assumption that the notes are  $\text{\hsize}$  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.

```
1509 \newcommand*{\para@vfootnote}[2]{%
1510 \insert\csname #1footins\endcsname
1511 \bgroup
1512 \notefontsetup
```

<sup>22</sup>Michael Downes, 'Line Breaking in  $\text{\unhboxed}$  Text', *TUGboat* **11** (1990), pp. 605–612.

<sup>23</sup>See *The TeXbook*, p. 455 (editions after January 1990).

<sup>24</sup>Wayne supplied his own macros to do this, but since they were almost identical to Michael's, we have used the latter's  $\text{\unvxh}$  macro since it is publicly documented.

<sup>25</sup>'Line Breaking', p. 610.

```

1513 \footsplitskips
1514 \setbox0=\vbox{\hsize=\maxdimen
1515 \noindent\csname #1footfmt\endcsname#2[#1]}%
1516 \setbox0=\hbox{\unvxh0[#1]}%
1517 \dp0=0pt
1518 \ht0=\csname #1footfudgefactor\endcsname\wd0

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

1519 \box0
1520 \penalty0
1521 \egroup}
1522

```

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 T<sub>E</sub>X attempts to split foot paragraphs. After trying out such a split (see *The T<sub>E</sub>Xbook*, p.124), T<sub>E</sub>X 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 doesn't 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`).

`\mppara@vfootnote` This version is for minipages.

```

1523 \newcommand*{\mppara@vfootnote}[2]{%
1524 \global\setbox\@nameuse{mp#1footins}\vbox{%
1525 \unvbox\@nameuse{mp#1footins}%
1526 \notefontsetup
1527 \footsplitskips
1528 \setbox0=\vbox{\hsize=\maxdimen
1529 \noindent\color@begingroup\csname #1footfmt\endcsname #2[#1]\color@endgroup}%
1530 \setbox0=\hbox{\unvxh0[#1]}%
1531 \dp0=\z@
1532 \ht0=\csname #1footfudgefactor\endcsname\wd0
1533 \box0
1534 \penalty0
1535 }}
1536

```

`\unvxh` Here is (modified) Michael's definition of `\unvxh`, used above. Michael's macro also takes care to remove some unwanted penalties and glue that T<sub>E</sub>X automatically attaches to the end of paragraphs. When T<sub>E</sub>X 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 T<sub>E</sub>Xbook*, pp.99–100). `\unvxh` cancels these unwanted paragraph-final items using `\unskip` and `\unpenalty`.

```

1537 \newcommandx*{\unvxh}[2][2=Z]{% 2th is optional for retro-compatibility

```

```

1538 \setbox0=\vbox{\unvbox#1%
1539 \global\setbox1=\lastbox}%
1540 \unhbox1
1541 \unskip % remove \rightskip,
1542 \unskip % remove \parfillskip,
1543 \unpenalty % remove \penalty of 10000,
1544 \hskip\csuse{afternote@#2}} % but add the glue to go between the notes
1545

```

`\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 ascendant compatibility).

```

1546 \newcommand*{\parafootfmt}[4][4=Z]{%
1547 \insertparafootsep{#4}%
1548 \ledsetnormalparstuff%
1549 \printlinefootnote{#1}{#4}%
1550 {\select@lemmafont#1|#2}%
1551 \iftoggle{nosep@}{\hskip\csuse{inplaceoflemmaseparator@#4}}{\ifcsempy{lemmaseparator@#4}%
1552 {\hskip\csuse{inplaceoflemmaseparator@#4}}}%
1553 {\nobreak\hskip\csuse{beforelemmaseparator@#4}\csuse{lemmaseparator@#4}\hskip\csuse{afterlemmasep
1554 }}%
1555 #3\penalty-10 }

```

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 `\insertparafootsep` command is used to insert the `\parafootsep@series` between each note in the *same* page.

`\para@footgroup` 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 `\para@vfootnote`.

The call to `\notefontsetup` is to ensure that the correct `\baselineskip` for the footnotes is used. The argument is the note series letter.

```

1556 \newcommand*{\para@footgroup}[1]{%
1557 \unvbox\csname #1footins\endcsname
1558 \makehboxofhboxes
1559 \setbox0=\hbox{\unhbox0 \removehboxes}%
1560 \notefontsetup
1561 \noindent\unhbox0\par}
1562

```

`\mppara@footgroup` The minipage version.

```

1563 \newcommand*{\mppara@footgroup}[1]{%
1564 \vskip\skip@nameuse{mp#1footins}
1565 \normalcolor
1566 \@nameuse{#1footnoterule}%
1567 \unvbox\csname mp#1footins\endcsname

```

```

1568 \makehboxoffhboxes
1569 \setbox0=\hbox{\unhbox0 \removehboxes}%
1570 \notefontsetup
1571 \noindent\unhbox0\par}}
1572
\makehboxoffhboxes
\removehboxes 1573 \newcommand*{\makehboxoffhboxes}{\setbox0=\hbox{}%
1574 \loop
1575 \unpenalty
1576 \setbox2=\lastbox
1577 \ifhbox2
1578 \setbox0=\hbox{\box2\unhbox0}%
1579 \repeat}
1580
1581 \newcommand*{\removehboxes}{\setbox0=\lastbox
1582 \ifhbox0{\removehboxes}\unhbox0 \fi}
1583

```

### 22.5.1 Insertion of footnotes separator

The command `\insertparafootsep{<series>}` must be called at the beginning of `\parafootftm` (and like commands).

```

\prevpage@num
\insertparafootsep 1584 \newcommand{\insertparafootsep}[1]{%
1585 \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
1586 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
1587 {\ifnumequal{\csuse{prevline#1}}{\line@num}%
1588 {\ifcsempy{symplicenum}{\csuse{parafootsep@#1}}{}}}%
1589 {\csuse{parafootsep@#1}}}%
1590 }%
1591 {\csuse{parafootsep@#1}}}%
1592 }%
1593 {}%
1594 \global\csname prevpage#1@num\endcsname=\page@num%
1595 }

```

## 22.6 Columnar footnotes

`\rigidbalance` We will now define macros for three-column notes and two-column notes. Both sets of macros will use `\rigidbalance`, which splits a box (#1) into into a number (#2) of 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 don't depend on white space. Note also the extra unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it goes into the alignment.

The LaTeX `\line` macro has no relationship to the TeX `\line`. The LaTeX equivalent is `\@@line`.



```

1596 \newcount\@k \newdimen\@h
1597 \newcommand*\rigidbalance}[3]{\setbox0=\box#1 \@k=#2 \@h=#3
1598 \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
1599 \valign{##\vfil\cr\dosplits}}}}
1600
1601 \newcommand*\dosplits{\ifnum\@k>0 \noalign{\hfil}\splitoff
1602 \global\advance\@k-1\cr\dosplits\fi}
1603
1604 \newcommand*\splitoff{\dimen0=\ht0
1605 \divide\dimen0 by\@k \advance\dimen0 by\@h
1606 \setbox2 \vsplit0 to \dimen0
1607 \unvbox2 }
1608

```

### Three columns

`\footthreecol` You say `\footthreecol{A}` to have the A series of footnotes typeset in three columns. It is important to call this only after `\hsize` has been set for the document.

```

1609 \newcommand*\footthreecol}[1]{%
1610 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
1611 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
1612 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
1613 \threecolfootsetup{#1}

```

The additional setup for minipages.

```

1614 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
1615 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
1616 \mpthreecolfootsetup{#1}
1617 }
1618

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (p. ?? 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 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 doesn't apply the `\count` scaling.

```

1619 \newcommand*\threecolfootsetup}[1]{%

```

```

1620 \count\csname #1footins\endcsname 333
1621 \multiply\dimen\csname #1footins\endcsname \thr@@}

```

`\mpthreecolfootsetup` The setup for minipages.

```

1622 \newcommand*{\mpthreecolfootsetup}[1]{%
1623 \count\csname mp#1footins\endcsname 333
1624 \multiply\dimen\csname mp#1footins\endcsname \thr@@}
1625

```

`\threecolvfootnote` `\threecolvfootnote` is the `\vfootnote` command for three-column notes. The call to `\notefontsetup` ensures that the `\splittopskip` and `\splitmaxdepth` take their values from the right `\strutbox`: the one used in 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 2) the full text of the note (including numbers, lemma and text).

```

1626 \newcommand*{\threecolvfootnote}[2]{%
1627 \insert\csname #1footins\endcsname\bgroup
1628 \notefontsetup
1629 \footssplitskips
1630 \csname #1footfmt\endcsname #2[#1]\egroup}

```

`\threecolfootfmt` `\threecolfootfmt` is the command that formats one note. It uses `\raggedright`, which will usually be preferable with such short lines. Setting the `\parindent` to zero means that, within each individual note, the lines begin flush left.

The arguments are 1) the line numbers, 2) the lemma and 3) the text of the `-footnote` command 4) optional (for ascendant compatibility) : the series.

```

1631 \newcommandx*{\threecolfootfmt}[4][4=Z]{%
1632 \normal@pars
1633 \hsize .3\hsize
1634 \parindent=0pt
1635 \tolerance=5000
1636 \raggedright
1637 \leavevmode
1638 \strut{\printlinefootnote{#1}{#4}}%
1639 {\select@lemmafont#1|#2}%
1640 \iftoggle{nosep@}{\hskip\csuse{inplaceoflemmaseparator@#4}}{\ifcsemt{lemmaseparator@#4}}%
1641 {\hskip\csuse{inplaceoflemmaseparator@#4}}%
1642 {\nobreak\hskip\csuse{beforelemmaseparator@#4}\csuse{lemmaseparator@#4}\hskip\csuse{afterlemmaseparator@#4}}%
1643 }%
1644 #3\strut\par\allowbreak}

```

`\threecolfootgroup` And here is the `footgroup` macro that's called within the output routine to re-group the notes into three columns. Once again, the call to `\notefontsetup` 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.

```
1645 \newcommand*{\threecolfootgroup}[1]{\notefontsetup
1646 \splittopskip=\ht\strutbox
1647 \expandafter
1648 \rigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}}
```

`\mpthreecolfootgroup` The setup for minipages.

```
1649 \newcommand*{\mpthreecolfootgroup}[1]{%
1650 \vskip\skip\@nameuse{mp#1footins}
1651 \normalcolor
1652 \@nameuse{#1footnoterule}
1653 \splittopskip=\ht\strutbox
1654 \expandafter
1655 \rigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
1656
```

## Two columns

`\foottwocol` You say `\foottwocol{A}` to have the A series of footnotes typeset in two columns. It is important to call this only after `\hsize` has been set for the document.

```
1657 \newcommand*{\foottwocol}[1]{%
1658 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
1659 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
1660 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
1661 \twocolfootsetup{#1}}
```

The additional setup for minipages.

```
1662 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
1663 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
1664 \mptwocolfootsetup{#1}
1665 }
1666
```

`\twocolfootsetup` Here is a series of macros which are very similar to their three-column counterparts.  
`\twocolvfootnote` In this case, each note is assumed to contribute only a half a line of text. And  
`\twocolfootfmt` the notes are set in columns `0.45\hsize` wide, giving a gap between them of one  
`\twocolfootgroup` tenth of the `\hsize`.

```
1667 \newcommand*{\twocolfootsetup}[1]{%
1668 \count\csname #1footins\endcsname 500
1669 \multiply\dimen\csname #1footins\endcsname \tw@}

1670 \newcommand*{\twocolvfootnote}[2]{\insert\csname #1footins\endcsname\bgroup
1671 \notefontsetup
1672 \footsplitskips
1673 \csname #1footfmt\endcsname #2[#1]\egroup}
```

```

1674 \newcommandx*{\twocolfootfmt}[4][4=Z]{% 4th arg is optional, for ascendant compatibility
1675 \normal@pars
1676 \hsize .45\hsize
1677 \parindent=0pt
1678 \tolerance=5000
1679 \raggedright
1680 \leavevmode
1681 \strut{\printlinefootnote{#1}{#4}}%
1682 {\select@lemmafont#1|#2}%
1683 \iftoggle{nosep@}{\hskip\csuse{inplaceoflemmaseparator@#4}}{\ifcsempy{lemmaseparator@#4}
1684 {\hskip\csuse{inplaceoflemmaseparator@#4}}}%
1685 {\nobreak\hskip\csuse{beforelemmaseparator@#4}\csuse{lemmaseparator@#4}\hskip\csuse{a
1686 }}%
1687 #3\strut\par\allowbreak}

1688 \newcommand*{\twocolfootgroup}[1]{\notefontsetup
1689 \splittopskip=\ht\strutbox
1690 \expandafter
1691 \rigidbalance\csname #1footins\endcsname \tw@ \splittopskip}}
1692

```

`\mptwocolfootsetup` The versions for minipages.

```

\mptwocolfootgroup 1693 \newcommand*{\mptwocolfootsetup}[1]{%
1694 \count\csname mp#1footins\endcsname 500
1695 \multiply\dimen\csname mp#1footins\endcsname \tw@}

1696 \newcommand*{\mptwocolfootgroup}[1]{%
1697 \vskip\skip\@nameuse{mp#1footins}
1698 \normalcolor
1699 \@nameuse{#1footnoterule}
1700 \splittopskip=\ht\strutbox
1701 \expandafter
1702 \rigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
1703

```

## 22.7 Display

### 22.7.1 Options

`\numberonlyfirstinline` [*series*] [*position*] is used to print line number only in the first time. It set to true the `\ifnumberonlyfirstinline@series` toggle.

`\numberonlyfirstinline`

```

1704 \newtoggle{numberonlyfirstinline@A}
1705 \newtoggle{numberonlyfirstinline@B}
1706 \newtoggle{numberonlyfirstinline@C}
1707 \newtoggle{numberonlyfirstinline@D}
1708 \newtoggle{numberonlyfirstinline@E}
1709 \newtoggle{numberonlyfirstinline@Z}
1710
1711 \newcommandx*{\numberonlyfirstinline}[2][1,2={true}]{%

```

```

1712 \ifstrempy{#1}{
1713 \settoggle{numberonlyfirstinline@A}{#2}
1714 \settoggle{numberonlyfirstinline@B}{#2}
1715 \settoggle{numberonlyfirstinline@C}{#2}
1716 \settoggle{numberonlyfirstinline@D}{#2}
1717 \settoggle{numberonlyfirstinline@E}{#2}
1718 }
1719 {
1720 \settoggle{numberonlyfirstinline@#1}{#2}
1721 }
1722 }

```

`\ifledplinenum` `\ifledplinenum` and `\symlinenum` are kept for ascendant compatibility. Should'nt  
`\symlinenum` be use anymore.

```

1723 \newif\ifledplinenum
1724 \ledplinenumtrue
1725 \newcommand*{\symlinenum}{}

```

`\symlinenum` `\symlinenum` is used to print a symbol in place of line number, in conjunction  
with `\numberonlyfirstinline`.

```

1726 \def\symlinenum@A{\symlinenum}
1727 \def\symlinenum@B{\symlinenum}
1728 \def\symlinenum@C{\symlinenum}
1729 \def\symlinenum@D{\symlinenum}
1730 \def\symlinenum@E{\symlinenum}
1731 \def\symlinenum@Z{\symlinenum}
1732 \newcommandx{\symlinenum}[2][1]{%
1733 \ifstrequal{#1}{}% If arg 1 is empty
1734 {%
1735 \csdef{symlinenum@A}{#2}%
1736 \csdef{symlinenum@B}{#2}%
1737 \csdef{symlinenum@C}{#2}%
1738 \csdef{symlinenum@D}{#2}%
1739 \csdef{symlinenum@E}{#2}%
1740 \csdef{symlinenum@Z}{#2}%
1741 }%
1742 {%
1743 \csdef{symlinenum@#1}{#2}%
1744 }
1745 }

```

```

1746 % \end{macrocode}

```

```

1747 % \end{macro}

```

```

1748 % \begin{macro}{\nonumberinfootnote}

```

1749 % `\cs{nonumberinfootnote}\oarg{series}\oarg{position}` is used to disable line number printing in foot

```

1750 % \begin{macrocode}

```

```

1751 \newtoggle{nonumberinfootnote@A}

```

```

1752 \newtoggle{nonumberinfootnote@B}

```

```

1753 \newtoggle{nonumberinfootnote@C}

```

```

1754 \newtoggle{nonumberinfootnote@D}

```

```

1755 \newtoggle{nonumberinfootnote@E}
1756 \newtoggle{nonumberinfootnote@Z}
1757 \newcommandx{\nonumberinfootnote}[2][1,2={true}]{%
1758 \ifstrempy{#1}{
1759 \settoggle{nonumberinfootnote@A}{#2}
1760 \settoggle{nonumberinfootnote@B}{#2}
1761 \settoggle{nonumberinfootnote@C}{#2}
1762 \settoggle{nonumberinfootnote@D}{#2}
1763 \settoggle{nonumberinfootnote@E}{#2}
1764 }
1765 {
1766 \settoggle{nonumberinfootnote@#1}{#2}
1767 }
1768 }

```

`\nonum@` `\nonum@` toggle is used to disable line number printing in a particular footnote.

```
1769 \newtoggle{nonum@}
```

`\nosep@` `\nonum@` toggle is used to disable lemma separator in a particular footnote.

```
1770 \newtoggle{nosep@}
```

`\beforenumberinfootnote` `\beforenumberinfootnote[ $\langle series \rangle$ ]{ $\langle length \rangle$ }` is used to add glue before line number printing in footnotes.

```

1771 \def\beforenumberinfootnote@A{0pt}
1772 \def\beforenumberinfootnote@B{0pt}
1773 \def\beforenumberinfootnote@C{0pt}
1774 \def\beforenumberinfootnote@D{0pt}
1775 \def\beforenumberinfootnote@E{0pt}
1776 \def\beforenumberinfootnote@Z{0pt}
1777 \newcommandx*\beforenumberinfootnote[2][1,usedefault]{%
1778 \ifstrequal{#1}{}% If arg 1 is empty
1779 {%
1780 \csdef{beforenumberinfootnote@A}{#2}%
1781 \csdef{beforenumberinfootnote@B}{#2}%
1782 \csdef{beforenumberinfootnote@C}{#2}%
1783 \csdef{beforenumberinfootnote@D}{#2}%
1784 \csdef{beforenumberinfootnote@E}{#2}%
1785 \csdef{beforenumberinfootnote@Z}{#2}%
1786 }%
1787 {%
1788 \csdef{beforenumberinfootnote@#1}{#2}%
1789 }%
1790 }

```

`\afternumberinfootnote` `\afternumberinfootnote[ $\langle series \rangle$ ]{ $\langle length \rangle$ }` is used add glue after line number in footnotes.

```

1791 \def\afternumberinfootnote@A{0.5em}
1792 \def\afternumberinfootnote@B{0.5em}
1793 \def\afternumberinfootnote@C{0.5em}

```

```

1794 \def\afternumberinfootnote@D{0.5em}
1795 \def\afternumberinfootnote@E{0.5em}
1796 \def\afternumberinfootnote@Z{0.5em}
1797 \newcommandx*{\afternumberinfootnote}[2][1,usedefault]{%
1798 \ifstrequal{#1}{}% If arg 1 is empty
1799 {%
1800 \csdef{afternumberinfootnote@A}{#2}%
1801 \csdef{afternumberinfootnote@B}{#2}%
1802 \csdef{afternumberinfootnote@C}{#2}%
1803 \csdef{afternumberinfootnote@D}{#2}%
1804 \csdef{afternumberinfootnote@E}{#2}%
1805 \csdef{afternumberinfootnote@Z}{#2}%
1806 }%
1807 {%
1808 \csdef{afternumberinfootnote@#1}{#2}%
1809 }%
1810 }

```

`\beforesymmlinenum` `\beforesymmlinenum[<series>]{<length>}` is used to add glue before line symbol.

```

1811 \def\beforesymmlinenum@A{\beforenumberinfootnote@A}
1812 \def\beforesymmlinenum@B{\beforenumberinfootnote@B}
1813 \def\beforesymmlinenum@C{\beforenumberinfootnote@C}
1814 \def\beforesymmlinenum@D{\beforenumberinfootnote@D}
1815 \def\beforesymmlinenum@E{\beforenumberinfootnote@E}
1816 \def\beforesymmlinenum@Z{\beforenumberinfootnote@Z}
1817 \newcommandx*{\beforesymmlinenum}[2][1,usedefault]{%
1818 \ifstrequal{#1}{}% If arg 1 is empty
1819 {%
1820 \csdef{beforesymmlinenum@A}{#2}%
1821 \csdef{beforesymmlinenum@B}{#2}%
1822 \csdef{beforesymmlinenum@C}{#2}%
1823 \csdef{beforesymmlinenum@D}{#2}%
1824 \csdef{beforesymmlinenum@E}{#2}%
1825 \csdef{beforesymmlinenum@Z}{#2}%
1826 }%
1827 {%
1828 \csdef{beforesymmlinenum@#1}{#2}%
1829 }%
1830 }

```

`\aftersymmlinenum` `\aftersymmlinenum[<series>]{<length>}` is used to add glue after line symbol.

```

1831 \def\aftersymmlinenum@A{\afternumberinfootnote@A}
1832 \def\aftersymmlinenum@B{\afternumberinfootnote@B}
1833 \def\aftersymmlinenum@C{\afternumberinfootnote@C}
1834 \def\aftersymmlinenum@D{\afternumberinfootnote@D}
1835 \def\aftersymmlinenum@E{\afternumberinfootnote@E}
1836 \def\aftersymmlinenum@Z{\afternumberinfootnote@Z}
1837 \newcommandx*{\aftersymmlinenum}[2][1,usedefault]{%
1838 \ifstrequal{#1}{}% If arg 1 is empty
1839 {%

```

```

1840 \csdef{aftersymmlinenum@A}{#2}%
1841 \csdef{aftersymmlinenum@B}{#2}%
1842 \csdef{aftersymmlinenum@C}{#2}%
1843 \csdef{aftersymmlinenum@D}{#2}%
1844 \csdef{aftersymmlinenum@E}{#2}%
1845 \csdef{aftersymmlinenum@Z}{#2}%
1846 }%
1847 {%
1848 \csdef{aftersymmlinenum@#1}{#2}%
1849 }%
1850 }

```

`\inplaceofnumber` `\inplaceofnumber[(series)]{(value)}` is used to define glue which is printed where no number or symbol is printed.

```

1851 \def\inplaceofnumber@A{1em}
1852 \def\inplaceofnumber@B{1em}
1853 \def\inplaceofnumber@C{1em}
1854 \def\inplaceofnumber@D{1em}
1855 \def\inplaceofnumber@E{1em}
1856 \def\inplaceofnumber@Z{1em}
1857 \newcommandx*\inplaceofnumber[2][1,usedefault]{%
1858 \ifstrequal{#1}{}% If arg 1 is empty
1859 {%
1860 \csdef{inplaceofnumber@A}{#2}%
1861 \csdef{inplaceofnumber@B}{#2}%
1862 \csdef{inplaceofnumber@C}{#2}%
1863 \csdef{inplaceofnumber@D}{#2}%
1864 \csdef{inplaceofnumber@E}{#2}%
1865 \csdef{inplaceofnumber@Z}{#2}%
1866 }%
1867 {%
1868 \csdef{inplaceofnumber@#1}{#2}%
1869 }%
1870 }

```

`\lemmaseparator` The `\lemmaseparator[(series)]{(lemma)}` is used to change lemma separator.

```

1871 \let\lemmaseparator@A\rbracket
1872 \let\lemmaseparator@B\rbracket
1873 \let\lemmaseparator@C\rbracket
1874 \let\lemmaseparator@D\rbracket
1875 \let\lemmaseparator@E\rbracket
1876 \let\lemmaseparator@Z\rbracket
1877 \let\lemmaseparator@E\rbracket
1878 \newcommandx*\lemmaseparator[2][1,usedefault]{%
1879 \ifstrequal{#1}{}% If arg 1 is empty
1880 {%
1881 \csdef{lemmaseparator@A}{#2}%
1882 \csdef{lemmaseparator@B}{#2}%
1883 \csdef{lemmaseparator@C}{#2}%
1884 \csdef{lemmaseparator@D}{#2}%

```



```

1885 \csdef{lemmaseparator@E}{#2}%
1886 \csdef{lemmaseparator@Z}{#2}%
1887 }%
1888 {%
1889 \csdef{lemmaseparator@#1}{#2}%
1890 }%
1891 }

```

`\beforelemmaseparator` The `\beforelemmaseparator[<series>]{<length>}` is used to change glue between lemma and separator.

```

1892 \def\beforelemmaseparator@A{0em}
1893 \def\beforelemmaseparator@B{0em}
1894 \def\beforelemmaseparator@C{0em}
1895 \def\beforelemmaseparator@D{0em}
1896 \def\beforelemmaseparator@E{0em}
1897 \def\beforelemmaseparator@Z{0em}
1898 \newcommand*{\beforelemmaseparator}[2][1,usedefault]{%
1899 \ifstrequal{#1}{}% If arg 1 is empty
1900 {%
1901 \csdef{beforelemmaseparator@A}{#2}%
1902 \csdef{beforelemmaseparator@B}{#2}%
1903 \csdef{beforelemmaseparator@C}{#2}%
1904 \csdef{beforelemmaseparator@D}{#2}%
1905 \csdef{beforelemmaseparator@E}{#2}%
1906 \csdef{beforelemmaseparator@Z}{#2}%
1907 }%
1908 {%
1909 \csdef{beforelemmaseparator@#1}{#2}%
1910 }%
1911 }

```

`\afterlemmaseparator` The `\afterlemmaseparator[<series>]{<length>}` is used to change glue between lemma separator and note.

```

1912 \def\afterlemmaseparator@A{0.5em}
1913 \def\afterlemmaseparator@B{0.5em}
1914 \def\afterlemmaseparator@C{0.5em}
1915 \def\afterlemmaseparator@D{0.5em}
1916 \def\afterlemmaseparator@E{0.5em}
1917 \def\afterlemmaseparator@Z{0.5em}
1918 \newcommand*{\afterlemmaseparator}[2][1,usedefault]{%
1919 \ifstrequal{#1}{}% If arg 1 is empty
1920 {%
1921 \csdef{afterlemmaseparator@A}{#2}%
1922 \csdef{afterlemmaseparator@B}{#2}%
1923 \csdef{afterlemmaseparator@C}{#2}%
1924 \csdef{afterlemmaseparator@D}{#2}%
1925 \csdef{afterlemmaseparator@E}{#2}%
1926 \csdef{afterlemmaseparator@Z}{#2}%
1927 }%

```

```

1928 {%
1929 \csdef{afterlemmaseparator@#1}{#2}%
1930 }%
1931 }

```

`\nolemmaseparator` `\nolemmaseparator[<series>]` is just an alias for `\lemmaseparator[<series>]{}`.

```

1932 \newcommandx*{\nolemmaseparator}[1][1]{\lemmaseparator[#1]{}}
1933 % \end{macrocode}
1934 % \end{macro}
1935 % \begin{macro}{\inplaceofnumber}
1936 % \cs{inplaceoflemmaseparator}\oarg{series}\marg{value} is used to define glue which is p
1937 % \begin{macrocode}
1938 \def\inplaceoflemmaseparator@A{1em}
1939 \def\inplaceoflemmaseparator@B{1em}
1940 \def\inplaceoflemmaseparator@C{1em}
1941 \def\inplaceoflemmaseparator@D{1em}
1942 \def\inplaceoflemmaseparator@E{1em}
1943 \def\inplaceoflemmaseparator@Z{1em}
1944 \newcommandx*{\inplaceoflemmaseparator}[2][1,usedefault]{%
1945 \ifstrequal{#1}{}% If arg 1 is empty
1946 {%
1947 \csdef{inplaceoflemmaseparator@A}{#2}%
1948 \csdef{inplaceoflemmaseparator@B}{#2}%
1949 \csdef{inplaceoflemmaseparator@C}{#2}%
1950 \csdef{inplaceoflemmaseparator@D}{#2}%
1951 \csdef{inplaceoflemmaseparator@E}{#2}%
1952 \csdef{inplaceoflemmaseparator@Z}{#2}%
1953 }%
1954 {%
1955 \csdef{inplaceoflemmaseparator@#1}{#2}%
1956 }%
1957 }

```

`\afternote` The `\afternote[<series>]{<length>}` is used to change length after note.

There is a point to be careful about regarding the `\afternote@series` commands. Remember that in `\para@vfootnote` we do some measurements on the footnote box, and use the resulting size to make an estimate of how much the note will contribute to the height of our final footnote paragraph. This information is used by the output routine to allocate the right amount of vertical space on the page for the notes (*The TeXbook*, pp. 398–399).

The length of the footnote includes the natural size of the glue specified by `\afternote@series`, but not its stretch or shrink components, since at this point the note has no need to stretch or shrink. Later, when the paragraph is actually composed by `\parafootgroup` in the output routine, `TeX` will almost certainly do some stretching and shrinking of this glue in order to make the paragraph look nice. Probably the stretching and shrinking over the whole paragraph will cancel each other out. But if not, the actual vertical size of the paragraph may not match the size the output routine had been told to expect, and you may get an overfull/underfull `\vbox` message from the output routine. To minimize

the risk of this, you can do two things: keep the `plus` and `minus` components of `\interparanoteglue` small compared with its natural glue, and keep them the same as each other. As a general precaution, keep the size and flexibility of the `\skip\footins` glue on the high side too: because the reckoning is approximate, footnote blocks may be up to a line bigger or smaller than the output routine allows for, so keep some flexible space between the text and the notes.

```

1958 \def\afternote@A{1em plus.4em minus.4em}
1959 \def\afternote@B{1em plus.4em minus.4em}
1960 \def\afternote@C{1em plus.4em minus.4em}
1961 \def\afternote@D{1em plus.4em minus.4em}
1962 \def\afternote@E{1em plus.4em minus.4em}
1963 \def\afternote@Z{1em plus.4em minus.4em}
1964 \newcommandx*\afternote}[2][1,usedefault]{%
1965 \ifstrequal{#1}{}% If arg 1 is empty
1966 {%
1967 \csdef{afternote@A}{#2}%
1968 \csdef{afternote@B}{#2}%
1969 \csdef{afternote@C}{#2}%
1970 \csdef{afternote@D}{#2}%
1971 \csdef{afternote@E}{#2}%
1972 \csdef{afternote@Z}{#2}%
1973 }%
1974 {%
1975 \csdef{afternote@#1}{#2}%
1976 }%
1977 }
1978 % \end{macrocode}
1979 % \end{macro}
1980 % \begin{macro}{\interparanoteglue}
1981 % \begin{macro}{\ipn@skip}
1982 % The \cs{ipn@skip} skip and \cs{interparanoteglue} command are kept for ascendant compatibility, but
1983 % \begin{macrocode}
1984 \newskip\ipn@skip
1985 \newcommand*\interparanoteglue[1]{%
1986 {\notefontsetup\global\ipn@skip=#1 \relax}}
1987 \interparanoteglue{1em plus.4em minus.4em}

```

`\parafootftmsep` The `\parafootftmsep` macro is kept for ascendant compatibility. Its default value of `\parafootsep@series`.

```
1988 \newcommand{\parafootftmsep}{}

```

`\parafootsep` The `\parafootsep[series]{sep}` is used for separator of notes in paragraphed footnotes.

```

1989 \def\parafootsep@A{\parafootftmsep}
1990 \def\parafootsep@B{\parafootftmsep}
1991 \def\parafootsep@C{\parafootftmsep}
1992 \def\parafootsep@D{\parafootftmsep}
1993 \def\parafootsep@E{\parafootftmsep}
1994 \def\parafootsep@Z{\parafootftmsep}

```

```

1995 \newcommandx*{\parafootsep}[2][1,usedefault]{%
1996 \ifstrequal{#1}{}% If arg 1 is empty
1997 {%
1998 \csdef{parafootsep@A}{#2}%
1999 \csdef{parafootsep@B}{#2}%
2000 \csdef{parafootsep@C}{#2}%
2001 \csdef{parafootsep@D}{#2}%
2002 \csdef{parafootsep@E}{#2}%
2003 \csdef{parafootsep@Z}{#2}%
2004 }%
2005 {%
2006 \csdef{parafootsep@#1}{#2}%
2007 }%
2008 }
2009 % \end{macrocode}
2010 % \end{macro}
2011 % \subsubsection{Line number printing}
2012 % \begin{macro}{\printlinefootnote}
2013 % The \cs{printlinefootnote} macro is called in each \cs{<type>footfmt} command. It prints
2014 % \begin{macrocode}
2015 \newcommand{\printlinefootnote}[2]{%
2016 \iftoggle{nonum@}{%Try if the line number must printed for this specific not (by default
2017 \hspace{\csuse{inplaceofnumber@#2}}%
2018 }%
2019 {%
2020 {%
2021 \iftoggle{nonumberinfootnote@#2}%Try if the line number must printed (by default
2022 {%
2023 \hspace{\csuse{inplaceofnumber@#2}}%
2024 }%
2025 {%
2026 {\iftoggle{numberonlyfirstinline@#2}% If for this series the line number must
2027 {%
2028 \ifcsdef{prevline#2}%
2029 {%Be sure the \prevline exists.
2030 \ifnumequal{\csuse{prevline#2}}{\line@num}%Try it
2031 {%
2032 \ifcsstring{symlinenum@#2}{}% Try if a symbol is define
2033 {%
2034 \hspace{\csuse{inplaceofnumber@#2}}%
2035 }%
2036 {\hspace{\csuse{beforeymlinenum@#2}}\notenumfont\csuse{symlinenum
2037 }%
2038 {%
2039 \hspace{\csuse{beforenumberinfootnote@#2}}\notenumfont\printlines#1|
2040 }%
2041 }%
2042 }%
2043 \hspace{\csuse{beforenumberinfootnote@#2}}\notenumfont\printlines#1|
2044 }%

```

```

2045 }%
2046 {%
2047 \hspace{\csuse{beforenumberinfootnote@#2}}\notenumfont\printlines#1|\hspace{\csuse{afternumberinfootnote@#2}}
2048 }%
2049 \csnumgdef{prevline#2}{\line@num}%
2050 }%
2051 }%
2052 }%
2053 }%
2054 }

```

## 23 Output routine

Now we begin the output routine and associated things.

I have deleted all the crop mark code.

There are a couple of macros from plain TeX that we need (at least for now).

```

\pageno \pageno is a page number, starting at 1, and \advancepageno increments the num-
\advancepageno ber.
2055 \countdef\pageno=0 \pageno=1
2056 \newcommand*{\advancepageno}{\ifnum\pageno<\z@ \global\advance\pageno\m@ne
2057 \else\global\advance\pageno\@ne\fi}
2058

```

The next portion is probably the trickiest part of moving from TeX to LaTeX. The original code is below, but we need something very different.

This is a new output routine, with changes to handle printing all our footnotes. Those changes have not been added directly, but are in macros that get called here: that should make it easier to see what would need to be taken over to a different output routine. We continue to use the `\pagebody`, `\makeheadline`, `\makefootline`, and `\dosupereject` macros of PLAIN TeX; for those macros, and the original version of `\output`, see *The TeXbook*, p.364.

```

\output{\edmac@output}
\def\edmac@output{\shipout\vbox{\normal@pars
\box{\makeheadline\pagebody\makefootline}%
}%
\advancepageno
\ifnum\outputpenalty>-\@MM\else\dosupereject\fi}

\def\pagecontents{\page@start
\ifvoid\topins\else\unvbox\topins\fi
\dimen@=\dp\cc1v \unvbox\cc1v % open up \box255
\do@feet
\ifr@ggedbottom \kern-\dimen@ \vfil \fi}

```

`\do@feet` ships out all the footnotes. Standard `EDMAC` has only five feet, but there is nothing in principal to prevent you from creating an arachnoid or centipedal edition; straightforward modifications of `EDMAC` are all that's required. However, the myriapedal edition is ruled out by `TEX`'s limitations: the number of insertion classes is limited to 255.

```
\def\do@feet{%
 \ifvoid\footins\else
 \vskip\skip\footins
 \footnoterule
 \unvbox\footins
 \fi
 \ifvoid\Afootins\else
 \Afootstart{A}\Afootgroup{A}%
 \fi
 \ifvoid\Bfootins\else
 \Bfootstart{B}\Bfootgroup{B}%
 \fi
 \ifvoid\Cfootins\else
 \Cfootstart{C}\Cfootgroup{C}%
 \fi
 \ifvoid\Dfootins\else
 \Dfootstart{D}\Dfootgroup{D}%
 \fi
 \ifvoid\Efootins\else
 \Efootstart{E}\Efootgroup{E}%
 \fi}
```

For information (and so that I don't forget it), the code that now follows is part of the standard LaTeX output routine.

With luck we might only have to change `\@makecol` and `\@reinserts`. The kernel definition of these, and perhaps some other things, is:

```
\gdef \@makecol {%
 \ifvoid\footins
 \setbox\@outputbox \box\@cclv
 \else
 \setbox\@outputbox \vbox {%
 \boxmaxdepth \@maxdepth
 \@tempdima\dp\@cclv
 \unvbox \@cclv
 \vskip \skip\footins
 \color@begingroup
 \normalcolor
 \footnoterule
 \unvbox \footins
 \color@endgroup
 }%
 \fi}
```

```

\edef\@freelist{\@freelist\@midlist}%
\global \let \@midlist \@empty
\@combinefloats
\ifvbox\@kludgeins
 \@makespecialcolbox
\else
 \setbox\@outputbox \vbox to\@colht {%
 \@texttop
 \dimen@ \dp\@outputbox
 \unvbox\@outputbox
 \vskip -\dimen@
 \@textbottom
 }%
\fi
\global \maxdepth \@maxdepth
}

\gdef \@reinserts{%
 \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
 \ifvbox\@kludgeins\insert\@kludgeins{\unvbox\@kludgeins}\fi
}

```

Now we start actually changing things.

`\m@m@makecolfloats` These macros are defined in the memoir class and form part of the definition of `\m@m@makecoltext` `\@makecol`.

```

\m@m@makecolintro 2059 \providecommand{\m@m@makecolfloats}{%
2060 \edef\@freelist{\@freelist\@midlist}%
2061 \global \let \@midlist \@empty
2062 \@combinefloats}
2063 \providecommand{\m@m@makecoltext}{%
2064 \ifvbox\@kludgeins
2065 \@makespecialcolbox
2066 \else
2067 \setbox\@outputbox \vbox to\@colht {%
2068 \@texttop
2069 \dimen@ \dp\@outputbox
2070 \unvbox\@outputbox
2071 \vskip -\dimen@
2072 \@textbottom}%
2073 \fi}
2074 \providecommand{\m@m@makecolintro}{%
2075

```

`\l@d@makecol` This is a partitioned version of the ‘standard’ `\@makecol`, with the initial code put into another macro.

```

2076 \gdef\l@d@makecol{%
2077 \l@ddofootinsert
2078 \m@m@makecolfloats

```

```

2079 \m@m@makecoltext
2080 \global \maxdepth \@maxdepth}
2081

```

`\l@ddofootinsert` This macro essentially holds the initial portion of the kernel `\@makecol` code.

```

2082 \newcommand*{\l@ddofootinsert}{%
2083 %%% \page@start
2084 \ifvoid\footins
2085 \setbox\@outputbox \box\@cclv
2086 \else
2087 \setbox\@outputbox \vbox {%
2088 \boxmaxdepth \@maxdepth
2089 \@tempdima\dp\@cclv
2090 \unvbox \@cclv
2091 \vskip \skip\footins
2092 \color@begingroup
2093 \normalcolor
2094 \footnoterule
2095 \unvbox \footins
2096 \color@endgroup
2097 }%
2098 \fi

```

That's the end of the copy of the kernel code. We finally call a macro to handle all the additional EDMAC feet.

```

2099 \l@ddoxtrafeet
2100 }
2101

```

`\doxtrafeet` `\doxtrafeet` is the code extending `\@makecol` to cater for the extra ledmac feet. We have two classes of extra footnotes. We order the footnote inserts so that the regular footnotes are first, then class 1 (familiar footnotes) and finally class 2 (critical footnotes).

```

2102 \newcommand*{\l@ddoxtrafeet}{%
2103 \doxtrafeeti
2104 \doxtrafeetii}
2105

```

`\doxtrafeetii` `\doxtrafeetii` is the code extending `\@makecol` to cater for the extra critical feet (class 2 feet). NOTE: the code is likely to be 'featurefull'.

```

2106 \newcommand*{\doxtrafeetii}{%
2107 \setbox\@outputbox \vbox{%
2108 \unvbox\@outputbox
2109 \@opxtrafeetii}}

```

`\@opxtrafeetii` The extra critical feet to be added to the output.

```

2110 \newcommand*{\@opxtrafeetii}{%
2111 \ifvoid\Afootins\else\Afootstart{A}\Afootgroup{A}\fi
2112 \ifvoid\Bfootins\else\Bfootstart{B}\Bfootgroup{B}\fi

```



```

2113 \ifvoid\Cfootins\else\Cfootstart{C}\Cfootgroup{C}\fi
2114 \ifvoid\Dfootins\else\Dfootstart{D}\Dfootgroup{D}\fi
2115 \ifvoid\Efootins\else\Efootstart{E}\Efootgroup{E}\fi}
2116

```

`\l@ddodoreintrafeet` `\l@ddodoreintrafeet` is the code for catering for the extra footnotes within `\@reinserts`. The implementation may well have to change. We use the same classes and ordering as in `\l@ddoxtrafeet`.

```

2117 \newcommand*{\l@ddodoreintrafeet}{%
2118 \doreintrafeeti
2119 \doreintrafeetii}
2120

```

`\doreintrafeetii` `\doreintrafeetii` is the code for catering for the class 2 extra critical footnotes within `\@reinserts`. The implementation may well have to change.

```

2121 \newcommand*{\doreintrafeetii}{%
2122 \ifvoid\Afootins\else\insert\Afootins{\unvbox\Afootins}\fi
2123 \ifvoid\Bfootins\else\insert\Bfootins{\unvbox\Bfootins}\fi
2124 \ifvoid\Cfootins\else\insert\Cfootins{\unvbox\Cfootins}\fi
2125 \ifvoid\Dfootins\else\insert\Dfootins{\unvbox\Dfootins}\fi
2126 \ifvoid\Efootins\else\insert\Efootins{\unvbox\Efootins}\fi
2127 }
2128

```

`\l@d@reinserts` And here is the modified version of `\@reinserts`.

```

2129 \gdef \l@d@reinserts{%
2130 \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
2131 \l@ddodoreintrafeet
2132 \ifvbox\@kludgeins\insert\@kludgeins{\unvbox\@kludgeins}\fi
2133 }
2134

```

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.)

```

2135 \@ifclassloaded{memoir}{%
 memoir is loaded so we use memoir’s built in hooks.
2136 \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
2137 \g@addto@macro{\m@mdodoreintrafeet}{\l@ddodoreintrafeet}%
2138 }{%
 memoir has not been loaded, so redefine @makecol and @reinserts.
2139 \gdef\@makecol{\l@d@makecol}%
2140 \gdef\@reinserts{\l@d@reinserts}%
2141 }
2142

```

`\addfootins` Let's make it easier for an author to create a new series by providing this macro,  
`\addfootins{<letter>}`, to add the series to the several lists.

```
2143 \newcommand*{\addfootins}[1]{%
```

```
2144 \footnormal{#1}
```

Add it to the output.

```
2145 \g@addto@macro{\@opxtrafeetii}{%
```

```
2146 \ifvoid\@nameuse{#1footins}\else
```

```
2147 \@nameuse{#1footstart{#1}}\@nameuse{#1footgroup}{#1}\fi}
```

Add it to the reinsertions.

```
2148 \g@addto@macro{\doreinxtrafeetii}{%
```

```
2149 \ifvoid\@nameuse{#1footins}\else
```

```
2150 \insert\@nameuse{#1footins}{\unvbox\@nameuse{#1footins}}\fi}
```

Add it to minipages.

```
2151 \g@addto@macro{\l@dedbeginmini}{%
```

```
2152 \expandafter\let\csname #1footnote\endcsname = \@nameuse{mp#1footnote}}}
```

And at the end of a minipage.

```
2153 \g@addto@macro{\l@dedendmini}{%
```

```
2154 \ifvoid\@nameuse{mp#1footins}\else\@nameuse{mpfootgroup#1{#1}}\fi}
```

```
2155 }
```

```
2156
```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet. `\@led@extranofeet` is a hook for  
`\@led@extranofeet` handling further footnotes.

```
2157 \newif\if@led@nofoot
```

```
2158 \newcommand*{\@led@extranofeet}{}%
```

```
2159
```

```
2160 \ifclassloaded{memoir}{%
```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

`\@mem@extranofeet`

```
2161 \g@addto@macro{\@mem@extranofeet}{%
```

```
2162 \ifvoid\Afootins\else\@mem@nofootfalse\fi
```

```
2163 \ifvoid\Bfootins\else\@mem@nofootfalse\fi
```

```
2164 \ifvoid\Cfootins\else\@mem@nofootfalse\fi
```

```
2165 \ifvoid\Dfootins\else\@mem@nofootfalse\fi
```

```
2166 \ifvoid\Efootins\else\@mem@nofootfalse\fi
```

```
2167 \ifvoid\footinsA\else\@mem@nofootfalse\fi
```

```
2168 \ifvoid\footinsB\else\@mem@nofootfalse\fi
```

```
2169 \ifvoid\footinsC\else\@mem@nofootfalse\fi
```

```
2170 \@led@extranofeet}
```

```
2171 }{%
```

As memoir is not loaded we have to do it all here.

```

\@led@testifnofoot
\@doclearpage 2172 \newcommand*{\@led@testifnofoot}{%
2173 \@led@nofoottrue
2174 \ifvoid\footins\else\@led@nofootfalse\fi
2175 \ifvoid\Afootins\else\@led@nofootfalse\fi
2176 \ifvoid\Bfootins\else\@led@nofootfalse\fi
2177 \ifvoid\Cfootins\else\@led@nofootfalse\fi
2178 \ifvoid\Dfootins\else\@led@nofootfalse\fi
2179 \ifvoid\Efootins\else\@led@nofootfalse\fi
2180 \ifvoid\footinsA\else\@led@nofootfalse\fi
2181 \ifvoid\footinsB\else\@led@nofootfalse\fi
2182 \ifvoid\footinsC\else\@led@nofootfalse\fi
2183 \@led@extranofeet}
2184
2185 \renewcommand{\@doclearpage}{%
2186 \@led@testifnofoot
2187 \if@led@nofoot
2188 \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
2189 \setbox\@tempboxa\box\@cclv
2190 \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
2191 \global \let \@toplist \@empty
2192 \global \let \@botlist \@empty
2193 \global \@colroom \@colht
2194 \ifx \@currlist \@empty
2195 \else
2196 \@latexerr{Float(s) lost}\@ehb
2197 \global \let \@currlist \@empty
2198 \fi
2199 \@makefcolumn\@deferlist
2200 \@whiles\if@fcolmade \fi{\@opcol\@makefcolumn\@deferlist}%
2201 \if@twocolumn
2202 \if@firstcolumn
2203 \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%
2204 \global \let \@dbltoplist \@empty
2205 \global \@colht \textheight
2206 \beginingroup
2207 \@dblfloatplacement
2208 \@makefcolumn\@dbldeferlist
2209 \@whiles\if@fcolmade \fi{\@outputpage
2210 \@makefcolumn\@dbldeferlist}%
2211 \endgroup
2212 \else
2213 \vbox{}\clearpage
2214 \fi
2215 \fi
2216 \else
2217 \setbox\@cclv\vbox{\box\@cclv\vfil}%
2218 \l@makecol\@opcol
2219 \clearpage

```

```

2220 \fi}
2221 }
2222

```

## 24 Cross referencing

I have rewritten portions of the code in this section so that the LaTeX .aux file is used. This will also handle \included files.

Further, I have renamed some of the original EDMAC macros so that they do not clash with the LaTeX label/ref commands (EDMAC and LaTeX use very different mechanisms). In particular, the original EDMAC \label and \pageref have been renamed as \edlabel and \edpageref respectively.

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 saying \edpageref{foo}, or \lineref{foo} or \sublineref{foo}. These reference commands will produce, respectively, the page, line and sub-line 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 \lineref commands will refer to the latest occurrence of \label{foo}.

\labelref@list Set up a new list, \labelref@list, to hold the page, line and sub-line numbers for each label.

```

2223 \list@create{\labelref@list}

```

\zz@@@ A convenience macro to zero two labeling counters in one go.

```

2224 %% \newcommand*\zz@@@{000|000|000} % set three counters to zero in one go
2225 \newcommand*\zz@@@{000|000} % set two counters to zero in one go
2226

```

\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.<sup>26</sup>

This version of the original EDMAC \label uses \@bsphack and \@esphack to eliminate extra space problems and also the LaTeX write methods for the .aux file.

Jesse Billett<sup>27</sup> found that the original code could be off by several pages. This version, hopefully cures that, and also allows for non-arabic page numbering.

```

2227 \newcommand*\edlabel}[1]{\@bsphack
2228 \write\linenum@out{\string\@lab}%
2229 \ifx\labelref@list\empty

```

<sup>26</sup>The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

<sup>27</sup>(jdb43@cam.ac.uk) via the ctt thread 'ledmac cross referencing', 25 August 2003.

```

2230 \xdef\label@refs{\zz@@@}%
2231 \else
2232 \gl@p\labelref@list\to\label@refs
2233 \ifvmode
2234 \advancelabel@refs
2235 \fi
2236 \fi
2237 % \edef\next{\write\@aux{\string\l@dmake@labels\label@refs|{#1}}}%
2238 % \next}

```

Use code from the kernel `\label` command to write the correct page number (it seems possible that the original EDMAC's `\page@num` scheme might also have had problems in this area).

```

2239 \protected@write\@auxout{}%
2240 {\string\l@dmake@labels\space\thepage|\label@refs|{#1}}%
2241 \@esphack}
2242

```

```

\advancelabel@refs
\labelrefsparseline 2243 %In cases where \cs{edlabel} is the first element in a paragraph, we have a problem with line counts,
\labelrefsparsesubline 2244 %Hence, we need to test \cs{edlabel} if it occurs at the start of a paragraph. To do so, we use \cs{
2245 %We do so using \cs[advancelabel@refs] command.
2246 \newcommand{\advancelabel@refs}{%
2247 \newcount\line%
2248 \ledmac@warning{\label@refs}
2249 \line=\expandafter\labelrefsparseline\label@refs%
2250 \advance\line by 1%
2251 \ifsublines@%
2252 \newcount\subline%
2253 \subline=\labelrefsparsesubline\label@refs%
2254 \advance\subline by 1%
2255 \def\label@refs{\the\line|\the\subline}%
2256 \else%
2257 \def\label@refs{\the\line|0}%
2258 \fi%
2259 }
2260 \def\labelrefsparseline#1|#2{#1}
2261 \def\labelrefsparsesubline#1|#2{#2}

```

`\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).

```

2262 \newcommand*{\l@dmake@labels}{%
2263 \def\l@dmake@labels#1|#2|#3|#4{%
2264 \expandafter\ifx\csname the@label#4\endcsname \relax\else
2265 \led@warn@DuplicateLabel{#4}%

```

```

2266 \fi
2267 \expandafter\gdef\csname the@label#4\endcsname{#1|#2|#3}%
2268 \ignorespaces}
2269

```

LaTeX 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.

```

2270 \AtBeginDocument{%
2271 \def\l@dmake@labels#1|#2|#3|#4{%
2272 }
2273

```

**\@lab** The \@lab command, which appears in the \linenum@out file, appends the current values of page, line and sub-line to the \labelref@list. These values are defined by the earlier \@page, \@l, and the \sub@on and \sub@off commands appearing in the \linenum@out file.

LaTeX 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.

```

2274 \newcommand*{\@lab}{\xright@appenditem
2275 {\linenumrep{\line@num}}|}%
2276 \ifsublines@ \sublinenumrep{\subline@num}\else 0\fi}\to\labelref@list}
2277

```

**\edpageref** If the specified label exists, \edpageref gives its page number. 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 don't 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. LaTeX already defines a \pageref, so changing the name to \edpageref.

```

2278 \newcommand*{\edpageref}[1]{\l@dref@undefined{#1}\l@dgetref@num{1}{#1}}
2279 \newcommand*{\xpageref}[1]{\l@dgetref@num{1}{#1}}
2280

```

**\lineref** If the specified label exists, \lineref gives its line number.

```

\lineref 2281 \newcommand*{\lineref}[1]{\l@dref@undefined{#1}\l@dgetref@num{2}{#1}}
2282 \newcommand*{\xlineref}[1]{\l@dgetref@num{2}{#1}}
2283

```

**\sublineref** If the specified label exists, \sublineref gives its sub-line number.

```

\sublineref 2284 \newcommand*{\sublineref}[1]{\l@dref@undefined{#1}\l@dgetref@num{3}{#1}}
2285 \newcommand*{\xsublineref}[1]{\l@dgetref@num{3}{#1}}
2286

```

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.

```
2287 \newcommand*{\l@dref@undefined}[1]{%
2288 \expandafter\ifx\csname the@label#1\endcsname\relax
2289 \led@warn@RefUndefined{#1}%
2290 \fi}
2291
```

`\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) or sub-line (3) number. (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.

```
2292 \newcommand*{\l@dgetref@num}[2]{%
2293 \expandafter
2294 \ifx\csname the@label#2\endcsname \relax
2295 000%
2296 \else
2297 \expandafter\expandafter\expandafter
2298 \l@dlabel@parse\csname the@label#2\endcsname| #1%
2299 \fi}
2300
```

`\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, or 3) which defines which of the earlier three numbers to pick out. (It was earlier given as the first argument of `\l@dgetref@num`.)

```
2301 \newcommand*{\l@dlabel@parse}{%
2302 \def\l@dlabel@parse#1|#2|#3|#4{%
2303 \ifcase #4\relax
2304 \or #1%
2305 \or #2%
2306 \or #3%
2307 \fi}
2308
```

`\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 doesn’t, 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 `\label{elephant}`. The point of this is to be able to manufacture footnote line references to passages which can't be specified in the normal way as the first argument to `\critext` for one reason or another. Using `\xxref` in the second argument of `\critext` lets you set things up at least semi-automatically.

```
2309 \newcommand*{\xxref}[2]{%
2310 {\expandafter\ifx\csname the@label#1\endcsname
2311 \relax \expandafter\let\csname the@label#1\endcsname\zz@@@fi
2312 \expandafter\ifx\csname the@label#2\endcsname \relax
2313 \expandafter\let\csname the@label#2\endcsname\zz@@@fi
2314 \linenum{\csname the@label#1\endcsname|}%
2315 \csname the@label#2\endcsname}}
2316
```

`\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 say '`\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. LaTeX defines a `\makelabel` macro which is used in lists. I've changed the name to `\edmakelabel`.

```
2317 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\endcsname{#2}}
2318
```

(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 pp. 72 and 52), since `\xxref` makes a call to `\linenum` in order to do its work.)

## 25 Endnotes

```
\l@d@end Endnotes of all varieties are saved up in a file, typically named <jobname>.end.
\ifl@dend@ \l@d@end is the output stream number for this file, and \ifl@dend@ is a flag that's
\l@dend@true true when the file is open.
\l@dend@false 2319 \newwrite\l@d@end
 2320 \newif\ifl@dend@
```

```
\l@dend@open \l@dend@open and \l@dend@close are the macros that are used to open and close
\l@dend@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's no need to defer any writing to catch information
from the output routine.
```

```
2321 \newcommand{\l@dend@open}[1]{\global\l@dend@true\immediate\openout\l@d@end=#1\relax}
2322 \newcommand{\l@dend@close}{\global\l@dend@false\immediate\closeout\l@d@end}
2323
```



`\l@dend@stuff` `\l@dend@stuff` is used by `\beginnumbering` to do everything that's 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.

```

2324 \newcommand{\l@dend@stuff}{%
2325 \ifl@dend@relax\else
2326 \l@dend@open{\jobname.end}%
2327 \fi
2328 \immediate\write\l@d@end{\string\l@d@section{\the\section@num}}
2329 }

```

`\Aendnote` The following five macros each function to write one endnote to the `.end` file.  
`\Bendnote` Like the footnotes, these endnotes come in five series, A through E. We change  
`\Cendnote` `\newlinechar` so that in the file every space becomes the start of a new line; this  
`\Dendnote` generally ensures that a long note doesn't exceed restrictions on the length of lines  
`\Eendnote` in files.

```

2330 \newcommand*{\Aendnote}[1]{\{\newlinechar='40
2331 \immediate\write\l@d@end{\string\Aend%
2332 {\ifnumberedpar@l@d@nums\fi}%
2333 {\ifnumberedpar@tag\fi}{#1}}\ignorespaces}
2334 \newcommand*{\Bendnote}[1]{\{\newlinechar='40
2335 \immediate\write\l@d@end{\string\Bend%
2336 {\ifnumberedpar@l@d@nums\fi}%
2337 {\ifnumberedpar@tag\fi}{#1}}\ignorespaces}
2338 \newcommand*{\Cendnote}[1]{\{\newlinechar='40
2339 \immediate\write\l@d@end{\string\Cend%
2340 {\ifnumberedpar@l@d@nums\fi}%
2341 {\ifnumberedpar@tag\fi}{#1}}\ignorespaces}
2342 \newcommand*{\Dendnote}[1]{\{\newlinechar='40
2343 \immediate\write\l@d@end{\string\Dend%
2344 {\ifnumberedpar@l@d@nums\fi}%
2345 {\ifnumberedpar@tag\fi}{#1}}\ignorespaces}
2346 \newcommand*{\Eendnote}[1]{\{\newlinechar='40
2347 \immediate\write\l@d@end{\string\Eend%
2348 {\ifnumberedpar@l@d@nums\fi}%
2349 {\ifnumberedpar@tag\fi}{#1}}\ignorespaces}
2350 }

```

`\Aend` `\Aendnote` and the like write commands called `\Aend` and so on to the endnote  
`\Bend` file; these are analogous to the various `footfmt` commands above, and they take  
`\Cend` the same arguments. When we process this file, we'll want to pick out the notes of  
`\Dend` one series and ignore all the rest. To do that, we equate the `end` command for the  
`\Eend` series we want to `\endprint`, and leave the rest equated to `\@gobblethree`, which  
`\endprint` just skips over its three arguments.<sup>28</sup> The `\endprint` here is nearly identical in  
`\@gobblethree` its functioning to `\normalfootfmt`.

`\l@d@section` <sup>28</sup>Christophe Hebeisen ([christophe.hebeisen@a3.epfl.ch](mailto:christophe.hebeisen@a3.epfl.ch)) emailed on 2003/11/05 to say he had found that `\@gobblethree` was also defined in the `amsfonts` package.

The endnote file also contains `\l@d@section` commands, which supply the section numbers from the main text; standard `ledmac` does nothing with this information, but it's there if you want to write custom macros to do something with it.

```

2351 \def\endprint#1#2#3{{\notefontsetup{\notenumfont\printendlines#1}}%
2352 \enspace{\select@lemmafонт#1|#2}\enskip#3\par}}
2353 \providecommand*{\@gobblethree}[3]{}
2354 \let\Aend=\@gobblethree
2355 \let\Bend=\@gobblethree
2356 \let\Cend=\@gobblethree
2357 \let\Dend=\@gobblethree
2358 \let\Eend=\@gobblethree
2359 \let\l@d@section=\@gobble
2360

```

`\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.

```

2361 \newcommand*{\setprintendlines}[6]{%
2362 \l@d@pnumfalse \l@d@dashfalse
2363 \ifnum#4=#1 \else
2364 \l@d@pnumtrue
2365 \l@d@dashtrue
2366 \fi

```

We print the ending line number if: (1) we're printing the ending page number, or (2) it's different from the starting line number.

```

2367 \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
2368 \ifnum#2=#5 \else
2369 \l@d@elintrue
2370 \l@d@dashtrue
2371 \fi

```

We print the starting sub-line if it's nonzero.

```

2372 \l@d@ssubfalse
2373 \ifnum#3=0 \else
2374 \l@d@ssubtrue
2375 \fi

```

We print the ending sub-line if it's nonzero and: (1) it's different from the starting sub-line number, or (2) the ending line number is being printed.

```

2376 \l@d@eslfalse
2377 \ifnum#6=0 \else

```

```

2378 \ifnum#6=#3
2379 \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
2380 \else
2381 \l@d@esltrue
2382 \l@d@dashtrue
2383 \fi
2384 \fi}

```

`\printendlines` Now we're ready to print it all.

```

2385 \def\printendlines#1|#2|#3|#4|#5|#6|#7|{\begingroup
2386 \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%

```

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).

```

2387 \printnpnum{#1} \linenumrep{#2}%
2388 \ifl@d@ssub \fullstop \sublinenumrep{#3}\fi
2389 \ifl@d@dash \endashchar\fi
2390 \ifl@d@pnum \printnpnum{#4}\fi
2391 \ifl@d@elin \linenumrep{#5}\fi
2392 \ifl@d@esl \ifl@d@elin \fullstop\fi \sublinenumrep{#6}\fi
2393 \endgroup}
2394

```

`\printnpnum` A macro to print a page number in an endnote.

```

2395 \newcommand*{\printnpnum}[1]{p.#1} }
2396

```

`\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.

```

2397 \newcommand*{\doendnotes}[1]{\l@dend@close
2398 \begingroup
2399 \makeatletter
2400 \expandafter\let\csname #1end\endcsname=\endprint
2401 \input\jobname.end
2402 \endgroup}

```

`\noendnotes` You can say `\noendnotes` before the first `\beginnumbering` in your file if you aren't going to be using any of the endnote commands: this will suppress the creation of an `.end` file. If you do have some lingering endnote commands in your file, the notes will be written to your terminal and to the log file.

```

2403 \newcommand*{\noendnotes}{\global\let\l@dend@stuff=\relax
2404 \global\chardef\l@dend=16 }

```

## 26 Side notes

Regular `\marginpars` do not work inside numbered text — they don't produce any note but do put an extra unnumbered blank line into the text.

`\l@dold@xympar` Changing `\@xympar` a little at least ensures that `\marginpar` in numbered text do not disturb the flow.

```

2405 \let\l@dold@xympar\@xympar
2406 \renewcommand{\@xympar}{%
2407 \ifnumberedpar@
2408 \led@warn@NoMarginpars
2409 \@esphack
2410 \else
2411 \l@dold@xympar
2412 \fi}
2413
```

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).

```

\sidenotemargin
\l@dgetsidenote@margin
2414 \newcount\sidenote@margin
2415 \newcommand*{\sidenotemargin}[1]{%
2416 \l@dgetsidenote@margin{#1}%
2417 \ifnum\@l@dttempcntb>\m@ne
2418 \global\sidenote@margin=\@l@dttempcntb
2419 \fi}}
2420 \newcommand*{\l@dgetsidenote@margin}[1]{%
2421 \def\@tempa{#1}\def\@tempb{left}%
2422 \ifx\@tempa\@tempb
2423 \@l@dttempcntb \z@
2424 \else
2425 \def\@tempb{right}%
2426 \ifx\@tempa\@tempb
2427 \@l@dttempcntb \@ne
2428 \else
2429 \def\@tempb{outer}%
2430 \ifx\@tempa\@tempb
2431 \@l@dttempcntb \tw@
2432 \else
2433 \def\@tempb{inner}%
2434 \ifx\@tempa\@tempb
2435 \@l@dttempcntb \thr@@
2436 \else
2437 \led@warn@BadSidenotemargin
2438 \@l@dttempcntb \m@ne
2439 \fi
2440 \fi
2441 \fi
2442 \fi}
2443 \sidenotemargin{right}
2444
```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

`\l@drp@rbox`

```

2445 \newbox\l@dlp@rbox
2446 \newbox\l@drp@rbox
2447

```

\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 2448 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep 2449 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup 2450 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup 2451 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
2452 \newcommand*{\ledlsnotefontsetup}{\raggedleft\footnotesize}
2453 \newcommand*{\ledrsnotefontsetup}{\raggedright\footnotesize}
2454

```

\ledleftnote \ledleftnote{<text>} and \ledrightnote{<text>} are the user commands for left  
\ledrightnote and right sidenotes. \ledsidenote{<text>} is the command for a moveable sidenote.

```

\ledsidenote 2455 \newcommand*{\ledleftnote}[1]{\edtext{}{\l@dlsnote{#1}}}
2456 \newcommand*{\ledrightnote}[1]{\edtext{}{\l@drsnote{#1}}}
2457 \newcommand*{\ledsidenote}[1]{\edtext{}{\l@dcsnote{#1}}}
2458
2459

```

\l@dlsnote The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminis-  
\l@drsnote cent of the critical footnotes code.

```

\l@dcsnote 2460 \newif\ifrightrightnoteup
2461 \rightnoteuptrue
2462 \newcommand*{\l@dlsnote}[1]{%
2463 \newcommand{\content}{#1}%
2464 \ifnumberedpar@
2465 \xright@appenditem{\noexpand\l@dlsnote{\csexpandonce{content}}}%
2466 \to\inserts@list
2467 \global\advance\insert@count \@ne
2468 \fi\ignorespaces}
2469 \newcommand*{\l@drsnote}[1]{%
2470 \newcommand{\content}{#1}%
2471 \ifnumberedpar@
2472 \xright@appenditem{\noexpand\l@drsnote{\csexpandonce{content}}}%
2473 \to\inserts@list
2474 \global\advance\insert@count \@ne
2475 \fi\ignorespaces}
2476 \newcommand*{\l@dcsnote}[1]{%
2477 \newcommand{\content}{#1}%
2478 \ifnumberedpar@
2479 \xright@appenditem{\noexpand\l@dcsnote{\csexpandonce{content}}}%
2480 \to\inserts@list
2481 \global\advance\insert@count \@ne
2482 \fi\ignorespaces}
2483

```

```

\vl@dlsnote Put the left/right text into boxes, but just save the moveable text.
\vl@drsnote 2484 \newcommand*{\vl@dlsnote}[1]{\setl@dlp@rbox{#1}}
\vl@dcsnote 2485 \newcommand*{\vl@drsnote}[1]{\setl@drp@rbox{#1}}
2486 \newcommand*{\vl@dcsnote}[1]{\gdef\l@dcsnotetext{#1}}
2487

\setl@dlp@rbox \setl@dlp@rbox{<lednums>}{<tag>}{<text>} puts <text> into the \l@dlp@rbox box.
\setl@drp@rbox And similarly for the right side box. It is these boxes that finally get displayed in the
margins.

2488 \newcommand*{\setl@dlp@rbox}[1]{%
2489 {\parindent\z@\hspace=\ledlsnotewidth\ledlsnotefontsetup
2490 \global\setbox\l@dlp@rbox
2491 \ifleftnoteup
2492 =\vbox to\z@{\vss #1}%
2493 \else
2494 =\vbox to 0.70\baselineskip{\strut#1\vss}%
2495 \fi}}
2496 %% \global\setbox\l@dlp@rbox=\vbox to\z@{\vss #1}}% aligns on top line
2497 \newcommand*{\setl@drp@rbox}[1]{%
2498 {\parindent\z@\hspace=\ledrsnotewidth\ledrsnotefontsetup
2499 \global\setbox\l@drp@rbox
2500 \ifrightrightnoteup
2501 =\vbox to\z@{\vss #1}%
2502 \else
2503 =\vbox to 0.7\baselineskip{\strut#1\vss}%
2504 \fi}}
2505 \newif\ifleftnoteup
2506 \leftnoteuptrue

\savel@dcsnote Save the moveable note text in \l@dcsnotetext.
\l@dcsnotetext 2507 \newcommand*{\savel@dcsnote}[3]{%
2508 \gdef\l@dcsnotetext{#3}}
2509

\affixside@note This macro puts any moveable sidenote text into the left or right sidenote box, de-
pending on which margin it is meant to go in. It's a very much stripped down version
of \affixlin@num.

2510 \newcommand*{\affixside@note}{%
2511 \gdef\@templ@d{}}
2512 \ifx\@templ@d\l@dcsnotetext \else
2513 \if@twocolumn
2514 \if@firstcolumn
2515 \setl@dlp@rbox{\l@dcsnotetext}%
2516 \else
2517 \setl@drp@rbox{\l@dcsnotetext}%
2518 \fi
2519 \else
2520 \@l@dttempcntb=\sidenote@margin
2521 \ifnum\@l@dttempcntb>\@ne

```

```

2522 \advance\@l@dttempcntb by\page@num
2523 \fi
2524 \ifodd\@l@dttempcntb
2525 \setl@drp@rbox{\l@dcsnotetext}%
2526 \else
2527 \setl@dlp@rbox{\l@dcsnotetext}%
2528 \fi
2529 \fi
2530 \fi}
2531

```

## 27 Familiar footnotes

The original EDMAC provided the five series of critical footnotes, and LaTeX provides a single numbered footnote. The ledmac package uses the EDMAC mechanism to provide a few 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 seems such a useful ability that it is provided automatically by ledmac.

`\multiplefootnotemarker` These macros may have been defined by the memoir class, are provided by the footmisc package and perhaps by other footnote packages.

```

2532 \providecommand*\multiplefootnotemarker}{3sp}
2533 \providecommand*\multfootsep}{\normalfont,}
2534

```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the memoir class.

```

2535 \providecommand*\m@mmf@prepare}{%
2536 \kern-\multiplefootnotemarker
2537 \kern\multiplefootnotemarker\relax}

```

`\m@mmf@check` This may have been defined in the memoir class. If it recognises the last kern as \multiplefootnotemarker it typesets \multfootsep.

```

2538 \providecommand*\m@mmf@check}{%
2539 \ifdim\lastkern=\multiplefootnotemarker\relax
2540 \edef\x@sf{\the\spacefactor}%
2541 \unkern
2542 \multfootsep
2543 \spacefactor\x@sf\relax
2544 \fi}
2545

```

We have to modify \@footnotetext and \@footnotemark. However, if memoir is used the modifications have already been made.

```

2546 \ifclassloaded{memoir}{-}{%

```

`\@footnotetext` Add \m@mmf@prepare at the end of \@footnotetext.

```

2547 \let\l@dold@footnotetext\@footnotetext
2548 \renewcommand{\@footnotetext}[1]{%
2549 \l@dold@footnotetext{#1}%
2550 \m@mmf@prepare}

```

`\@footnotemark` Modify `\@footnotemark` to cater for adjacent `\footnotes`.

```

2551 \renewcommand*{\@footnotemark}{%
2552 \leavevmode
2553 \ifhmode
2554 \edef\x@sf{\the\spacefactor}%
2555 \m@mmf@check
2556 \nobreak
2557 \fi
2558 \@makefnmark
2559 \m@mmf@prepare
2560 \ifhmode\spacefactor\x@sf\fi
2561 \relax}

```

Finished the modifications for the non-memoir case.

```

2562 }
2563

```

`\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.

```

2564 \let\l@doldold@footnotetext\@footnotetext
2565 \renewcommand{\@footnotetext}[1]{%
2566 \ifnumberedpar@
2567 \edtext{}{\l@dbfnote{#1}}%
2568 \else
2569 \l@doldold@footnotetext{#1}%
2570 \fi}

```

`\l@dbfnote` `\l@dbfnote` adds the footnote to the insert list, and `\vl@dbfnote` calls the original  
`\vl@dbfnote` `\@footnotetext`.

```

2571 \newcommand{\l@dbfnote}[1]{%
2572 \ifnumberedpar@
2573 \xright@appenditem{\noexpand\vl@dbfnote{#1}}{\@thefnmark}}%
2574 \to\inserts@list
2575 \global\advance\insert@count \@ne
2576 \fi\ignorespaces}
2577 \newcommand{\vl@dbfnote}[2]{%
2578 \def\@thefnmark{#2}%
2579 \l@doldold@footnotetext{#1}}
2580

```

Now we can get on with providing the extra series of numbered footnotes. The general naming convention is to add an uppercase letter, denoting the series, at the



end of macro names (the EDMAC series have an uppercase letter at the start of macro names).

First we'll give all the code for the A series, then the much more limited code for defining additional series.

## 27.1 The A series footnotes

`\footnoteA` `\footnoteA{<text>}` is the user level command.

```
2581 \newcommand{\footnoteA}[1]{%
2582 \stepcounter{footnoteA}%
2583 \protected@xdef\thefnmarkA{\thefootnoteA}%
2584 \@footnotemarkA
2585 \vfootnoteA{A}{#1}\m@mmf@prepare}
2586
```

`\footinsA` The insert for the A series.

```
2587 \newinsert\footinsA
```

`\c@footnoteA` The A series counter.

```
\thefootnoteA 2588 \newcounter{footnoteA}
2589 \renewcommand{\thefootnoteA}{\arabic{footnoteA}}
2590
```

`\footfootmarkA` This macro typesets the A series marker at the start of the footnote text (where it appears at the foot of the page).

```
2591 \newcommand*{\footfootmarkA}{\thefootnoteA}
2592
```

`\mpfootnoteA` The extras for minipages.

```
\mpfootinsA 2593 \newcommand{\mpfootnoteA}[1]{%
2594 \stepcounter{footnoteA}%
2595 \protected@xdef\thefnmarkA{\thefootnoteA}%
2596 \@footnotemarkA
2597 \mpvfootnoteA{A}{#1}\m@mmf@prepare}
2598 \newinsert\mpfootinsA
2599
```

We have to specify the default footnote style for the A series. This is done later.

That completes the specific macros that have to be specified for the A series. Similar ones are required for any other series.

## 27.2 Footnote formats

Some of the code for the various formats is remarkably similar to that in section 22.3.

The following macros generally set things up for the 'standard' footnote format.

```

\prebodyfootmark Two convenience macros for use by \...@footnotemark... macros.
\postbodyfootmark 2600 \newcommand*{\prebodyfootmark}{%
2601 \leavevmode
2602 \ifhmode
2603 \edef\@x@sf{\the\spacefactor}%
2604 \m@mmf@check
2605 \nobreak
2606 \fi}
2607 \newcommand{\postbodyfootmark}{%
2608 \m@mmf@prepare
2609 \ifhmode\spacefactor\@x@sf\fi\relax}
2610

\normal@footnotemarkX \normal@footnotemarkX{<series>} sets up the typesetting of the marker at the point
where the footnote is called for.

2611 \newcommand*{\normal@footnotemarkX}[1]{%
2612 \prebodyfootmark
2613 \@nameuse{bodyfootmark#1}%
2614 \postbodyfootmark}
2615

\normalbodyfootmarkX The \normalbodyfootmarkX{<series>} really typesets the in-text marker. The style
is the normal superscript.

2616 \newcommand*{\normalbodyfootmarkX}[1]{%
2617 \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}}

\normalvfootnoteX \normalvfootnoteX{<series>}{<text>} does the \insert for the <series> and calls
the series' \footfmt... to format the <text>.

2618 \newcommand*{\normalvfootnoteX}[2]{%
2619 \insert\@nameuse{footins#1}\bgroup
2620 \notefontsetup
2621 \footplitskips
2622 \spaceskip=\z@skip \xspaceskip=\z@skip
2623 \@nameuse{footfmt#1}{#1}{#2}\egroup}
2624

\mpnormalvfootnoteX The minipage version.

2625 \newcommand*{\mpnormalvfootnoteX}[2]{%
2626 \global\setbox\@nameuse{mpfootins#1}\vbox{%
2627 \unvbox\@nameuse{mpfootins#1}
2628 \notefontsetup
2629 \hsize\columnwidth
2630 \@parboxrestore
2631 \color@begingroup
2632 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
2633

\normalfootfmtX \normalfootfmtX{<series>}{<text>} typesets the footnote text, prepended by the
marker.
```

```

2634 \newcommand*{\normalfootfmtX}[2]{%
2635 \ledsetnormalparstuff
2636 {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2637 #2\strut\par}}
2638

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

2639 \newcommand*{\normalfootfootmarkX}[1]{%
2640 \textsuperscript{\@nameuse{thefnmark#1}}}
2641

```

`\normalfootstartX` `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

2642 \newcommand*{\normalfootstartX}[1]{%
2643 \vskip\skip\@nameuse{footins#1}%
2644 \leftskip=\z@
2645 \rightskip=\z@
2646 \@nameuse{footnoterule#1}}
2647

```

`\normalfootnoteruleX` The rule drawn before the footnote series group.

```

2648 \let\normalfootnoteruleX=\footnoterule
2649

```

`\normalfootgroupX` `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

2650 \newcommand*{\normalfootgroupX}[1]{%
2651 \unvbox\@nameuse{footins#1}}
2652

```

`\mpnormalfootgroupX` The minipage version.

```

2653 \newcommand*{\mpnormalfootgroupX}[1]{%
2654 \vskip\skip\@nameuse{mpfootins#1}
2655 \normalcolor
2656 \@nameuse{footnoterule#1}
2657 \unvbox\@nameuse{mpfootins#1}}
2658

```

`\normalbfnoteX`

```

2659 \newcommand{\normalbfnoteX}[2]{%
2660 \ifnumberedpar@
2661 \xright@appenditem{\noexpand\vbfnoteX{#1}{#2}{\@nameuse{thefootnote#1}}}%
2662 \to\inserts@list
2663 \global\advance\insert@count \@ne
2664 \fi\ignorespaces}
2665

```

`\vbfnoteX`

```
2666 \newcommand{\vbfnoteX}[3]{%
2667 \@namedef{@thefnmark#1}{#3}%
2668 \@nameuse{regvfootnote#1}{#1}{#2}}
2669
```

`\vnumfootnoteX`

```
2670 \newcommand{\vnumfootnoteX}[2]{%
2671 \ifnumberedpar@
2672 \edtext{}{\normalbfnoteX{#1}{#2}}%
2673 \else
2674 \@nameuse{regvfootnote#1}{#1}{#2}%
2675 \fi}
2676
```

`\footnormalX` `\footnormalX{<series>}` initialises the settings for the *<series>* footnotes. This should always be called for each series.

```
2677 \newcommand*{\footnormalX}[1]{%
2678 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
2679 \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
2680 \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
2681 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
2682 \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
2683 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
2684 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
2685 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
2686 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
2687 \count\csname footins#1\endcsname=1000
2688 \dimen\csname footins#1\endcsname=\ledfootinsdim
2689 \skip\csname footins#1\endcsname=1.2em \@plus .6em \@minus .6em
```

Aditions for minipages.

```
2690 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
2691 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
2692 \count\csname mpfootins#1\endcsname=1000
2693 % \dimen\csname mpfootins#1\endcsname=0.8\vsizer
2694 \dimen\csname mpfootins#1\endcsname=\ledfootinsdim
2695 \skip\csname mpfootins#1\endcsname=1.2em \@plus .6em \@minus .6em
2696 }
2697
```

### 27.2.1 Two column footnotes

The following macros set footnotes in two columns. It is assumed that the length of each footnote is less than the column width.

`\foottwocolX` `\foottwocolX{<series>}`

```
2698 \newcommand*{\foottwocolX}[1]{%
2699 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
```

```

2700 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
2701 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
2702 \twocolfootsetupX{#1}
2703 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
2704 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
2705 \mptwocolfootsetupX{#1}}
2706

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX 2707 \newcommand*{\twocolfootsetupX}[1]{%
2708 \count\csname footins#1\endcsname 500
2709 \multiply\dimen\csname footins#1\endcsname by \tw@}
2710 \newcommand*{\mptwocolfootsetupX}[1]{%
2711 \count\csname mpfootins#1\endcsname 500
2712 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
2713

\twocolvfootnoteX \twocolvfootnoteX{<series>}
2714 \newcommand*{\twocolvfootnoteX}[2]{%
2715 \insert\csname footins#1\endcsname\bgroup
2716 \notefontsetup
2717 \footssplitskips
2718 \spaceskip=\z@skip \xspaceskip=\z@skip
2719 \@nameuse{footfmt#1}-{#1}-{#2}\egroup}
2720

\twocolfootfmtX \twocolfootfmtX{<series>}
2721 \newcommand*{\twocolfootfmtX}[2]{%
2722 \normal@pars
2723 \hsize .45\hsize
2724 \parindent=\z@
2725 %%% \parfillskip=0pt \@plus 1fil
2726 \tolerance=5000\relax
2727 \raggedright
2728 \leavevmode
2729 {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2730 #2\strut\par}\allowbreak}
2731

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX 2732 \newcommand*{\twocolfootgroupX}[1]{\notefontsetup
2733 \splittopskip=\ht\strutbox
2734 \expandafter
2735 \rigidbalance\csname footins#1\endcsname \tw@ \splittopskip}}
2736 \newcommand*{\mptwocolfootgroupX}[1]{%
2737 \vskip\skip\@nameuse{mpfootins#1}
2738 \normalcolor
2739 \@nameuse{footnoterule#1}
2740 \splittopskip=\ht\strutbox

```

```

2741 \expandafter
2742 \rigidbalance\csname mpfootins#1\endcsname \tw@ \splittopskip}}
2743

```

### 27.2.2 Three column footnotes

The following macros set footnotes in three columns. It is assumed that the length of each footnote is less than the column width.

```

\footthreecolX \footthreecolX{<series>}
2744 \newcommand*{\footthreecolX}[1]{%
2745 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
2746 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
2747 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
2748 \threecolfootsetupX{#1}
2749 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
2750 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
2751 \mpthreecolfootsetupX{#1}}
2752

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX 2753 \newcommand*{\threecolfootsetupX}[1]{%
2754 \count\csname footins#1\endcsname 333
2755 \multiply\dimen\csname footins#1\endcsname by \thr@@}
2756 \newcommand*{\mpthreecolfootsetupX}[1]{%
2757 \count\csname mpfootins#1\endcsname 333
2758 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
2759

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
2760 \newcommand*{\threecolvfootnoteX}[2]{%
2761 \insert\csname footins#1\endcsname\bgroup
2762 \notefontsetup
2763 \footsplitskips
2764 \@nameuse{footfmt#1}{#1}{#2}\egroup}
2765

\threecolfootfmtX \threecolfootfmtX{<series>}
2766 \newcommand*{\threecolfootfmtX}[2]{%
2767 \normal@pars
2768 \hsize .3\hsize
2769 \parindent=\z@
2770 %%% \parfillskip=0pt \@plus 1fil
2771 \tolerance=5000\relax
2772 \raggedright
2773 \leavevmode
2774 {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2775 #2\strut\par}\allowbreak}
2776

```

```

\threecolfootgroupX \threecolfootgroupX{<series>}
\mpthreecolfootgroupX 2777 \newcommand*{\threecolfootgroupX}[1]{\notefontsetup
2778 \splittopskip=\ht\strutbox
2779 \expandafter
2780 \rigidbalance\csname footins#1\endcsname \thr@@ \splittopskip}}
2781 \newcommand*{\mpthreecolfootgroupX}[1]{\%
2782 \vskip\skip\@nameuse{mpfootins#1}
2783 \normalcolor
2784 \@nameuse{footnoterule#1}
2785 \splittopskip=\ht\strutbox
2786 \expandafter
2787 \rigidbalance\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
2788

```

### 27.2.3 Paragraphed footnotes

The following macros set footnotes as one paragraph.

```

\footparagraphX \footparagraphX{<series>}
2789 \newcommand*{\footparagraphX}[1]{\%
2790 \expandafter\newcount\csname prevpage#1@num\endcsname
2791 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
2792 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
2793 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
2794 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
2795 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
2796 \count\csname footins#1\endcsname=1000
2797 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
2798 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
2799 \count\csname mpfootins#1\endcsname=1000
2800 \para@footsetupX{#1}}
2801

\para@footsetupX \para@footsetupX{<series>}
2802 \newcommand*{\para@footsetupX}[1]{\notefontsetup
2803 \dimen0=\baselineskip
2804 \multiply\dimen0 by 1024
2805 \divide\dimen0 by \hsize \multiply\dimen0 by \footfudgefiddle\relax
2806 \expandafter
2807 \xdef\csname footfudgefactor#1\endcsname{\%
2808 \expandafter\strip@pt\dimen0 }}
2809

\parafootstartX \parafootstartX{<series>}
2810 \newcommand*{\parafootstartX}[1]{\%
2811 \vskip\skip\@nameuse{footins#1}%
2812 \leftskip=\z@
2813 \rightskip=\z@
2814 \parindent=\z@

```

```

2815 \vskip\skip\@nameuse{footins#1}%
2816 \@nameuse{footnoterule#1}}
2817
\para@vfootnoteX \para@vfootnoteX{<series>}{<text>}
\mppara@vfootnoteX 2818 \newcommand*{\para@vfootnoteX}[2]{%
2819 \insert\csname footins#1\endcsname
2820 \bgroup
2821 \notefontsetup
2822 \footsplitskips
2823 \setbox0=\vbox{\hsize=\maxdimen
2824 \noindent\@nameuse{footfmt#1}{#1}{#2}}%
2825 \setbox0=\hbox{\unvbox0[#1]}%
2826 \dp0=\z@
2827 \ht0=\csname footfudgefactor#1\endcsname\wd0
2828 \box0
2829 \penalty0
2830 \egroup}
2831 \newcommand*{\mppara@vfootnoteX}[2]{%
2832 \global\setbox\@nameuse{mpfootins#1}\vbox{%
2833 \unvbox\@nameuse{mpfootins#1}
2834 \notefontsetup
2835 \footsplitskips
2836 \setbox0=\vbox{\hsize=\maxdimen
2837 \noindent\color@begingroup\@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
2838 \setbox0=\hbox{\unvbox0[#1]}%
2839 \dp0=\z@
2840 \ht0=\csname footfudgefactor#1\endcsname\wd0
2841 \box0
2842 \penalty0}}
2843
\parafootfmtX \parafootfmtX{<series>}
2844 \newcommand*{\parafootfmtX}[2]{%
2845 \insertparafootsep{#1}%
2846 \ledsetnormalparstuff
2847 {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2848 #2\penalty-10}}
2849
\para@footgroupX \para@footgroupX{<series>}
\mppara@footgroupX 2850 \newcommand*{\para@footgroupX}[1]{%
2851 \unvbox\csname footins#1\endcsname
2852 \makehboxofhboxes
2853 \setbox0=\hbox{\unhbox0 \removehboxes}%
2854 \notefontsetup
2855 \noindent\unhbox0\par}
2856 \newcommand*{\mppara@footgroupX}[1]{%
2857 \vskip\skip\@nameuse{mpfootins#1}

```



```

2858 \normalcolor
2859 \@nameuse{footnoterule#1}
2860 \unvbox\csname mpfootins#1\endcsname
2861 \makehboxofhboxes
2862 \setbox0=\hbox{\unhbox0 \removehboxes}%
2863 \notefontsetup
2864 \noindent\unhbox0\par}}
2865

```

## 27.3 Other series footnotes

Other series, such as B, are provided here.

`\footnoteB` `\footnoteB{<text>}` is the user command for a series B footnote.

```

2866 \newcommand{\footnoteB}[1]{%
2867 \stepcounter{footnoteB}%
2868 \protected@xdef\thefnmarkB{\thefootnoteB}%
2869 \@footnotemarkB
2870 \vfootnoteB{B}{#1}\m@mmf@prepare}
2871

```

```

\c@footnoteB
\thefootnoteB 2872 \newcounter{footnoteB}
2873 \renewcommand{\thefootnoteB}{\arabic{footnoteB}}
2874

```

```

\footinsB
2875 \newinsert\footinsB
2876

```

`\mpfootnoteB` The extras for minipages.

```

\mpfootinsB 2877 \newcommand{\mpfootnoteB}[1]{%
2878 \stepcounter{footnoteB}%
2879 \protected@xdef\thefnmarkB{\thefootnoteB}%
2880 \@footnotemarkB
2881 \mpvfootnoteB{B}{#1}\m@mmf@prepare}
2882 \newinsert\mpfootinsB
2883

```

`\footnoteC` `\footnoteC{<text>}` is the user command for a series C footnote.

```

2884 \newcommand{\footnoteC}[1]{%
2885 \stepcounter{footnoteC}%
2886 \protected@xdef\thefnmarkC{\thefootnoteC}%
2887 \@footnotemarkC
2888 \vfootnoteC{C}{#1}\m@mmf@prepare}

```

```

\c@footnoteC
\thefootnoteC 2889 \newcounter{footnoteC}
\footinsC 2890 \renewcommand{\thefootnoteC}{\arabic{footnoteC}}

```

```
2891 \newinsert\footinsC
2892
```

\mpfootnoteC The extras for minipages.

```
\mpfootinsC 2893 \newcommand{\mpfootnoteC}[1]{%
2894 \stepcounter{footnoteC}%
2895 \protected@xdef\@thefnmarkC{\thefootnoteC}%
2896 \@footnotemarkC
2897 \mpvfootnoteC{C}{#1}\m@mmf@prepare}
2898 \newinsert\mpfootinsC
2899
```

Don't forget to initialise the series.

```
2900 \footnormalX{A}
2901 \footnormalX{B}
2902 \footnormalX{C}
2903
```

\doxtrafeeti We have to add all the new kinds of familiar footnotes to the output routine. These  
\doreinxtrafeeti are the class 1 feet.

```
2904 \newcommand*{\doxtrafeeti}{%
2905 \setbox\@outputbox \vbox{%
2906 \unvbox\@outputbox
2907 \ifvoid\footinsA\else\footstartA{A}\footgroupA{A}\fi
2908 \ifvoid\footinsB\else\footstartB{B}\footgroupB{B}\fi
2909 \ifvoid\footinsC\else\footstartC{C}\footgroupC{C}\fi
2910 }}
2911
2912 \newcommand{\doreinxtrafeeti}{%
2913 \ifvoid\footinsA\else\insert\footinsA{\unvbox\footinsA}\fi
2914 \ifvoid\footinsB\else\insert\footinsB{\unvbox\footinsB}\fi
2915 \ifvoid\footinsC\else\insert\footinsC{\unvbox\footinsC}\fi
2916 }
2917
```

\addfootinsX Make life just a little easier for those who want additional series of class 1 footnotes.

```
2918 \newcommand*{\addfootinsX}[1]{%
2919 \footnormalX{#1}%
2920 \g@addto@macro{\doxtrafeeti}{%
2921 \setbox\@outputbox \vbox{%
2922 \unvbox\@outputbox
2923 \ifvoid\@nameuse{footins#1}\else
2924 \@nameuse{footstart#1}{#1}\@nameuse{footgroup#1}{#1}\fi}%
2925 \g@addto@macro{\doreinxtrafeeti}{%
2926 \ifvoid\@nameuse{footins#1}\else
2927 \insert\@nameuse{footins#1}{\unvbox\@nameuse{footins#1}}\fi}%
2928 \g@addto@macro{\l@dfambeginmini}{%
2929 \expandafter\expandafter\expandafter\let\expandafter\expandafter
2930 \csname footnote#1\endcsname \csname mpfootnote#1\endcsname}%

```

```

2931 \g@addto@macro{\l@dfamendmini}{%
2932 \ifvoid\@nameuse{mpfootins#1}\else\@nameuse{mpfootgroup#1}{#1}\fi}%
2933 }

```

## 28 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'll arrange this so that additional series can be easily added.

`\l@dfheetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage` They can be extended  
`\l@dfheetendmini` to handle other things if necessary.

```

2934 \newcommand*{\l@dfheetbeginmini}{\l@dedbeginmini\l@dfambeginmini}
2935 \newcommand*{\l@dfheetendmini}{\l@dedendmini\l@dfamendmini}
2936

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.  
`\l@dedendmini` They can be extended to cater for additional series.

```

2937 \newcommand*{\l@dedbeginmini}{%
2938 \let\Afootnote=\mpAfootnote%
2939 \let\Bfootnote=\mpBfootnote%
2940 \let\Cfootnote=\mpCfootnote%
2941 \let\Dfootnote=\mpDfootnote%
2942 \let\Efootnote=\mpEfootnote%
2943 \newcommand*{\l@dedendmini}{%
2944 \ifvoid\mpAfootins\else\mpAfootgroup{A}\fi%
2945 \ifvoid\mpBfootins\else\mpBfootgroup{B}\fi%
2946 \ifvoid\mpCfootins\else\mpCfootgroup{C}\fi%
2947 \ifvoid\mpDfootins\else\mpDfootgroup{D}\fi%
2948 \ifvoid\mpEfootins\else\mpEfootgroup{E}\fi%
2949

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.  
`\l@dfamendmini` They can be extended to cater for additional series.

```

2950 \newcommand*{\l@dfambeginmini}{%
2951 \let\footnoteA=\mpfootnoteA%
2952 \let\footnoteB=\mpfootnoteB%
2953 \let\footnoteC=\mpfootnoteC%
2954 \newcommand*{\l@dfamendmini}{%
2955 \ifvoid\mpfootinsA\else\mpfootgroupA{A}\fi%
2956 \ifvoid\mpfootinsB\else\mpfootgroupB{B}\fi%
2957 \ifvoid\mpfootinsC\else\mpfootgroupC{C}\fi%
2958

```

`\@iiminipage` This is our extended form of the kernel `\@iiminipage` defined in `ltboxes.dtx`.

```

2959 \def\@iiminipage#1#2[#3]#4{%
2960 \leavevmode

```

```

2961 \pboxswfalse
2962 \setlength\@tempdima{#4}%
2963 \def\@mpargs{#{1}-{#2}[#{3}]#{4}}%
2964 \setbox\@tempboxa\vbox\bgroup
2965 \color@begingroup
2966 \hsize\@tempdima
2967 \textwidth\hsize \columnwidth\hsize
2968 \@parboxrestore
2969 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
2970 \let\@footnotetext\@mpfootnotetext

```

The next line is our addition to the original.

```

2971 \l@dfeetbeginmini% added
2972 \let\@listdepth\@mplistdepth \@mplistdepth\z@
2973 \@minipagerestore
2974 \@setminipage}
2975

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

2976 \def\endminipage{%
2977 \par
2978 \unskip
2979 \ifvoid\@mpfootins\else
2980 \l@dunboxmpfoot
2981 \fi

```

The next line is our addition to the original.

```

2982 \l@dfeetendmini% added
2983 \@minipagefalse
2984 \color@endgroup
2985 \egroup
2986 \expandafter\@iiiparbox\@mpargs{\unvbox\@tempboxa}}
2987

```

`\l@dunboxmpfoot`

```

2988 \newcommand*{\l@dunboxmpfoot}{%
2989 \vskip\skip\@mpfootins
2990 \normalcolor
2991 \footnoterule
2992 \unvbox\@mpfootins}
2993

```

`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.

```

2994 \newenvironment{ledgroup}{%
2995 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
2996 \let\@footnotetext\@mpfootnotetext
2997 \l@dfeetbeginmini%
2998 }{%

```

```

2999 \par
3000 \unskip
3001 \ifvoid\@mpfootins\else
3002 \l@dunboxmpfoot
3003 \fi
3004 \l@dfeetendmini%
3005 }
3006

```

**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.

```

3007 \newenvironment{ledgroupsize}[2][1]{%

```

Set the various text measures.

```

3008 \hsize #2\relax
3009 %% \textwidth #2\relax
3010 %% \columnwidth #2\relax

```

Initialize fills for centering.

```

3011 \let\ledllfill\hfil
3012 \let\ledrlfill\hfil
3013 \def\@tempa{#1}\def\@tempb{1}%

```

Left adjusted numbered lines

```

3014 \ifx\@tempa\@tempb
3015 \let\ledllfill\relax
3016 \else
3017 \def\@tempb{r}%
3018 \ifx\@tempa\@tempb

```

Right adjusted numbered lines

```

3019 \let\ledrlfill\relax
3020 \fi
3021 \fi

```

Set up the footnoting.

```

3022 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
3023 \let\@footnotetext\@mpfootnotetext
3024 \l@dfeetbeginmini%
3025 }{%
3026 \par
3027 \unskip
3028 \ifvoid\@mpfootins\else
3029 \l@dunboxmpfoot
3030 \fi
3031 \l@dfeetendmini%
3032 }
3033

```

## 29 Indexing

Here's some code for indexing using page & line numbers.

`\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 3034 \newcommand{\pagelinesep}{-}
 3035 \newcommand{\edindexlab}{\&\&}
 3036 \newcounter{labidx}
 3037 \setcounter{labidx}{0}
 3038
```

`\doedindexlabel` This macro sets an `\edlabel`.

```
3039 \newcommand{\doedindexlabel}{\stepcounter{labidx}%
3040 \edlabel{\edindexlab\thelabidx}}
3041
```

`\thepageline` This macro makes up the page/line number combo from the label/ref.

```
3042 \newcommand{\thepageline}{%
3043 \thepage\pagelinesep\lineref{\edindexlab\thelabidx}}
3044
```

The memoir class provides more flexible indexing than the standard classes. We need different code if the memoir class is being used.

```
3045 \ifclassloaded{memoir}{%
 memoir is being used.
```

`\makeindex` Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do  
`\edindex` nothing. In this case `\edindex` has an optional argument. We use the hook provided  
in memoir v1.61.

```
3046 \g@addto@macro{\makememindexhook}{%
3047 \def\edindex{\@bsphack%
3048 \ifnextchar [{\l@d@index}{\l@d@index[\jobname]}}
3049 \newcommand{\edindex}[2][\jobname]{\@bsphack\@esphack}
```

`\l@d@index` `\l@d@index[file]` is the first stage of `\edindex`, handling the idx file. This is virtually a verbatim copy of memoir's `\@index`, the change being calling `\l@dwrindexm@m` instead of `\@wrindexm@m`.

```
3050 \def\l@d@index[#1]{%
3051 \@ifundefined{#1idxfile}%
3052 {\ifreportnoidxfile
3053 \led@warn@NoIndexFile{#1}%
3054 \fi
3055 \begingroup
3056 \@sanitize
3057 \@nowrindex}%
3058 {\def\@idxfile{#1}%
3059 \doedindexlabel
```

```

3060 \beginngroup
3061 \@sanitize
3062 \l@d@wrindexm@m}}

```

\l@d@wrindexm@m \l@d@wrindexm@m{item} writes the idx file name and the indexed item to the \l@d@wrindexhyp aux file. These are almost verbatim copies of memoir's \@wrindexm@m and \@wrindexhyp.

```

3063 \newcommand{\l@d@wrindexm@m}[1]{\l@d@wrindexhyp#1||\}
3064 \def\l@d@wrindexhyp#1|#2|#3\{\%
3065 \ifshowindexmark\@showidx{#1}\fi
3066 \ifx\#2\%
3067 \protected@write\@auxout{\%
3068 {\string\@wrindexm@m{\@idxfile}{#1|hyperpage}{\thepage}}}%
3069 \else
3070 \def\Hy@temp@a{#2}%
3071 \ifx\Hy@temp@a\HyInd@ParenLeft
3072 \protected@write\@auxout{\%
3073 {\string\@wrindexm@m{\@idxfile}{#1|#2hyperpage}{\thepage}}}%
3074 \else
3075 \protected@write\@auxout{\%
3076 {\string\@wrindexm@m{\@idxfile}{#1|#2}{\thepage}}}%
3077 \fi
3078 \fi
3079 \endgroup
3080 \@esphack}

```

That finishes the memoir-specific code.

```
3081 }{\%
```

memoir is not being used, which makes life somewhat simpler.

\makeindex Need to add the definition of \edindex to \makeindex, and initialise \edindex to \edindex do nothing.

```

3082 \g@addto@macro{\makeindex}{\%
3083 \def\edindex{\@bsphack
3084 \doedindexlabel
3085 \beginngroup
3086 \@sanitize
3087 \@wredindex}}
3088 \newcommand{\edindex}[1]{\@bsphack\@esphack}

```

\@wredindex Write the index information to the idx file.

```

3089 \newcommand{\@wredindex}[1]{\%
3090 \protected@write\@indexfile{\%
3091 {\string\indexentry{#1}{\thepage}}}%
3092 \endgroup
3093 \@esphack}

```

That finishes the non-memoir index code.

```

3094 }
3095

```

`\l@d@wrindexhyp` If the `hyperref` package is not loaded, it doesn't make sense to clutter up the index with hyperreffering things.

```

3096 \AtBeginDocument{\@ifpackageloaded{hyperref}{}{%
3097 \def\l@d@wrindexhyp#1||\{\%
3098 \ifshowindexmark\@showidx{#1}\fi
3099 \protected@write\@auxout{}%
3100 {\string\@wrindexm@{\@idxfile}{#1}{\thepageline}}%
3101 \endgroup
3102 \@esphack}}
3103

```

### 30 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.

```

3104 \newtoks\@emptytoks
3105

```

The rest is from `amsmath`.

`\l@denbody` A token register to contain the body.

```

3106 \newtoks\l@denbody
3107

```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```

3108 \newcommand{\addtol@denbody}[1]{%
3109 \global\l@denbody\expandafter{\the\l@denbody#1}}
3110

```

`\l@dcollect@body` The macro `\l@dcollect@body` starts the scan for the `\end{...}` 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`.

```

3111 \newcommand{\l@dcollect@body}[1]{%
3112 \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
3113 \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}%
3114 \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
3115 \begingroup
3116 \expandafter\let\csname\@currenvir\endcsname\l@dcollect@body
3117 \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
3118 \processl@denbody}
3119

```



`\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.

```
3120 \def\l@dpush@begins#1\begin#2{%
3121 \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
3122
```

`\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.

```
3123 \def\l@dcollect@@body#1\end#2{%
3124 \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
3125 \expandafter\@gobble\l@dbegin@stack}%
3126 \ifx\@empty\l@dbegin@stack
3127 \endgroup
3128 \@checkend{#2}%
3129 \addtol@denvbody{#1}%
3130 \else
3131 \addtol@denvbody{#1\end{#2}}%
3132 \fi
3133 \processl@denvbody % A little tricky! Note the grouping
3134 }
3135
```

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>

```

## 31 Verse

This is principally Wayne Sullivan's code and commentary from EDSTANZA [Sul92].

The macro `\hangingsymbol` is used to insert a symbol on each hanging of verses. For example, in french typographie the symbol is '['. We obtain it by the next code:

```
\renewcommand{\hangingsymbol}{[,]}
```

The `\ifinstanza` boolean is used to be sure that we are in a stanza part.

```

\hangingsymbol
\ifinstanza 3136 \newcommand*{\hangingsymbol}{%
3137 \newif\ifinstanza
3138 \instanzafalse

\inserthangingsymbol The boolean \ifinserthangingsymbol is set to TRUE when \@lock is greater
\ifinserthangingsymbol than 1, i.e. when we are not in the first line of a verse. The switch of
\ifinserthangingsymbol \ifinserthangingsymbol is made in \do@line before the printing of line but after
the line number calculation.
3139 \newif\ifinserthangingsymbol
3140 \newcommand{\inserthangingsymbol}{%
3141 \ifinserthangingsymbol%
3142 \ifinstanza%
3143 \hfill\hangingsymbol%

```

```

3144 \fi%
3145 \fi%
3146 }

```

`\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.

```

3147 \newcommand*{\ampersand}{\char'\&}
3148

```

`\stanza@count` Before we can define the main macros we need to save and reset some category codes. To save the current values we use `\next` and `\body` from the `\loop` macro.

```

3149 \chardef\body=\catcode'\@
3150 \catcode'\@=11
3151 \chardef\next=\catcode'\&
3152 \catcode'\&=\active
3153

```

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.

```

3154 \newcount\stanza@count
3155 \newlength{\stanzaindentbase}
3156 \setlength{\stanzaindentbase}{20pt}
3157

```

`\strip@szacnt` The indentations of stanza lines are non-negative integer multiples of the unit called `\stanzaindentbase`. To make it easier for the user to specify these numbers, some list macros are defined. These take numerical values in a list separated by commas and assign the values to special control sequences using `\mathchardef`. Though this does limit the range from 0 to 32767, it should suffice for most applications, including *penalties*, which will be discussed below.

```

3158 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
3159 \newcommand*{\setstanzavalues}[2]{\def\@tempa{#2,|}%
3160 \stanza@count\z@
3161 \def\next{\expandafter\strip@szacnt\@tempa
3162 \ifx\@tempb\empty\let\next\relax\else
3163 \expandafter\mathchardef\csname #1@\number\stanza@count
3164 \@endcsname\@tempb\relax
3165 \advance\stanza@count\@ne\fi\next}%
3166 \next}
3167

```

`\setstanzaindent` In the original `\setstanzavalues{sza}{...}` had to be called to set the indents, and similarly `\setstanzavalues{szp}{...}` to set the penalties. These two macros are a convenience to give the user one less thing to worry about (misspelling the first argument). Since version 0.13, the `stanzaindent`s repetition counter can be used when the indentation is repeated every *n* verses. The `\managestanza@modulo`

is a command which modifies the counter stanza@modulo. The command adds 1 to stanza@modulo, but if stanza@modulo is equal to the stanzaindentrepetition counter, the command restarts it.

```

3168 \newcommand*\setstanzaindents}[1]{\setstanzavalues{sza}{#1}}
3169 \newcommand*\setstanzapenalties}[1]{\setstanzavalues{szp}{#1}}
3170
3171 \newcounter{stanzaindentsrepetition}
3172 \newcount\stanza@modulo
3173
3174 \newcommand*\managestanza@modulo}[0]{
3175 \advance\stanza@modulo\@ne
3176 \ifnum\stanza@modulo>\value{stanzaindentsrepetition}
3177 \stanza@modulo\@ne
3178 \fi
3179 }
```

\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 line and starts a numbered paragraph—each line is treated as a paragraph.

\sza@penalty \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.

```

3180 \def\stanza@line{
3181 \ifnum\value{stanzaindentsrepetition}=0
3182 \parindent=\csname sza@\number\stanza@count
3183 @\endcsname\stanzaindentbase
3184 \else
3185 \managestanza@modulo
3186 \parindent=\csname sza@\number\stanza@modulo
3187 @\endcsname\stanzaindentbase
3188 \fi
3189 \pstart\stanza@hang\ignorespaces}
3190 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
3191 \hangindent\expandafter
3192 \noexpand\csname sza@0@\endcsname\stanzaindentbase
3193 \hangafter\@ne}
3194 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
3195 \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
3196 \penalty\fi\count@}
```

\startstanzahook Now we have the components of the \stanza macro, which appears at the start of a group of lines. This macro initializes the count and checks to see if hanging indentation and penalties are to be included. Hanging indentation suspends the line count, so that the enumeration is by verse line rather than by print line. If the print line count is desired, invoke \let\startlock=\relax and do the same for \endlock. Here and above we have used \xdef to make the stored macros

\endstanzaextra

\stanza

take up a bit less space, but it also makes them more obscure to the reader. Lines of the stanza are delimited by ampersands &. The last line of the stanza must end with \&. For convenience the macro \endstanzaextra is included. The user may use this to add vertical space or penalties between stanzas.

As a further convenience, the macro \startstanzahook is called at the beginning of a stanza. This can be defined to do something useful.

```

3197 \let\startstanzahook\relax
3198 \let\endstanzaextra\relax
3199 \xdef\stanza{\noexpand\instanzatrue\expandafter
3200 \begingroup\startstanzahook%
3201 \catcode'\&\active\global\stanza@count\@ne\stanza@modulo\@ne
3202 \noexpand\ifnum\expandafter\noexpand
3203 \csname sza@0@\endcsname=\z@\let\noexpand\stanza@hang\relax
3204 \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
3205 \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
3206 \expandafter\noexpand\csname szp@0@\endcsname=\z@
3207 \let\noexpand\sza@penalty\relax\noexpand\fi \def\noexpand&{%
3208 \noexpand\endlock\noexpand\pend\noexpand\sza@penalty\global
3209 \advance\stanza@count\@ne\noexpand\stanza@line}\def\noexpand
3210 &f\noexpand\endlock\noexpand\pend\endgroup\noexpand\instanzafalse\expandafter\er
3211 \noexpand\stanza@line}
3212

```

\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.

```

3213 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
3214 \hskip -#1\llap{#2}\hskip #1\ignorespaces}
3215

```

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.

```

3216 \catcode'\&=\next
3217 \catcode'\@=\body
3218 %% \let\ampersand=\&
3219 \setstanzavalues{szp}{0}
3220

```

## 32 Arrays and tables

This is based on the work by Herbert Breger in developing tabmac.tex.

```

%%
% This is file tabmac.tex 1.0.
% You find here macros for tabular structures compatible with

```

```

% Edmac (authored by Lavagnino/Wujastyk). The use of the macros is
% explained in German language in file tabanlei.dvi. The macros were
% developed for Edmac 2.3, but this file has been adjusted to Edmac 3.16.
%
% ATTENTION: This file uses some Edmac control sequences (like
% \text, \Afootnote etc.) and redefines \morenoexpands. If you yourself
% redefined some Edmac control sequences, be careful: some adjustments
% might be necessary.
% October 1996
%
% My kind thanks to Nora G^deke for valuable support. Any hints and
% comments are welcome, please contact Herbert Breger,
% Leibniz-Archiv, Waterloostr. 8, D -- 30169 Hannover, Germany
% Tel.: 511 - 1267 327
%%
%%

```

The original tabmac.tex file was void of comments or any explanatory text other than the above notice. The algorithm is Breger's. I have made some cosmetic changes to the original code and reimplemented some things so they are more LaTeX-like. All the commentary is mine, as are any mistakes or errors.

`\l@dtabnoexpands` An extended and modified version of the original additional no expansions..

```

3221 \newcommand*{\l@dtabnoexpands}{%
3222 \def\ss{\noexpand\ss}%
3223 \def\"##1{\noexpand\"##1}%
3224 \def\'##1{\noexpand\'##1}%
3225 \def\'##1{\noexpand\'##1}%
3226 \def\^##1{\noexpand\^##1}%
3227 \def\phantom##1{\noexpand}%
3228 \def\hphantom##1{\noexpand\hphantom{##1}}%
3229 \def\underbrace##1{\noexpand\underbrace{##1}}%
3230 \def\dots{\noexpand\dots}%
3231 \let\rtab=0%
3232 \let\ctab=0%
3233 \let\ltab=0%
3234 \let\rtabtext=0%
3235 \let\ltabtext=0%
3236 \let\ctabtext=0%
3237 \let\edbeforetab=0%
3238 \let\edaftertab=0%
3239 \let\edatab=0%
3240 \let\edatabell=0%
3241 \let\edatleft=0%
3242 \let\edatright=0%
3243 \let\edvertline=0%
3244 \let\edvertdots=0%
3245 \let\edrowfill=0%
3246 }

```

3247

\l@dampcount \l@dampcount is a counter for the & column dividers and \l@dcolcount is a counter  
 \l@dcolcount for the columns. These were \Undcount and \stellencount respectively.

```
3248 \newcount\l@dampcount
3249 \l@dampcount=1\relax
3250 \newcount\l@dcolcount
3251 \l@dcolcount=0\relax
3252
```

\hilfsbox Some (temporary) helper items.

```
\hilfsskip 3253 \newbox\hilfsbox
\Hilfsbox 3254 \newskip\hilfsskip
\hilfscount 3255 \newbox\Hilfsbox
3256 \newcount\hilfscount
3257
```

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).

```
3258 \newdimen\dcoli
3259 \newdimen\dcolii
3260 \newdimen\dcoliii
3261 \newdimen\dcoliv
3262 \newdimen\dcolv
3263 \newdimen\dcolvi
3264 \newdimen\dcolvii
3265 \newdimen\dcolviii
3266 \newdimen\dcolix
3267 \newdimen\dcolx
3268 \newdimen\dcolxi
3269 \newdimen\dcolxii
3270 \newdimen\dcolxiii
3271 \newdimen\dcolxiv
3272 \newdimen\dcolxv
3273 \newdimen\dcolxvi
3274 \newdimen\dcolxvii
3275 \newdimen\dcolxviii
3276 \newdimen\dcolxix
3277 \newdimen\dcolxx
3278 \newdimen\dcolxxi
3279 \newdimen\dcolxxii
3280 \newdimen\dcolxxiii
3281 \newdimen\dcolxxiv
3282 \newdimen\dcolxxv
3283 \newdimen\dcolxxvi
3284 \newdimen\dcolxxvii
3285 \newdimen\dcolxxviii
3286 \newdimen\dcolxxix
```



```

3287 \newdimen\dcollxxx
3288 \newdimen\dcollerr % added for error handling
3289

```

`\l@dcollwidth` This is a cunning way of storing the columnwidths indexed by the column number `\l@dcollcount`, like an array. (was `\Dimenzuordnung`)

```

3290 \newcommand{\l@dcollwidth}{\ifcase \the\l@dcollcount \dcoli %???
3291 \or \dcoli \or \dcolii \or \dcoliii
3292 \or \dcoliv \or \dcolv \or \dcolvi
3293 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
3294 \or \dcolxi \or \dcolxii \or \dcolxiii
3295 \or \dcolxiv \or \dcolxv \or \dcolxvi
3296 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
3297 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
3298 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
3299 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
3300 \else \dcollerr \fi}
3301

```

`\step1@dcollcount` This increments the column counter, and issues an error message if it is too large.

```

3302 \newcommand*{\step1@dcollcount}{\advance\l@dcollcount\@ne
3303 \ifnum\l@dcollcount>30\relax
3304 \led@err@TooManyColumns
3305 \fi}
3306

```

`\l@dsetmaxcollwidth` Sets the column width to the maximum value seen so far. (was `\dimenzuordnung`)

```

3307 \newcommand{\l@dsetmaxcollwidth}{%
3308 \ifdim\l@dcollwidth < \wd\hilfsbox
3309 \l@dcollwidth = \wd\hilfsbox
3310 \else \relax \fi}
3311

```

`\EDTEXT` We need to be able to modify the `\edtext` and `\critext` macros and also restore `\xedtext` their original definitions.

```

\CRITEXT 3312 \let\EDTEXT=\edtext
\xcritext 3313 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
3314 \let\CRITEXT=\critext
3315 \long\def\xcritext #1#2/{\CRITEXT{#1}{#2}/}

```

`\EDLABEL` We need to be able to modify and restore the `\edlabel` macro.

```

\xedlabel 3316 \let\EDLABEL=\edlabel
3317 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}

```

`\EDINDEX` Macros supporting modification and restoration of `\edindex`.

```

\xedindex 3318 \let\EDINDEX=\edindex
\nulledindex 3319 \ifl@dmemoir
3320 \newcommand{\xedindex}{\@bsphack%
3321 \ifnextchar [{\l@d@index}{\l@d@index[\jobname]}}

```

```

3322 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
3323 \else
3324 \newcommand{\xedindex}{\@bsphack%
3325 \doedindexlabel
3326 \begingroup
3327 \@sanitize
3328 \@wredindex}
3329 \newcommand{\nulledindex}[1]{\@bsphack\@esphack}
3330 \fi
3331

```

\A@@footnote We need to be able to modify ledmac's footnote macros and restore their original  
\B@@footnote definitions. There are five of these.

```

\C@@footnote 3332 \let\A@@footnote=\Afootnote
\D@@footnote 3333 \let\B@@footnote=\Bfootnote
\E@@footnote 3334 \let\C@@footnote=\Cfootnote
3335 \let\D@@footnote=\Dfootnote
3336 \let\E@@footnote=\Efootnote

```

\@line@num Macro supporting restoration of \linenum.

```

3337 \let\@line@num=\linenum

```

\l@dgobbledarg \l@dgobbledarg replaces its delineated argument by \relax (was \verschwinden).

\l@dgobblearg \l@dgobblearg{<arg>} replaces its argument by \relax.

```

3338 \def\l@dgobbledarg #1/{\relax}
3339 \newcommand*\l@dgobblearg[1]{\relax}
3340

```

\Relax

```

\NEXT 3341 \let\Relax=\relax

```

\@hilfs@count 3342 \let\NEXT=\next

```

3343 \newcount\@hilfs@count
3344

```

\measuremcell Measure (recursively) the width required for a math cell. (was \messen)

```

3345 \def\measuremcell #1&{%
3346 \ifx #1\ \ifnum\l@dc@lcount=0\let\NEXT\relax%
3347 \else\l@dcheckcols%
3348 \l@dc@lcount=0%
3349 \let\NEXT\measuremcell%
3350 \fi%
3351 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
3352 \step\l@dc@lcount%
3353 \l@dsetmaxcolwidth%
3354 \let\NEXT\measuremcell%
3355 \fi\NEXT}
3356

```

`\measuretcell` Measure (recursively) the width required for a text cell. (was `\messentext`)

```

3357 \def\measuretcell #1{%
3358 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
3359 \else\l@dcheckcols%
3360 \l@dcolcount=0%
3361 \let\NEXT\measuretcell%
3362 \fi%
3363 \else\setbox\hilfsbox=\hbox{#1}%
3364 \step\l@dcolcount%
3365 \l@dsetmaxcolwidth%
3366 \let\NEXT\measuretcell%
3367 \fi\NEXT}
3368
```

`\measuremrow` Measure (recursively) the width required for a math row. (was `\Messen`)

```

3369 \def\measuremrow #1\{%
3370 \ifx #1&\let\NEXT\relax%
3371 \else\measuremcell #1&\&\&%
3372 \let\NEXT\measuremrow%
3373 \fi\NEXT}

```

`\measuretrow` Measure (recursively) the width required for a text row. (was `\Messentext`)

```

3374 \def\measuretrow #1\{%
3375 \ifx #1&\let\NEXT\relax%
3376 \else\measuretcell #1&\&\&%
3377 \let\NEXT\measuretrow%
3378 \fi\NEXT}
3379
```

`\edtabcolsep` The length `\edtabcolsep` controls the distance between columns. (was `\abstand`)

```

3380 \newskip\edtabcolsep
3381 \global\edtabcolsep=10pt
3382
```

`\NEXT`

```

\Next 3383 \let\NEXT\relax
3384 \let\Next=\next

```

`\variab`

```

3385 \newcommand{\variab}{\relax}
3386
```

`\l@dcheckcols` Check that the number of columns is consistent. (was `\tabfehlermeldung`)

```

3387 \newcommand*{\l@dcheckcols}{%
3388 \ifnum\l@dcolcount=1\relax
3389 \else
3390 \ifnum\l@dampcount=1\relax
3391 \else
3392 \ifnum\l@dcolcount=\l@dampcount\relax

```

```

3393 \else
3394 \l@d@err@UnequalColumns
3395 \fi
3396 \fi
3397 \l@dampcount=\l@dcolcount
3398 \fi}
3399

```

\l@dmodforcritext Modify and restore various macros for when \critext is used.

```

\l@drestoreforcritext 3400 \newcommand{\l@dmodforcritext}{%
3401 \let\critext\relax%
3402 \let\Afootnote\l@dgobbledarg%
3403 \let\Bfootnote\l@dgobbledarg%
3404 \let\Cfootnote\l@dgobbledarg%
3405 \let\Dfootnote\l@dgobbledarg%
3406 \let\Efootnote\l@dgobbledarg%
3407 \let\edindex\nulledindex%
3408 \let\linenum@gobble}
3409 \newcommand{\l@drestoreforcritext}{%
3410 \def\Afootnote##1##2/{\A@@footnote{##1}{##2}}%
3411 \def\Bfootnote##1##2/{\B@@footnote{##1}{##2}}%
3412 \def\Cfootnote##1##2/{\C@@footnote{##1}{##2}}%
3413 \def\Dfootnote##1##2/{\D@@footnote{##1}{##2}}%
3414 \def\Efootnote##1##2/{\E@@footnote{##1}{##2}}%
3415 \let\edindex\xedndex}
3416

```

\l@dmodforedtext Modify and restore various macros for when \edtext is used.

```

\l@drestoreforedtext 3417 \newcommand{\l@dmodforedtext}{%
3418 \let\edtext\relax
3419 \let\Afootnote\l@dgonblearg
3420 \let\Bfootnote\l@dgonblearg
3421 \let\Cfootnote\l@dgonblearg
3422 \let\Dfootnote\l@dgonblearg
3423 \let\Efootnote\l@dgonblearg
3424 \let\edindex\nulledindex
3425 \let\linenum@gobble}
3426 \newcommand{\l@drestoreforedtext}{%
3427 \def\Afootnote##1{\A@@footnote{##1}}%
3428 \def\Bfootnote##1{\B@@footnote{##1}}%
3429 \def\Cfootnote##1{\C@@footnote{##1}}%
3430 \def\Dfootnote##1{\D@@footnote{##1}}%
3431 \def\Efootnote##1{\E@@footnote{##1}}%
3432 \let\edindex\xedndex}
3433

```

\l@dnullfills Nullify and restore some column fillers, etc.

```

\l@drestorefills 3434 \newcommand{\l@dnullfills}{%
3435 \def\edlabel##1{}%

```

```

3436 \def\edrowfill##1##2##3{%
3437 }
3438 \newcommand{\l@restorefills}{%
3439 \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
3440 }
3441

```

The original definition of `\rverteilen` and friends (`'verteilen'` is approximately `'distribute'`) was along the lines:

```

\def\rverteilen #1&{\def\label##1{%
 \ifx #1! \ifnum\l@dcolcount=0\removelastskip
 \let\Next\relax%
 \else\l@dcolcount=0%
 \let\Next=\rverteilen%
 \fi%
\else%
 \footnoteverschw%
 \step1@dcolcount%
 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
 \let\critext=\xcritext\let\Dfootnote=\D@@footnote
 \let\Afootnote=\A@@footnote\let\Bfootnote=\B@@footnote
 \let\Cfootnote=\C@@footnote\let\linenum=\@line@num%
 \hilfsskip=\Dimenzuordnung%
 \advance\hilfsskip by -\wd\hilfsbox
 \def\label##1{\xlabel{##1}}%
 \hskip\hilfsskip$\displaystyle{#1}$%
 \hskip\edtabcolsep%
 \let\Next=\rverteilen%
\fi\Next}

```

where the lines

```

\let\critext=\xcritext\let\Dfootnote=\D@@footnote
\let\Afootnote=\A@@footnote\let\Bfootnote=\B@@footnote
\let\Cfootnote=\C@@footnote\let\linenum=\@line@num%
\hilfsskip=\Dimenzuordnung%
\advance\hilfsskip by -\wd\hilfsbox
\def\label##1{\xlabel{##1}}%

```

were common across the several `*verteilen*` macros, and also

```

\def\footnoteverschw{%
 \let\critext\relax
 \let\Afootnote=\verschwinden
 \let\Bfootnote=\verschwinden
 \let\Cfootnote=\verschwinden
 \let\Dfootnote=\verschwinden
 \let\linenum=\@gobble}

```

`\letsforverteilen` Gathers some lets and other code that is common to the `*verteilen*` macros.

```

3442 \newcommand{\letsforverteilen}{%
3443 \let\critext\xcritext
3444 \let\edtext\xedtext
3445 \let\edindex\xedindex
3446 \let\Afootnote\A@@footnote
3447 \let\Bfootnote\B@@footnote
3448 \let\Cfootnote\C@@footnote
3449 \let\Dfootnote\D@@footnote
3450 \let\Efootnote\E@@footnote
3451 \let\linenum\@line@num
3452 \hilfsskip=\l@dcwidth%
3453 \advance\hilfsskip by -\wd\hilfsbox
3454 \def\edlabel##1{\xedlabel{##1}}
3455
```

`\setmcellright` Typeset (recursively) cells of display math right justified. (was `\rverteilen`)

```

3456 \def\setmcellright #1{\def\edlabel##1{%
3457 \let\edindex\nulledindex
3458 \ifx #1\\ \ifnum\l@dcwidth=0%\removelastskip
3459 \let\Next\relax%
3460 \else\l@dcwidth=0%
3461 \let\Next=\setmcellright%
3462 \fi%
3463 \else%
3464 \disablel@dtabfeet%
3465 \step1@dcwidth%
3466 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
3467 \letsforverteilen%
3468 \hskip\hilfsskip$\displaystyle{#1}$%
3469 \hskip\edtabcolsep%
3470 \let\Next=\setmcellright%
3471 \fi\Next}
3472
```

`\settcclright` Typeset (recursively) cells of text right justified. (was `\rverteilentext`)

```

3473 \def\settcclright #1{\def\edlabel##1{%
3474 \let\edindex\nulledindex
3475 \ifx #1\\ \ifnum\l@dcwidth=0%\removelastskip
3476 \let\Next\relax%
3477 \else\l@dcwidth=0%
3478 \let\Next=\settcclright%
3479 \fi%
3480 \else%
3481 \disablel@dtabfeet%
3482 \step1@dcwidth%
3483 \setbox\hilfsbox=\hbox{#1}%
3484 \letsforverteilen%
3485 \hskip\hilfsskip#1%

```

```

3486 \hskip\edtabcolsep%
3487 \let\Next=\settcellright%
3488 \fi\Next}

```

`\setmcellleft` Typeset (recursively) cells of display math left justified. (was `\lverteilen`)

```

3489 \def\setmcellleft #1&{\def\edlabel##1{}%
3490 \let\edindex\nulledindex
3491 \ifx #1\\ \ifnum\l@dc@count=0 \let\Next\relax%
3492 \else\l@dc@count=0%
3493 \let\Next=\setmcellleft%
3494 \fi%
3495 \else \disablel@dtabfeet%
3496 \step1@dc@count%
3497 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
3498 \letsforverteilen
3499 $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
3500 \let\Next=\setmcellleft%
3501 \fi\Next}
3502

```

`\settcellleft` Typeset (recursively) cells of text left justified. (was `\lverteilentext`)

```

3503 \def\settcellleft #1&{\def\edlabel##1{}%
3504 \let\edindex\nulledindex
3505 \ifx #1\\ \ifnum\l@dc@count=0 \let\Next\relax%
3506 \else\l@dc@count=0%
3507 \let\Next=\settcellleft%
3508 \fi%
3509 \else \disablel@dtabfeet%
3510 \step1@dc@count%
3511 \setbox\hilfsbox=\hbox{#1}%
3512 \letsforverteilen
3513 #1\hskip\hilfsskip\hskip\edtabcolsep%
3514 \let\Next=\settcellleft%
3515 \fi\Next}

```

`\setmcellcenter` Typeset (recursively) cells of display math centered. (was `\zverteilen`)

```

3516 \def\setmcellcenter #1&{\def\edlabel##1{}%
3517 \let\edindex\nulledindex
3518 \ifx #1\\ \ifnum\l@dc@count=0\let\Next\relax%
3519 \else\l@dc@count=0%
3520 \let\Next=\setmcellcenter%
3521 \fi%
3522 \else \disablel@dtabfeet%
3523 \step1@dc@count%
3524 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
3525 \letsforverteilen%
3526 \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
3527 \hskip\edtabcolsep%
3528 \let\Next=\setmcellcenter%

```

```

3529 \fi\Next}
3530

```

`\settccllcenter` Typeset (recursively) cells of text centered. (new)

```

3531 \def\settccllcenter #1{\def\edlabel##1{%
3532 \let\edindex\nulledindex
3533 \ifx #1\ \ifnum\l@dcolcount=0 \let\Next\relax%
3534 \else\l@dcolcount=0%
3535 \let\Next=\settccllcenter%
3536 \fi%
3537 \else \disablel@dtabfeet%
3538 \stepl@dcolcount%
3539 \setbox\hilfsbox=\hbox{#1}%
3540 \letsforverteilen%
3541 \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
3542 \hskip\edtabcolsep%
3543 \let\Next=\settccllcenter%
3544 \fi\Next}
3545

```

`\NEXT`

```

3546 \let\NEXT=\relax
3547

```

`\setmrowright` Typeset (recursively) rows of right justified math. (was `\rsetzen`)

```

3548 \def\setmrowright #1\{%
3549 \ifx #1& \let\NEXT\relax
3550 \else \centerline{\setmcellright #1&\&\&}
3551 \let\NEXT=\setmrowright
3552 \fi\NEXT}

```

`\settroright` Typeset (recursively) rows of right justified text. (was `\rsetzentext`)

```

3553 \def\settroright #1\{%
3554 \ifx #1& \let\NEXT\relax
3555 \else \centerline{\settccllright #1&\&\&}
3556 \let\NEXT=\settroright
3557 \fi\NEXT}
3558

```

`\setmrowleft` Typeset (recursively) rows of left justified math. (was `\lsetzen`)

```

3559 \def\setmrowleft #1\{%
3560 \ifx #1& \let\NEXT\relax
3561 \else \centerline{\setmcellleft #1&\&\&}
3562 \let\NEXT=\setmrowleft
3563 \fi\NEXT}

```

`\settrorleft` Typeset (recursively) rows of left justified text. (was `\lsetzentext`)

```

3564 \def\settrorleft #1\{%
3565 \ifx #1& \let\NEXT\relax

```



```

3566 \else \centerline{\settcellleft #1&\\&\\&}
3567 \let\NEXT=\settrrowleft
3568 \fi\NEXT}
3569

```

`\setmrowcenter` Typeset (recursively) rows of centered math. (was `\zsetzen`)

```

3570 \def\setmrowcenter #1\\{%
3571 \ifx #1& \let\NEXT\relax%
3572 \else \centerline{\setmcellcenter #1&\\&\\&}
3573 \let\NEXT=\setmrowcenter
3574 \fi\NEXT}

```

`\settrrowcenter` Typeset (recursively) rows of centered text. (new)

```

3575 \def\settrrowcenter #1\\{%
3576 \ifx #1& \let\NEXT\relax
3577 \else \centerline{\settcellcenter #1&\\&\\&}
3578 \let\NEXT=\settrrowcenter
3579 \fi\NEXT}
3580

```

`\nullsetzen` (was `\nullsetzen`)

```

3581 \newcommand{\nullsetzen}{%
3582 \step1@dcolcount%
3583 \l@dcolwidth=0pt%
3584 \ifnum\l@dcolcount=30\let\NEXT\relax%
3585 \l@dcolcount=0\relax
3586 \else\let\NEXT\nullsetzen%
3587 \fi\NEXT}
3588

```

`\edatleft` `\edatleft[ $\langle\math\rangle\{\langle symbol\rangle\}\{\langle len\rangle\}$ ]` (combination and generalisation of original `\Seklam` and `\Seklamgl`). Left  $\langle symbol\rangle$ ,  $2\langle len\rangle$  high with prepended  $\langle math\rangle$  vertically centered.

```

3589 \newcommand{\edatleft}[3][\@empty]{%
3590 \ifx#1\@empty
3591 \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
3592 depth 0pt \right. $\hss}\vfil}
3593 \else
3594 \vbox to 4pt{\vss\hbox{$#1\left#2\vrule width0pt height #3
3595 depth 0pt \right. $\}\vfil}
3596 \fi}

```

`\edatright` `\edatright[ $\langle\math\rangle\{\langle symbol\rangle\}\{\langle len\rangle\}$ ]` (combination and generalisation of original `\seklam` and `\seklamgl`). Right  $\langle symbol\rangle$ ,  $2\langle len\rangle$  high with appended  $\langle math\rangle$  vertically centered.

```

3597 \newcommand{\edatright}[3][\@empty]{%
3598 \ifx#1\@empty
3599 \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
3600 depth 0pt \right#2 $\hss}\vfil}

```

```

3601 \else
3602 \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
3603 depth 0pt \right#2 #1 $\}\vfil}
3604 \fi}
3605

```

`\edvertline` `\edvertline{<len>}` vertical line `<len>` high. (was `\sestrich`)

```

3606 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
3607

```

`\edvertdots` `\edvertdots{<len>}` vertical dotted line `<len>` high. (was `\sepunkte`)

```

3608 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
3609 {\cleaders\hbox{$\m@th\hbox{.}\vbox to 0.5em{ }$}\vfil}}}
3610

```

I don't know if this is relevant here, and I haven't tried it, but the following appeared on CTT.

From: mdw@nsict.org (Mark Wooding)  
 Newsgroups: comp.text.tex  
 Subject: Re: Dotted line  
 Date: 13 Aug 2003 13:51:14 GMT

Alexis Eisenhofer <alexis@eisenhofer.de> wrote:  
 > Can anyone provide me with the LaTeX command for a vertical dotted line?

How dotted? Here's the basic rune.

```

\newbox\linedotbox
\setbox\linedotbox=\vbox{...}
\leaders\copy\linedotbox\vskip2in

```

For just dots, this works:

```

\setbox\linedotbox=\vbox{\hbox{\normalfont.}\kern2pt}

```

For dashes, something like

```

\setbox\linedotbox=\vbox{\leaders\vrule\vskip2pt\vskip2pt}

```

is what you want. (Adjust the '2pt' values to taste. The first one is the length of the dashes, the second is the length of the gaps.)

For dots in mid-paragraph, you need to say something like

```

\lower10pt\vbox{\leaders\copy\linedotbox\vskip2in}

```

which is scungy but works.

-- [mdw]

`\edfilldimen` A length. (was `\klamdimen`)

```

3611 \newdimen\edfilldimen
3612 \edfilldimen=0pt
3613

```

```

\c@addcolcount A counter to hold the number of a column. We use a roman number so that we can
\theadcolcount grab the column dimension from \dcol....

3614 \newcounter{addcolcount}
3615 \renewcommand{\theadcolcount}{\roman{addcolcount}}

\l@dtabaddcols \l@dtabaddcols{<startcol>}{<endcol>} adds the widths of the columns <startcol>
through <endcol> to \edfilldimen. It is a LaTeX style reimplementaion of the
original \@add@.

3616 \newcommand{\l@dtabaddcols}[2]{%
3617 \l@dccheckstartend{#1}{#2}%
3618 \ifl@dstartendok
3619 \setcounter{addcolcount}{#1}%
3620 \@whilenum \value{addcolcount}<#2\relax \do
3621 {\advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
3622 \advance\edfilldimen by \edtabcolsep
3623 \stepcounter{addcolcount}}%
3624 \advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
3625 \fi
3626 }
3627

\ifl@dstartendok \l@dccheckstartend{<startcol>}{<endcol>} checks that the values of <startcol> and
\l@dccheckstartend <endcol> are sensible. If they are then \ifl@dstartendok is set TRUE, otherwise it
is set FALSE.

3628 \newif\ifl@dstartendok
3629 \newcommand{\l@dccheckstartend}[2]{%
3630 \l@dstartendoktrue
3631 \ifnum #1<\@ne
3632 \l@dstartendokfalse
3633 \led@err@LowStartColumn
3634 \fi
3635 \ifnum #2>30\relax
3636 \l@dstartendokfalse
3637 \led@err@HighEndColumn
3638 \fi
3639 \ifnum #1>#2\relax
3640 \l@dstartendokfalse
3641 \led@err@ReverseColumns
3642 %%% \ledmac@error{Start column is greater than end column}{\@ehc}%
3643 \fi
3644 }
3645

\edrowfill \edrowfill{<startcol>}{<endcol>}fill fills columns <startcol> to <endcol> inclusive
\@edrowfill@ with <fill> (e.g. \hrulefill, \upbracefill). This is a LaTeX style reimplementaion
\@EDROWFILL@ and generalization of the original \waklam, \Waklam, \waklamec, \wastricht and
\wapunktel macros.

3646 \newcommand*\edrowfill}[3]{%

```

```

3647 \l@dtabaddcols{#1}{#2}%
3648 \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
3649 \let\@edrowfill@=\edrowfill
3650 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
3651

```

`\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`. (was `\linksltab`)

```

3652 \newcommand{\leftltab}[1]{%
3653 \hb@xt@ \z@{\vbox{\edtabindent%
3654 \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
3655

```

`\leftrtab`      `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`. (was `\linksrtab`)

```

3656 \newcommand{\leftrtab}[2]{%
3657 #2\hb@xt@ \z@{\vbox{\edtabindent%
3658 \advance\Hilfsskip by\dcoli%
3659 \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
3660

```

`\leftctab`      `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`. (was `\linksztab`)

```

3661 \newcommand{\leftctab}[2]{%
3662 \hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
3663 \advance\Hilfsskip by 0.5\dcoli%
3664 \setbox\hilfsbox=\hbox{\def\edlabel##1}{%
3665 \disablel@dtabfeet$\displaystyle{#2}$}%
3666 \advance\Hilfsskip by -0.5\wd\hilfsbox%
3667 \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
3668 #2}
3669

```

`\rightctab`      `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`. (was `\rechtsztab`)

```

3670 \newcommand{\rightctab}[2]{%
3671 \setbox\hilfsbox=\hbox{\def\edlabel##1}{%
3672 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
3673 #1\hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
3674 \advance\Hilfsskip by 0.5\l@dcolwidth%
3675 \advance\Hilfsskip by -\wd\hilfsbox%
3676 \setbox\hilfsbox=\hbox{\def\edlabel##1}{%
3677 \disablel@dtabfeet$\displaystyle{#1}$}%
3678 \advance\Hilfsskip by -0.5\wd\hilfsbox%
3679 \advance\Hilfsskip by \edtabcolsep%
3680 \moveright\Hilfsskip\hbox{ #2}}\hss}%
3681 }
3682

```

`\rightltab` `\rightltab{<math>}<{<text>}` for `\edaftertab` in `\ltab`. (was `\rechtsltab`)

```

3683 \newcommand{\rightltab}[2]{%
3684 \setbox\hilfsbox=\hbox{\def\edlabel##1{%
3685 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
3686 #1\hb@xt@#2{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
3687 \advance\Hilfsskip by\l@dcolwidth%
3688 \advance\Hilfsskip by-\wd\hilfsbox%
3689 \setbox\hilfsbox=\hbox{\def\edlabel##1{%
3690 \disablel@dtabfeet$\displaystyle{#1}$}%
3691 \advance\Hilfsskip by-\wd\hilfsbox%
3692 \advance\Hilfsskip by\edtabcolsep%
3693 \moveright\Hilfsskip\hbox{ #2}}\hss}%
3694 }
3695 }
```

`\rightrtab` `\rightrtab{<math>}<{<text>}` for `\edaftertab` in `\rtab`. (was `\rechtsrtab`)

```

3696 \newcommand{\rightrtab}[2]{%
3697 \setbox\hilfsbox=\hbox{\def\edlabel##1{%
3698 \disablel@dtabfeet#2}%
3699 #1\hb@xt@#2{\vbox{\edtabindent%
3700 \advance\Hilfsskip by-\wd\hilfsbox%
3701 \advance\Hilfsskip by\edtabcolsep%
3702 \moveright\Hilfsskip\hbox{ #2}}\hss}%
3703 }
3704 }
```

`\rtab` `\rtab{<body>}` typesets `<body>` as an array with the entries right justified. (was `\edbeforetab` `\rtab`) (Here and elsewhere, `\edbeforetab` and `\edaftertab` were originally `\davor` and `\danach`) The original `\rtab` and friends included a fair bit of common code which I have extracted into macros.

The process is first to measure the `<body>` to get the column widths, and then in a second pass to typeset the body.

```

3705 \newcommand{\rtab}[1]{%
3706 \l@dnnullfills
3707 \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
3708 \def\edaftertab##1##2{\righttab{##1}{##2}}%
3709 \measurebody{#1}%
3710 \l@drestorefills
3711 \variab
3712 \setmrowright #1\&\&\&%
3713 \enablel@dtabfeet}
3714 }
```

`\measurebody` `\measurebody{<body>}` measures the array `<body>`.

```

3715 \newcommand{\measurebody}[1]{%
3716 \disablel@dtabfeet%
3717 \l@dcolcount=0%
3718 \nullsetzen%
3719 \l@dcolcount=0}
```

```

3720 \measuremrow #1\\&\\%
3721 \global\l@dampcount=1}
3722

```

`\rtabtext` `\rtabtext{⟨body⟩}` typesets `⟨body⟩` as a tabular with the entries right justified. (was `\rtabtext`)

```

3723 \newcommand{\rtabtext}[1]{%
3724 \l@dnnullfills
3725 \measuretbody{#1}%
3726 \l@drestorefills
3727 \variab
3728 \setthrowright #1\\&\\%
3729 \enablel@dtabfeet}
3730

```

`\measuretbody` `\measuretbody{⟨body⟩}` measures the tabular `⟨body⟩`.

```

3731 \newcommand{\measuretbody}[1]{%
3732 \disablel@dtabfeet%
3733 \l@dcolcount=0%
3734 \nullsetzen%
3735 \l@dcolcount=0
3736 \measuretrrow #1\\&\\%
3737 \global\l@dampcount=1}
3738

```

`\ltab` Array with entries left justified. (was `\ltab`)

```

\edbeforetab 3739 \newcommand{\ltab}[1]{%
\edaftertab 3740 \l@dnnullfills
3741 \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
3742 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
3743 \measuretbody{#1}%
3744 \l@drestorefills
3745 \variab
3746 \setmrowleft #1\\&\\%
3747 \enablel@dtabfeet}
3748

```

`\ltabtext` Tabular with entries left justified. (was `\ltabtext`)

```

3749 \newcommand{\ltabtext}[1]{%
3750 \l@dnnullfills
3751 \measuretbody{#1}%
3752 \l@drestorefills
3753 \variab
3754 \setmrowleft #1\\&\\%
3755 \enablel@dtabfeet}
3756

```

`\ctab` Array with centered entries. (was `\ztab`)

```

\edbeforetab 3757 \newcommand{\ctab}[1]{%
\edaftertab

```

```

3758 \l@dnnullfills
3759 \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
3760 \def\edaftertab##1##2{\rightctab{##1}{##2}}%
3761 \measuretbody{#1}%
3762 \l@drestorefills
3763 \variab
3764 \setmrowcenter #1\&\%
3765 \enablel@dtabfeet}
3766

```

`\ctabtext` Tabular with entries centered. (new)

```

3767 \newcommand{\ctabtext}[1]{%
3768 \l@dnnullfills
3769 \measuretbody{#1}%
3770 \l@drestorefills
3771 \variab
3772 \setmrowcenter #1\&\%
3773 \enablel@dtabfeet}
3774

```

`\spreadtext` (was `\breitertext`)

```

3775 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
3776 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}

```

`\spreadmath` (was `\breiter`, ‘breiter’ = ‘broadly’)

```

3777 \newcommand{\spreadmath}[1]{%
3778 \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
3779

```

I have left the remaining TABMAC alone, apart from changing some names. I'm not yet sure what they do or how they do it. Authors should not use any of these as they are likely to be mutable.

`\tabellzwischen` (was `\tabellzwischen`)

```

3780 \def\tabellzwischen #1{%
3781 \ifx #1\ \let\NEXT\relax \l@dcolcount=0
3782 \else \step\l@dcolcount%
3783 \l@dcolwidth = #1 mm
3784 \let\NEXT=\tabellzwischen
3785 \fi \NEXT }
3786

```

`\edatabell` For example `\edatabell 4 & 19 & 8 \` specifies 3 columns with widths of 4, 19, and 8mm. (was `\atabell`)

```

3787 \def\edatabell #1\{%
3788 \tabellzwischen #1\&\}

```

`\Setzen` (was `\Setzen`, ‘setzen’ = ‘set’)

```

3789 \def\Setzen #1{%

```

```

3790 \ifx #1\relax \let\NEXT=\relax
3791 \else \step1@dcolcount%
3792 \let\tabelskip=\l@dcolwidth
3793 \EDTAB #1|
3794 \let\NEXT=\Setzen
3795 \fi\NEXT}
3796

```

\EDATAB (was \ATAB)

```

3797 \def\EDATAB #1\{
3798 \ifx #1\Relax \centerline{\Setzen #1\relax&}
3799 \let\Next\relax
3800 \else \centerline{\Setzen #1&\relax&}
3801 \let\Next=\EDATAB
3802 \fi\Next}

```

\edatab (was \atab)

```

3803 \newcommand{\edatab}[1]{
3804 \variab%
3805 \EDATAB #1\\Relax\\}
3806

```

\HILFSskip More helpers.

```

\Hilfsskip 3807 \newskip\HILFSskip
3808 \newskip\Hilfsskip
3809

```

\EDTABINDENT (was \TABINDENT)

```

3810 \newcommand{\EDTABINDENT}{
3811 \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
3812 \else\step1@dcolcount%
3813 \advance\Hilfsskip by\l@dcolwidth%
3814 \ifdim\l@dcolwidth=0pt\advance\hilfscount\@ne
3815 \else\advance\Hilfsskip by \the\hilfscount\edtabcolsep%
3816 \hilfscount=1\fi%
3817 \let\NEXT=\EDTABINDENT%
3818 \fi\NEXT}%

```

\edtabindent (was \tabindent)

```

3819 \newcommand{\edtabindent}{
3820 \l@dcolcount=0\relax
3821 \Hilfsskip=Opt%
3822 \hilfscount=1\relax
3823 \EDTABINDENT%
3824 \hilfsskip=\hsize%
3825 \advance\hilfsskip -\Hilfsskip%
3826 \Hilfsskip=0.5\hilfsskip%
3827 }%
3828

```



\EDTAB (was \TAB)

```
3829 \def\EDTAB #1|#2|{%
3830 \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
3831 \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
3832 \advance\tabelskip -\wd\tabhilfbox%
3833 \advance\tabelskip -\wd\tabHilfbox%
3834 \unhbox\tabhilfbox\hskip\tabelskip%
3835 \unhbox\tabHilfbox}%
3836
```

\EDTABtext (was \TABtext)

```
3837 \def\EDTABtext #1|#2|{%
3838 \setbox\tabhilfbox=\hbox{#1}%
3839 \setbox\tabHilfbox=\hbox{#2}%
3840 \advance\tabelskip -\wd\tabhilfbox%
3841 \advance\tabelskip -\wd\tabHilfbox%
3842 \unhbox\tabhilfbox\hskip\tabelskip%
3843 \unhbox\tabHilfbox}%

```

\tabhilfbox Further helpers.

```
\tabHilfbox 3844 \newbox\tabhilfbox
3845 \newbox\tabHilfbox
3846
```

```
%%%%%%%%%%
% That finishes tabmac
%%%%%%%%%%
```

edarrayl The 'environment' forms for \ltab, \ctab and \rtab.

```
edarrayc 3847 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{}
edarrayr 3848 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{}
3849 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{}
3850
```

edtabularl The 'environment' forms for \ltabtext, \ctabtext and \rtabtext.

```
edtabularc 3851 \newenvironment{edtabularl}{\l@dcollect@body\ltabtext}{}
edtabularr 3852 \newenvironment{edtabularc}{\l@dcollect@body\ctabtext}{}
3853 \newenvironment{edtabularr}{\l@dcollect@body\rtabtext}{}
3854
```

Here's the code for enabling \edtext (instead of \critext).

\usingcritext Declarations for using \critext{}.../ or using \edtext{}{} inside tabulars. The \disablel@dtabfeet default at this point is for \edtext.

```
\enablel@dtabfeet 3855 \newcommand{\usingcritext}{%
\usingedtext 3856 \def\disablel@dtabfeet{\l@dmodforcritext}%
3857 \def\enablel@dtabfeet{\l@drestoreforcritext}}
3858 \newcommand{\usingedtext}{%
```

```
3859 \def\disablel@dtabfeet{\l@dmodforedtext}%
3860 \def\enablel@dtabfeet{\l@drestoreforedtext}}
3861
3862 \usingedtext
3863
```

### **33 The End**

i/code<sub>i</sub>

## A Historic compatibility

. Since version 0.18, some change was made in code allow easy customization. It can make some trouble for people which have made their own customizations. The next sections explain how to correct.

### A.1 Version 0.18

This version add new commands to change display of footnote (see 4.2 p. 17). Author think all old customisation of `\<X>footmt` commands should be used anymore, because of new standard hook. But if you want, for any reason, to conserve it, you should replace :

```
\def\Afootmt#1#2#3{<code>}
```

Or

```
\newcommand{\Afootmt}[3]{<code>}
```

(and similar `\Xfootmt` commands) by :

```
\newcommandx*{\Afootmt}[4][4]{<code>}
```

### A.2 Version 0.18.1

As the `\Xfootnote` commands can have now an optional argument, you should modify your own `\Xfootnote` commands, using `\Afootnote` model. Note that in version 1.0, you won't have to define `\Xfootnote` commands : it will be generic.

## B Examples

This section presents some sample documents.

The examples in sections B.2 through B.5, plus B.7, were originally written for TeX. I have done some limited conversions of these so that they look more like LaTeX code. In particular wherever possible I have replaced `\def` commands by either `\newcommand` or `\renewcommand` as appropriate. I have also replaced the original TeX font handling commands by the LaTeX font commands.

The other examples were written natively in LaTeX.

The figures are from processed versions of the files. Having latexed a file I used DVIPS to get Encapsulated PostScript, then the `epstopdf` script to get a PDF version as well, for example:

```
> latex ledeasy
> latex ledeasy
```

```
> latex ledeasy
> dvips -E -o ledeasy.eps ledeasy
> epstopdf ledeasy.eps % produces ledeasy.pdf
```

For those who aren't fascinated by LaTeX code, I show the all the typeset results first, then the code that produced them.

# Simple Example

Peter Wilson\*

## Contents

|          |                        |          |
|----------|------------------------|----------|
| <b>1</b> | <b>First</b>           | <b>1</b> |
| 1.1      | Example text . . . . . | 1        |
| <b>2</b> | <b>Last</b>            | <b>1</b> |

## 1 First

This is a simple example of using the `ledmac` package with ordinary LaTeX constructs.

### 1.1 Example text

- 1 The `ledmac` package lets you do some unusual things in a LaTeX document.
- 2 For example you can have lines numbered and there are several levels of foot-
- 3 notes. You can label lines within the numbered text and refer to them outside. Sidenotes
- 4 Do not try and use any normal LaTeX `marginpars`<sup>1</sup> or `exotica` within the num- are OK
- 5 bered portions of the text.

## 2 Last

I forgot to mention that you can use ordinary footnotes<sup>2,3</sup> outside the numbered text. You can also<sup>a</sup> have<sup>b</sup> formatted footnotes<sup>c</sup> in normal<sup>d</sup> text.

There are 5 numbered lines in the example shown in section 1.1.

---

\*Standing on the shoulders of giants.

<sup>1</sup>You will get a warning but no text.

<sup>2</sup>An ordinary footnote

<sup>3</sup>And another

---

<sup>a</sup>Additionally <sup>b</sup>Specify <sup>c</sup>Like this <sup>d</sup>Text that does not have line numbers

---

2 several] This is an 'A' footnote.

4 `exotica`] Like floats.

---

2 levels] This is a 'B' level footnote.

Figure 1: Output from `ledeasy.tex`.

This is an example of some text with variant readings recorded as ‘A’ foot-  
notes. From here on, though, we shall have ‘C’. For spice, let us mark a longer  
3 passage, but give a different lemma for it, so that we don’t get a huge amount  
4 of text in a note. Finally, we shouldn’t forget the paragraphed notes, which are  
5 so useful when there are a great number of short notes to be recorded.  
6 This is a second paragraph, giving more *examples* of text with variant read-  
7 ings recorded as ‘A’ footnotes. From here on, though, we shall have ‘B’ notes in  
8 the text. For spice, let us mark a longer passage, but give a different lemma for  
9 it, so that we don’t get a *huge* amount of text in a note. Finally, we shouldn’t  
10 forget the column notes, which are so useful when there are many short notes  
11 to be recorded.

|                                                                                          |                                              |                                |
|------------------------------------------------------------------------------------------|----------------------------------------------|--------------------------------|
| <hr/>                                                                                    |                                              |                                |
| 1 example:: eximemple C, D.                                                              | 6 <i>examples</i> :: eximples L, M.          |                                |
| 1 variant:: alternative, A, B.                                                           | 6 variant:: alternative, A, B.               |                                |
| 2 though:: however $\alpha$ , $\beta$                                                    |                                              |                                |
| <hr/>                                                                                    |                                              |                                |
| 2 ‘C’] B, <i>pace</i> the text                                                           | 9 shouldn’t] ought not to                    | 10 useful] very, very useful   |
| 7 though] however $\alpha$ , $\beta$                                                     | L, M                                         | L, P                           |
| 7 ‘B’] B, as correctly                                                                   | 10 forget the] omit to                       | 10 many] lots of Z             |
| stated in the text                                                                       | mention the §, ¶                             | 11 recorded] recorded and      |
| 9 Finally] In the end X,                                                                 | 10 column] blocked M, N                      | put down: M                    |
| Y                                                                                        | 10 notes] variants H                         | (repetition)                   |
| 9 we] we here K                                                                          |                                              |                                |
| <hr/>                                                                                    |                                              |                                |
| 2–4 For spice ... note: The note here is type ‘C’                                        |                                              |                                |
| 8–9 For spice, ... note: This is a rogue note of type ‘C’.                               |                                              |                                |
| <hr/>                                                                                    |                                              |                                |
| 3 huge: vast E, F; note that this is a ‘D’ note to section of text within a longer lemma |                                              |                                |
| 9 huge: vast E, F; note that this is a ‘D’ note to text within a longer lemma.           |                                              |                                |
| <hr/>                                                                                    |                                              |                                |
| 4 Finally: in the end X, Y                                                               | 4 we: us K                                   | 4 shouldn’t: ought not to L, M |
| 4 forget                                                                                 | 4 paragraphed: blocked M, N                  | 4 notes: variants HH, KK       |
| the: omit to mention the §, ¶                                                            | 4 a great number of: many, many (preferably) | 5 recorded:                    |
| 5 useful: truly useful L, P                                                              | noted: repetition                            |                                |

Figure 2: Output from `ledfeat.tex`.

*Oedipus entreth.*

Or that with wrong the right and doubtlesse heire,  
 Shoulde banisht be out of his princely seate.  
 Yet thou O queene, so fyle thy sugred tounge,  
 And with suche counsell decke thy mothers tale,  
 That peace may bothe the brothers heartes inflame, 5  
 And rancour yelde, that erst possesse the same.

*Eteocl.* Mother, beholde, youre hestes for to obey,  
 In person nowe am I resorted hither:  
 In haste therefore, fayne woulde I knowe what cause  
 With hastie speede, so moued hath your mynde 10  
 To call me nowe so causelesse out of tyme,  
 When common wealth moste craues my onely ayde:  
 Fayne woulde I knowe, what queynt commoditie  
 Persuades you thus to take a truce for tyme,  
 And yelde the gates wide open to my foe, 15  
 The gates that myght our stately state defende,  
 And nowe are made the path of our decay.

„ *Ioca.* Represse deare son, those raging stormes of wrath,  
 „That so bedimme the eyes of thine intente,  
 „As when the tongue (a redy Instrument) 20  
 „Would fayne pronounce the meaning of the minde,  
 „It cannot speake one honest seemely worde.  
 „But when disdayne is shrunke, or sette asyde,  
 „And mynde of man with leysure can discourse  
 „What seemely woordes his tale may best beseeme, 25  
 „And that the tounge vnfoldes without affectes  
 „Then may proceede an answere sage and graue,  
 „And euery sentence sawst with sobernesse:  
 Wherefore vnbende thyne angrie browes deare chylde,  
 And caste thy rolling eyes none other waye, 30  
 That here doost not *Medusaes* face beholde,  
 But him, euen him, thy blood and brother deare.  
 And thou beholde, my *Polinices* eke,  
 Thy brothers face, wherein when thou mayst see  
 Thine owne image, remember therwithall, 35  
 That what offence thou woldst to him were done,

0.1 entreth] *intrat* MS 20–22 As ... worde.] *not in* 73 20 the] *thie* MS 21 fayne  
 pronounce] *faynest tell* MS 21 the minde] *thy minde* MS 22 It ... worde.] *Thie swelling*  
 hart puft vp with wicked ire / Can scarce pronounce one inward louing thought. MS 31  
*Medusaes*] *One of the furies.* 75m

[SCENE III.—*Venice.*]*Enter JESSICA and [LAUNCELOT] the clown.*

- Jes.* I am sorry thou wilt leave my father so,  
 Our house is hell, and thou (a merry devil)  
 Didst rob it of some taste of tediousness,—  
 But fare thee well, there is a ducat for thee,  
 And Launcelot, soon at supper shalt thou see 5  
 Lorenzo, who is thy new master's guest,  
 Give him this letter,—do it secretly,—  
 And so farewell: I would not have my father  
 See me in talk with thee.
- Laun.* Adieu! tears exhibit my tongue, most beautiful pagan, most sweet 10  
 Jew!—if a Christian do not play the knave and get thee, I am much  
 deceived; but adieu! these foolish drops do something drown my  
 manly spirit: adieu! *[Exit.]*
- Jes.* Farewell good Launcelot.  
 Alack, what heinous sin is it in me 15  
 To be ashamed to be my father's child!

Scene III] *Capell*; *om.* *Q*, *F*; Scene IV *Pope*. *Venice*] *om.* *Q*, *F*; *Shylock's house* *Theobald*;  
*The same. A Room in Shylock's House* *Capell*. Launcelot] *Rowe*; *om.* *Q*, *F*. 1. I am] *Q*,  
*F*; I'm *Pope*. 9. in] *Q*; *om.* *F*. 10. *Laun.*] *Q2*; *Clowne*. *Q*, *F*. 10. Adieu!] Adieu, *Q*, *F*.  
11. Jew!] Iewe, *Q*, *F*. do] *Q*, *F*; did *F2*. 12. adieu!] adieu, *Q*, *F*. 12. something] *Q*;  
somewhat *F*. 13. adieu!] adieu. *Q*, *F*. S. D.] *Q2*, *F*; *om.* *Q*; after l. 15 *Capell*. 16. child!]  
child, *Q*, *F*; Child? *Rowe*.

5. *soon*] early.

10. *exhibit*] *Eccles* paraphrased "My tears  
 serve to express what my tongue should, if  
 sorrow would permit it," but probably it is  
 Launcelot's blunder for prohibit (*Halliwell*)  
 or inhibit (*Clarendon*).

10. *pagan*] This may have a scurrilous un-  
 dertone: cf. 2 *H* 4, II. ii. 168.

11. *do*] *Malone* upheld the reading of *Qq*

and *F* by comparing II. vi. 23: "When you  
 shall please to play the thieves for wives";  
 Launcelot seems fond of hinting at what is  
 going to happen (cf. II. v. 22–3). If *F2*'s "did"  
 is accepted, *get* is used for beget, as in III. v.  
 9.

12–13. *foolish... spirit*] "tears do not be-  
 come a man" (*AYL.*, III. iv. 3); cf. also *H* 5,  
 IV. vi. 28–32.

Figure 4: Output from `ledarden.tex`.



1 incipit . . . ΠΕΡΙΦΥΣΕΩΝ] *om. R, incipit quartus M* 2 ΑΝΑΚΕΦΑΛΙΟΣΙΣ] *FJP, lege ἀνακεφαλαιώσεις* 2 physiologiae] *physiologiae P, physeologiae R* 3 quod] *p. natura transp.* *MR* 3 ΥΠΕΡΟΥΣΙΑΔΕΣ] *codd. Vtrum ὑπερουσιώδης (hoc est supersessentialis) natura cum Gale (p.160) an ὑπερουσιότης (hoc est supersessentialis natura) cum Floss (PL 122,741C) intelligendum sit, ambigitur* 7 ΟΜΟΤΗΣΙΟΣ] *codd., lege ὁμοούσιος* 7 *et*] *R<sup>1</sup>, om. R<sup>0</sup>* 9 ΑΓΚΥΡΑΤΩ] *anchurato MR* 9 *de fide*] *Glo(ssa): Ita enim uocatur sermo eius de fide* ΑΓΚΥΡΑΤΟΣ, *id est procuratus mg. add. FJP* 10 *agentia*] *actiua MR* 10 *formantia*] *formatiua MR* 11 *operantia*] *operatiua MR* 13 *eiusdem*] *eiusdemque M* 13 *eiusdem* *uirtutis, eiusdem subsistentiae*] *om. M* 13 *subsistentiae*] *substantia R* 14 *similiter*] *ex simili MR* 15 *sunt*] *om. M* 25 *spiritus sanctus*] *sanctus spiritus R*

## Chronicle of Guelders

Guillelmus de Berchen

### St. Stephen's Church in Nijmegen

Nobilis itaque comes Otto imperio et dominio Novimagensi sibi, ut praefer-  
tur, impignoratis et commissis proinde praeesse cupiens, anno LIIII superius 1254  
descripto, mense Iunio, una cum iudice, scabinis ceterisque civibus civitatis  
Novimagensis, pro ipsius et inhabitantium in ea necessitate, commodo et utili-  
tate, ut ecclesia eius parochialis extra civitatem sita destrueretur et infra muros  
5 transferretur ac de novo construeretur, a reverendo patre domino Conrado de  
Hofsteden, archiepiscopo Coloniensi, licentiam, et a venerabilibus dominis de-  
cano et capitulo sanctorum Apostolorum Coloniensi, ipsius ecclesiae ab antiquo  
veris et pacificis patronis, consensum, citra tamen praeiudicium, damnum aut  
10 gravamen iurium et bonorum eorundem, impetravit.

Et exinde liberum locum eiusdem civitatis qui dicitur Hundisbrug, de prae-  
libati Wilhelmi Romanorum regis, ipsius fundi domini, consensu, ad aedifican-  
dum et consecrandum ecclesiam et coemeterium, eisdem decano et capitulo de  
expresso eiusdem civitatis assensu libera contradiderunt voluntate, obligantes  
15 se ipsi comes et civitas dictis decano et capitulo, quod in recompensationem  
illius areae infra castrum et portam, quae fuit dos ecclesiae, in qua plebanus  
habitare solebat—quae tunc per novum fossatum civitatis est destructa—aliam  
aream competentem et ecclesiae novae, ut praefertur, aedificandae satis conti-  
guam, ipsi plebano darent et assignarent. Et desuper apud dictam ecclesiam  
20 sanctorum Apostolorum est littera sigillis ipsorum Ottonis comitis et civitatis  
Novimagensis sigillata.

3 p. 227 R 4 p. 97 N 6 p. 129 D 12 f. 72v M 13 p. 228 R 20 p. 130 D

2 proinde] primum D 5 ecclesia eius] ecclesia D: eius eius H extra civitatem *om.* H  
infra] intra D 6 transferretur] transferreretur NH 7 Hofsteden] Hoffstede D: Hoffsteden  
H Coloniensi] Colononiensi H dominis] viris H 8 Coloniensi] Coloniae H 10 iurium]  
virium D 11 liberum] librum H qui] quae D Hundisbrug] Hundisburch D: Hunsdisbrug  
R 12 regis] imperatoris D 13 et consecrandum *om.* H eisdem] eiusdem D 15 comes]  
comites D dictis *om.* H 17 tunc] nunc H 18 ut...aedificandae *om.* H 18–19 contiguam]  
contiguum M 19 apud *om.* H 20 est] et H littera] litteram H 21 Novimagensis]  
Novimagii D sigillata] sigillis communita H

6–7 William is confusing two charters that are five years apart. Permission from St. Apost-  
les' Church in Cologne had been obtained as early as 1249. Cf. Sloet, *Oorkondenboek* nr.  
707 (14 November 1249): "...nos devotionis tue precibus annuentes, ut ipsam ecclesiam  
faciens demoliri transferas in locum alium competentem, tibi auctoritate presentium indulge-  
mus..." 6–7 Conrad of Hochstaden was archbishop of Cologne in 1238–1261 11–21 Cf.  
Sloet, *Oorkondenboek* nr. 762 (June 1254)

Figure 6: Output from `ledecker.tex`.

## 22

[Seán Ó Braonáin cct] chuim Tomás Uí Dhúnlain  
 [Fonn: Máirseáil Uí Shúilleabháin (Páinseach na nUbh)]

- 1** A dhuine gan chéill do mhaisligh an chlér  
 b is tharcainnigh naomhscriupt na bhfáige,  
 c na haitheanta réab 's an t-aifreann thréig  
 d re taitheamh do chlaonchreideamh Mhártain,  
 e cá rachair 'od dhíon ar Íosa Nasardha  
 f nuair chaithfimid cruinn bheith ar mhaoileann  
 Josepha?  
 g Ní caraid Mac Crae chuim t'anama ' phlé  
 h ná Calvin bhais taobh ris an lá sin.
- 2** Nách damanta an scéal don chreachaire chlaon  
 b ghlac baiste na cléire 'na pháiste  
 c 's do glanadh mar ghréin ón bpeaca ró-dhaor  
 d trí ainibhfios Éva rinn Ádam,  
 e tuitim arís fé chuing na haicme sin  
 f tug atharrach brí don scríbhinn bheannaithe,  
 g d'aistrigh béasa agus reachta na cléire  
 h 's nách tugann aon ghéilleadh don Phápa?
- 3** Gach scolaire baoth, ní mholaim a cheird  
 b 'tá ag obair le géilleadh dá tháille  
 c don doirbhchoin chlaon dá ngorthar Mac Crae,  
 d deisceabal straeigh as an gcolláiste.  
 e Tá adaithe thíos in íochtar ifrinn,  
 f gan solas gan soilse i dtíorthaibh dorchá,  
 g tuigsint an léinn, gach cuirpeacht déin  
 h is Lucifer aosta 'na mháistir.

**22 Teideal:** Dhuinnluinn T, Seághan Mac Domhnaill cct B  
 1.a dhuinne T 1.a mhaslaidh T, mhaslaig B 1.c raob T 1.d le B 1.e  
 dod B 1.f chaithfamíd T 1.f maolinn B 1.g phleith T 1.h bhíos B  
 1.h leis B 2.a claon B 2.c glannuig T 2.d ainibhfios T, ainibhfios B  
 2.d Éabha B 2.g is B 2.h tuigíonn T 3.a sgollaire T 3.a mholluim T  
 3.b 'tág ccobar T 3.b re B 3.c dorbhchon daor B 3.d straodhaig T  
 3.e fhadoghthe tsíos T 3.e fadaighthe B 3.f sollus T 3.g cuirripeacht T  
 3.h Luicifer T, Lúcifer B 3.h mhaighistir T

Figure 7: Output from `ledbraonain.tex`.

## B.1 Simple example

This made-up example, `ledeasy.tex`, is included to show how simple it can be to use EDMAC in a LaTeX document. The code is given below and the result is shown in Figure 1.

---

```

3864 (*easy)
3865 % ledeasy.tex simple example of the ledmac package
3866 \documentclass{article}
3867 \usepackage{ledmac}
3868 %% number every line
3869 \setcounter{firstlinenum}{1}
3870 \setcounter{linenumincrement}{1}
3871 %% Show some B series familiar footnotes, lettered and paragraphed
3872 \renewcommand*{\thefootnoteB}{\alph{footnoteB}}
3873 \footparagraphX{B}
3874 %% no endnotes
3875 \noendnotes
3876 %% narrow sidenotes
3877 \setlength{\ledrsnotewidth}{4em}
3878 \title{Simple Example}
3879 \author{Peter Wilson\thanks{Standing on the shoulders of giants.}}
3880 \date{}
3881 \begin{document}
3882 \maketitle
3883 \tableofcontents
3884 \section{First}
3885 This is a simple example of using the \textsf{ledmac}
3886 package with ordinary LaTeX constructs.
3887
3888 \subsection{Example text}\label{subsec}
3889
3890 \beginnumbering
3891 \pstart
3892 The \textsf{ledmac} package lets you do some unusual things in
3893 a LaTeX document. For example you can have lines numbered and
3894 there are
3895 \edtext{several}{\Afootnote{This is an ‘A’ footnote.}}
3896 \edtext{levels}{\Bfootnote{This is a ‘B’ level footnote.}}
3897 of footnotes.
3898 You can label lines within the numbered text and refer to them
3899 outside. Do not try and use any normal LaTeX
3900 \marginpars\footnote{You will get a warning but no text.}%
3901 \ledrightrightnote{Sidenotes are OK}
3902 or \edtext{exotica}{\Afootnote{Like floats.}}
3903 within the numbered portions of the text\edlabel{line}.
3904 \pend
3905 \endnumbering
3906
3907 \section{Last}

```

```

3908
3909 I forgot to mention that you can use ordinary
3910 footnotes\footnote{An ordinary footnote}\footnote{And another}
3911 outside the numbered text. You can also\footnoteB{Additionally}
3912 have\footnoteB{Specify} formatted footnotes\footnoteB{Like this}
3913 in normal\footnoteB{Text that does not have line numbers} text.
3914
3915 There are \lineref{line} numbered lines in the example shown
3916 in section~\ref{subsec}.
3917
3918 \end{document}
3919 </easy>

```

## B.2 General example of features

This made-up example, `ledfeat.tex`, is included purely to illustrate some of `ledmac`'s main features. It is hard to find real-world examples that actually use as many layers of notes as this, so we made one up. The example is a bit tricky to read, but close study and comparison with the output (Figure 2) will be illuminating.

I have converted the original TeX code to look more like LaTeX code.

---

```

3920 <features>
3921 % ledfeat.tex Small test file for ledmac package
3922 \documentclass{article}
3923 \usepackage{ledmac}
3924
3925 \noendnotes % we aren't having any endnotes
3926
3927 \makeatletter
3928 % I'd like a spaced out colon after the lemma:
3929 \newcommand{\spacedcolon}{\rmfamily\thinspace:\thinspace}
3930 \renewcommand*{\normalfootfmt}[3]{%
3931 \ledsetnormalparstuff
3932 {\notenumfont\printlines#1|}\strut\enspace
3933 {\select@lemmafont#1/#2}\spacedcolon\enskip#3\strut\par}
3934
3935 % And I'd like the 3-col notes printed with a hanging indent:
3936 \renewcommand*{\threecolfootfmt}[3]{%
3937 \normal@pars
3938 \hsize .3\hsize
3939 \setlength{\parindent}{0pt}
3940 \tolerance=5000 % high, but not infinite
3941 \raggedright
3942 \hangindent1.5em \hangafter1
3943 \leavevmode
3944 \strut\hbox to 1.5em{\notenumfont\printlines#1|\hfil}\ignorespaces
3945 {\select@lemmafont#1/#2}\rbracket\enskip
3946 #3\strut\par\allowbreak}
3947

```

```

3948 % And I'd like the 2-col notes printed with a double colon:
3949 \newcommand*{\doublecolon}{\rmfamily\thinspace:\thinspace}
3950 \renewcommand*{\twocolfootfmt}[3]{%
3951 \normal@pars
3952 \hsize .45\hsize
3953 \setlength{\parindent}{0pt}
3954 \tolerance=5000
3955 \raggedright
3956 \leavevmode
3957 \strut{\notenumfont\printlines#1/}\enspace
3958 {\select@lemmafont#1/#2}\doublecolon\enskip
3959 #3\strut\par\allowbreak}
3960
3961 % And in the paragraphed footnotes, I'd like a colon too:
3962 \renewcommand*{\parafootfmt}[3]{%
3963 \ledsetnormalparstuff
3964 {\notenumfont\printlines#1/}\enspace
3965 {\select@lemmafont#1/#2}\spacedcolon\enskip
3966 #3\penalty-10 }
3967 \makeatother
3968
3969 % I'd like the line numbers picked out in bold.
3970 \renewcommand{\notenumfont}{\bfseries}
3971 \lineation{page}
3972 \linenummargin{inner}
3973 \setcounter{firstlinenum}{3} % just because I can
3974 \setcounter{linenumincrement}{1}
3975 \foottwocol{A}
3976 \footthreecol{B}
3977 \footparagraph{E}
3978 % I've changed \normalfootfmt, so invoke it again for C and D notes.
3979 \footnormal{C}
3980 \footnormal{D}
3981
3982 \begin{document}
3983
3984 \begin{numbering}
3985
3986 \pstart
3987 This is an \edtext{example}{
3988 \Afootnote{eximemple C, D.}}
3989 of some %\footnote{A normal footnote}
3990 text with \edtext{variant}{
3991 \Afootnote{alternative, A, B.}}
3992 readings recorded as 'A' footnotes. From here on, \edtext{though}{
3993 \Afootnote{however α, β}},
3994 we shall have \edtext{'C'}{
3995 \Bfootnote{B, \textit{pace} the text}}.
3996 \edtext{For spice, let us mark a longer passage, but give a different
3997 lemma for it, so that we don't get a \edtext{huge}{

```

```

3998 \Dfootnote{vast E, F; note that this is
3999 a 'D' note to section of text within a longer lemma}}
4000 amount of text in a note}{\lemma{For spice \dots\ note}
4001 \Cfootnote{The note here is type 'C'}}.
4002 \edtext{Finally}{
4003 \Efootnote{in the end X, Y}},
4004 \edtext{we}{
4005 \Efootnote{us K}}
4006 \edtext{shouldn't}{
4007 \Efootnote{ought not to L, M}}
4008 \edtext{forget the}{
4009 \Efootnote{omit to mention the \S, \P}}
4010 \edtext{paragraphed}{
4011 \Efootnote{blocked M, N}}
4012 \edtext{notes}{
4013 \Efootnote{variants HH, KK}},
4014 which are so \edtext{useful}{
4015 \Efootnote{truly useful L, P}}
4016 when there are \edtext{a great number of}{
4017 \Efootnote{many, many (preferably)}}
4018 short notes to be \edtext{recorded}{
4019 \Efootnote{noted: repetition}}.
4020 \pend
4021
4022 \pstart
4023 This is a second paragraph, giving more \textit{\edtext{examples}}{
4024 \Afootnote{eximples L, M.}}}
4025 of text with \edtext{variant}{
4026 \Afootnote{alternative, A, B.}}
4027 readings recorded as 'A' footnotes. From here on, \edtext{though}{
4028 \Bfootnote{however α, β}},
4029 we shall have \edtext{'B'}{
4030 \Bfootnote{B, as correctly stated in the text}} notes in the text.
4031 \edtext{For spice, let us mark a longer passage, but give a different
4032 lemma for it, so that we don't get a \textit{\edtext{huge}}{
4033 \Dfootnote{vast E, F; note that this is
4034 a 'D' note to text within a longer lemma.}}}
4035 amount of text in a note}{\lemma{For spice, \dots\ note}
4036 \Cfootnote{This is a rogue note of type 'C'.}}.
4037 \edtext{Finally}{
4038 \Bfootnote{In the end X, Y}},
4039 \edtext{we}{
4040 \Bfootnote{we here K}}
4041 \edtext{shouldn't}{
4042 \Bfootnote{ought not to L, M}}
4043 \edtext{forget the}{
4044 \Bfootnote{omit to mention the \S, \P}}
4045 \edtext{column}{
4046 \Bfootnote{blocked M, N}}
4047 \edtext{notes}{

```

```

4048 \Bfootnote{variants H}},
4049 which are so \edtext{useful}{
4050 \Bfootnote{very, very useful L, P}}
4051 when there are \edtext{many}{
4052 \Bfootnote{lots of Z}}
4053 short notes to be \edtext{recorded}{
4054 \Bfootnote{recorded and put down: M (repetition)}}.
4055 \pend
4056
4057 \endnumbering
4058 \end{document}
4059 </features>

```

---

### B.3 Gascoigne

The first real-life example is taken from an edition of George Gascoigne's *A Hundreth Sundrie Flowres* that is being prepared by G. W. Pigman III, at the California Institute of Technology. Figure 3 shows the result of setting the text with `ledmac`.

I have LaTeXified the original code, and removed all the code related to the main document layout, relying on the standard LaTeX layout parameters..

---

```

4060 <*ioc>
4061 %% ledioc.tex
4062 \documentclass{article}
4063 \usepackage{ledmac}
4064
4065 \noendnotes
4066 \makeatletter
4067
4068 \newcommand{\os}{\scriptsize}
4069 \setcounter{firstsublinenum}{1000}
4070 \frenchspacing \setlength{\parskip}{0pt} \hyphenpenalty=1000
4071
4072 % Say \nolinenums if you want no line numbers in the notes.
4073 \newif\ifnolinenums
4074 \newcommand{\nolinenums}{\global\nolinenumstrue}
4075 \newcommand{\linenums}{\global\nolinenumsfalse}
4076
4077 \renewcommand{\rightlinenum}{\ifbypage@\ifnum\line@num<10\kern.5em\fi\else
4078 \ifnum\line@num<10\kern1em\else\ifnum\line@num<100
4079 \kern.5em\fi\fi\fi\kern.5em\numlabfont\the\line@num
4080 \ifnum\subline@num>0:\the\subline@num\fi}
4081
4082 \renewcommand{\leftlinenum}{\numlabfont\the\line@num
4083 \ifnum\subline@num>0:\the\subline@num\fi \kern.5em}
4084 \linenummargin{outer}
4085 \lineation{page}

```



```

4086
4087 \newcommand{\ggfootfmt}[3]{%
4088 \notefontsetup
4089 \let\par=\endgraf
4090 \rightskip=0pt \leftskip=0pt
4091 \setlength{\parindent}{0pt} \parfillskip=0pt plus 1fil
4092 \ifnolinenums\relax\else
4093 \begingroup \os \printlines#1\endgroup
4094 \enskip
4095 \fi
4096 {\rmfamily #2\def\@tempa{#2}\ifx\@tempa\empty
4097 \else\enskip\fi#3\penalty-10 }}
4098
4099 % Now reset the \Afootnote parameters and macros:
4100 \footparagraph{A}
4101 \let\Afootfmt=\ggfootfmt
4102 \dimen\Afootins=\vsize
4103 \skip\Afootins=3pt plus9pt
4104 \newcommand*{\ggfootstart}[1]{\vskip\skip\Afootins}
4105 \let\Afootstart=\ggfootstart
4106
4107 \newcommand*{\stage}[1]{\pstart\startsub\parindent=0pt
4108 \hangindent=3em\hangafter=0
4109 {\itshape #1}\let\par=\finishstage}
4110 \newcommand{\finishstage}{\pend\endsub}
4111 \newcommand{\sen}{\leavevmode\lower1ex\hbox{\textrm{'}}}%
4112 \newcommand{\senspeak}[1]{\pstart\obeylines\setbox0=\hbox{\textrm{'}}}%
4113 \leavevmode
4114 \lower1ex\copy0\kern-\wd0\hskip1em{\textit{#1}}}%
4115 \hbox tolex{\ignorespaces}
4116 \newcommand*{\speak}[1]{\pstart\obeylines\hskip1em{\textit{#1}}}%
4117 \hbox tolex{\ignorespaces}
4118 \def\nospeaker{\parindent=0em\pstart\let\par=\pend}
4119 \newcommand*{\nospeak}{\pstart\obeylines}
4120 \makeatother
4121
4122 \begin{document}
4123
4124 \setlength{\parindent}{0pt}
4125
4126 \beginnumbering
4127
4128 \stage{Oedipus \edtext{entreth}{\Afootnote{\textit{intrat} MS}}.}
4129
4130 \nospeak
4131 Or that with wrong the right and doubtlesse heire,
4132 Shoulde banisht be out of his princely seate.
4133 Yet thou O queene, so fyle thy sugred tounge,
4134 And with suche counsell decke thy mothers tale,
4135 That peace may bothe the brothers heartes inflame,

```

```

4136 And rancour yelde, that erst possesse the same.
4137 \pend
4138
4139 \speak{Eteocl.} Mother, beholde, youre hestes for to obey,
4140 In person nowe am I resorted hither:
4141 In haste therefore, fayne woulde I knowe what cause
4142 With hastie speede, so moued hath your mynde
4143 To call me nowe so causelesse out of tyme,
4144 When common wealth moste craues my onely ayde:
4145 Fayne woulde I knowe, what queynt commoditie
4146 Persuades you thus to take a truce for tyme,
4147 And yelde the gates wide open to my foe,
4148 The gates that myght our stately state defende,
4149 And nowe are made the path of our decay.
4150 \pend
4151
4152 \sensspeak{Ioca.} Represse deare son, those raging stormes of wrath,
4153 \sen That so bedimme the eyes of thine intende,
4154 \edtext{\sen As when \edtext{the}{\Afootnote{thie MS}} tongue %
4155 (a redy Instrument)
4156 \sen Would \edtext{fayne pronounce}{\Afootnote{faynest tell MS}} %
4157 the meaning of \edtext{the minde}{\Afootnote{thy minde MS}},
4158 \sen \edtext{It}{\lemma{It \dots\ worde.}\Afootnote{Thie %
4159 swelling hart puft vp with wicked ire / Can scarce pronounce %
4160 one inward louing thought. MS}} cannot speake one honest %
4161 seemely worde.}{\lemma{As \dots\ worde.}\Afootnote{\textit{not %
4162 in} \os73}}
4163 \sen But when disdayne is shrunke, or sette asyde,
4164 \sen And mynde of man with leysure can discourse
4165 \sen What seemely woordes his tale may best beseeme,
4166 \sen And that the toung vnfoldes without affectes
4167 \sen Then may proceede an answere sage and graue,
4168 \sen And euery sentence sawst with sobernesse:
4169 Wherefore vnbende thyne angrie browes deare chylde,
4170 And caste thy rolling eyes none other waye,
4171 That here doost not \edtext{\textit{Medusaes}}{\%
4172 \Afootnote{One of the furies. {\os75m}} face beholde,
4173 But him, euen him, thy blood and brother deare.
4174 And thou beholde, my \textit{Polinices} eke,
4175 Thy brothers face, wherein when thou mayst see
4176 Thine owne image, remember therewithall,
4177 That what offence thou woldst to him were done,
4178 \pend
4179 \endnumbering
4180
4181 \end{document}
4182
4183 \end{document}

```

---

## B.4 Shakespeare

The following text illustrates another input file of moderate complexity, with two layers of annotation in use. The example is taken from the Arden *Merchant of Venice*.

I have roughly converted the original TeX file to a LaTeX file. The file is below and the result of LaTeXing it is shown in Figure 4.

---

```

4184 (*arden)
4185 %% ledarden.tex
4186 \documentclass{article}
4187 \usepackage{ledmac}
4188
4189 \makeatletter
4190 \newcommand{\stage}[1]{\rlap{\hbox to \the\linenumsep{%
4191 \hfil\llap{[\textit{#1}]}}}
4192
4193 \newcommand{\speaker}[1]{\pstart\hangindent2em\hangafter1
4194 \leavevmode\textit{#1}\enspace\ignorespaces}
4195
4196 \newcommand{\exit}[1]{\hfill\stage{#1}}
4197
4198 % LEDMAC customizations:
4199 \noendnotes
4200 \setlength{\parindent}{0pt}
4201 \setlength{\linenumsep}{.4in}
4202 \rightskip\linenumsep
4203
4204 \renewcommand{\interparanoteglue}{1em plus.5em minus.1em}
4205
4206 \newcommand{\scf}{\tiny}
4207 \let\Afootnoterule=\relax \let\Bfootnoterule=\relax
4208
4209 \renewcommand{\rightlinenum}{\numlabfont\llap{\the\line@num}}
4210 \frenchspacing
4211
4212 % Footnote formats:
4213 % \nonumparafootfmt is a footnote format without line numbers.
4214 \newcommand{\nonumparafootfmt}[3]{%
4215 \ledsetnormalparstuff
4216 \rightskip=0pt
4217 \select@lemmafnt#1/#2\rbracket\enskip
4218 \itshape #3\penalty-10 }
4219
4220 \newcommand{\newparafootfmt}[3]{%
4221 \ledsetnormalparstuff
4222 {\notenumfont\printlines#1/}\fullstop\enspace
4223 {\select@lemmafnt#1/#2\rbracket\enskip
4224 \itshape #3\penalty-10 }
4225

```

```

4226 \newcommand{\newtwocolfootfmt}[3]{%
4227 \normal@pars
4228 \hsize .48\hsize
4229 \tolerance=5000
4230 \rightskip=0pt \leftskip=0pt \parindent=5pt
4231 \strut\notenumfont\printlines#1/\fullstop\enspace
4232 \itshape #2/>\rbracket\penalty100\hskip .5em plus .5em
4233 \normalfont #3\strut\goodbreak}
4234
4235 % Footnote style selections etc. (done last):
4236 \footparagraph{A}
4237 \foottwocol{B}
4238 \let\Afootfmt=newparafootfmt
4239 \let\Bfootfmt=newtwocolfootfmt
4240 \let\collation=Afootnote
4241 \let\note=Bfootnote
4242 \lineation{section}
4243 \linenummargin{right}
4244 \makeatother
4245
4246 %%%
4247
4248 \begin{document}
4249 \pagestyle{empty}
4250
4251 % Initially, we don't want line numbers.
4252 \let\Afootfmt=nonumparafootfmt
4253
4254 \beginnumbering
4255 \pstart
4256 \centerline{[\edtext{SCENE III}{
4257 \lemma{Scene III}
4258 \collation{Capell; om. Q, F; \textnormal{Scene IV} Pope.}}.---%
4259 \edtext{\textit{Venice}}}{
4260 \collation{om. Q, F; Shylock's house Theobald; The same.
4261 A Room in Shylock's House Capell.}}.]]}
4262 \pend
4263 \bigskip
4264
4265 \pstart
4266 \centerline{\textit{Enter} JESSICA \textit{and}
4267 [\edtext{LAUNCELOT}{
4268 \lemma{Launcelot}
4269 \collation{Rowe; om. Q, F.}}] \textit{the clown.}} \pend \bigskip
4270
4271 \let\Afootfmt=newparafootfmt % we do want line numbers from now
4272
4273 \setline{0}%
4274
4275 \speaker{Jes.}\edtext{I am}{

```

```

4276 \collation{Q, F; \textnormal{I'm} Pope.}}
4277 sorry thou wilt leave my father so,\\
4278 Our house is hell, and thou (a merry devil)\\
4279 Didst rob it of some taste of tediousness,---\\
4280 But fare thee well, there is a ducat for thee,\\
4281 And Launcelot, \edtext{soon}{
4282 \note{early.}}
4283 at supper shalt thou see\\
4284 Lorenzo, who is thy new master's guest,\\
4285 Give him this letter,---do it secretly,---\\
4286 And so farewell: I would not have my father\\
4287 See me \edtext{in}{
4288 \collation{Q; om. F.}}
4289 talk with thee.
4290 \pend
4291
4292 \speaker{Laun.}
4293 \edtext{}{\lemma{\textit{Laun.}}\collation{Q2; Clowne. Q, F.}}%
4294 \edtext{Adieu!}{
4295 \collation{\textnormal{Adiew}, Q, F.}}
4296 tears \edtext{exhibit}{
4297 \note{Eccles paraphrased "My tears serve to express what my
4298 tongue should, if sorrow would permit it," but probably it is
4299 Launce-\lot's blunder for prohibit (Halliwell) or inhibit
4300 (Clarendon).}}
4301 my tongue, most beautiful \edtext{pagan}{
4302 \note{This may have a scurrilous undertone: cf. \textit{2 H 4,}
4303 {\scf II.} \textrm{iii. 168.}}}%
4304 , most sweet \edtext{Jew!}{
4305 \collation{\textnormal{Iewe}, Q, F. \quad \textnormal{do}} Q, F;
4306 \textnormal{did} F2.}}%
4307 ---if a Christian \edtext{do}{
4308 \note{Malone upheld the reading of Qq and F by comparing {\scf II.}
4309 vi. 23: "When you shall please to play the thieves for
4310 wives"; Launcelot seems fond of hinting at what is going to
4311 happen (cf. {\scf II.} v. 22--3). If F2's "did" is accepted,
4312 \textit{get} is used for beget, as in {\scf III.} v. 9.}}
4313 not play the knave and get thee, I am much deceived; but \edtext{adieu!}{
4314 \collation{\textnormal{adiew}, Q, F.}}
4315 these \edtext{foolish drops do \edtext{something}{
4316 \collation{Q; \textnormal{somewhat} F.}}
4317 drown my manly spirit}{
4318 \lemma{foolish\textnormal{\dots}spirit}
4319 \note{"tears do not become a man" (\textit{AYL.}, {\scf III.}
4320 iv. 3); cf. also \textit{H 5,} {\scf IV.} vi. 28--32.}}%
4321 : \edtext{adieu!}{
4322 \collation{\textnormal{adiew}. Q, F. \quad \textnormal{S. D.}} Q2, F; om. Q;
4323 after l. 15 Capell.}}
4324 \exit{Exit.}
4325 \pend

```

```

4326
4327 \speaker{Jes.}
4328 Farewell good Launcelot.\
4329 Alack, what heinous sin is it in me\
4330 To be ashamed to be my father's \edtext{child!}{
4331 \collation{\textnormal{child}, Q, F; \textnormal{Child?} Rowe.}}
4332 \pend
4333 \endnumbering
4334
4335 \end{document}
4336
4337 </arden>

```

---

## B.5 Classical text edition

The next example, which was extracted from a longer file kindly supplied by Wayne Sullivan, University College, Dublin, Ireland, illustrates the use of `ledmac` to produce a Latin text edition, the *Periphyseon*, with Greek passages.<sup>29</sup> The Greek font used is that prepared by Silvio Levy and described in *TUGboat*.<sup>30</sup> The output of this file is shown in Figure 5. Note the use of two layers of footnotes to record testimonia and manuscript readings respectively.

I have converted the original EDMAC example file from TeX to something that looks more like LaTeX.

---

```

4338 (*periph)
4339 % ledmixed.tex
4340 \documentclass{article}
4341 \usepackage{ledmac}
4342
4343 \noendnotes
4344 %% \overfullrule0 pt
4345 \lefthyphenmin=3
4346

```

The LaTeX version uses the `lgreek` package to access Silvio Levy's greek font. The `delims` package option subverts<sup>31</sup> the normal meaning of `$` to switch in and out of math mode. We have to save the original meaning of `$` before calling the package. Later, we use `\Ma` and `\aM` for math mode switching.

```

4347 \let\Ma=$
4348 \let\aM=$
4349 \usepackage[delims]{lgreek}
4350
4351 % We need an addition to \no@expands since the \active $ in lgreek

```

---

<sup>29</sup>The bibliographic details of the forthcoming book are: Iohannis Scotti Erivgenae, *Periphyseon* (*De Divisione Naturae*) Liber Quartus [Scriptores Latini Hiberniae vol. xii], (Dublin: School of Celtic Studies, Dublin Institute for Advanced Studies, forthcoming 1992).

<sup>30</sup>*TUGboat* 9 (1988), pp. 20–24.

<sup>31</sup>It actually changes its category code.

```

4352 % causes problems:
4353 \newcommand{\morenoexpands}{\let$=0}
4354
4355 \makeatletter
4356
4357 \newbox\lp@rbox
4358
4359 \newcommand{\ffootnote}[2][{}]{%
4360 \newcommand{\content}{#2}%
4361 \ifnumberedpar@
4362 \xright@appenditem{\noexpand\vffootnote{f}{\ld@nums}{\csexpandonce{tag}}{\csexpandonce{content}}}{#2}%
4363 \to\inserts@list
4364 \global\advance\insert@count by 1
4365 % \else %% may be used only in numbered text
4366 % \vffootnote{f}{\{0/0/0/0/0/0/0\}{#1}}%
4367 \fi\ignorespaces}
4368
4369 \newcommand{\gfootnote}[2][{}]{%
4370 \newcommand{\content}{#2}%
4371 \ifnumberedpar@
4372 \xright@appenditem{\noexpand\vgfootnote{g}{#1}}{#2}%
4373 \to\inserts@list
4374 \global\advance\insert@count by 1
4375 % \else %% may be used only in numbered text
4376 % \vgfootnote{g}{#1}%
4377 \fi\ignorespaces}
4378
4379 \newcommand{\setlp@rbox}[3]{%
4380 {\parindent\z@\hspace=2.5cm\raggedleft\scriptsize
4381 \baselineskip 9pt%
4382 \global\setbox\lp@rbox=\vbox to\z@{\vss#3}}}
4383
4384 \newcommand{\vffootnote}[2]{\setlp@rbox#2}
4385
4386 \newcommand{\vgfootnote}[2]{\def\rd@ta{#2}}
4387
4388
4389
4390 \renewcommand{\affixline@num}{%
4391 \ifsublines@
4392 \@l@tempcntb=\subline@num
4393 \ifnum\subline@num>\c@firstsublinenum
4394 \@l@tempcnta=\subline@num
4395 \advance\@l@tempcnta by-\c@firstsublinenum
4396 \divide\@l@tempcnta by\c@sublinenumincrement
4397 \multiply\@l@tempcnta by\c@sublinenumincrement
4398 \advance\@l@tempcnta by\c@firstsublinenum
4399 \else
4400 \@l@tempcnta=\c@firstsublinenum
4401 \fi

```

```

4402 %
4403 \ifcase\sub@lock
4404 \or
4405 \ifnum\sublock@disp=1
4406 \@l@tempcntb=0 \@l@tempcnta=1
4407 \fi
4408 \or
4409 \ifnum\sublock@disp=2 \else
4410 \@l@tempcntb=0 \@l@tempcnta=1
4411 \fi
4412 \or
4413 \ifnum\sublock@disp=0
4414 \@l@tempcntb=0 \@l@tempcnta=1
4415 \fi
4416 \fi
4417 \else
4418 \@l@tempcntb=\line@num
4419 \ifnum\line@num>\c@firstlinenum
4420 \@l@tempcnta=\line@num
4421 \advance\@l@tempcnta by-\c@firstlinenum
4422 \divide\@l@tempcnta by\c@linenumincrement
4423 \multiply\@l@tempcnta by\c@linenumincrement
4424 \advance\@l@tempcnta by\c@firstlinenum
4425 \else
4426 \@l@tempcnta=\c@firstlinenum
4427 \fi
4428 \ifcase\@lock
4429 \or
4430 \ifnum\lock@disp=1
4431 \@l@tempcntb=0 \@l@tempcnta=1
4432 \fi
4433 \or
4434 \ifnum\lock@disp=2 \else
4435 \@l@tempcntb=0 \@l@tempcnta=1
4436 \fi
4437 \or
4438 \ifnum\lock@disp=0
4439 \@l@tempcntb=0 \@l@tempcnta=1
4440 \fi
4441 \fi
4442 \fi
4443 %
4444 \ifnum\@l@tempcnta=\@l@tempcntb
4445 \@l@tempcntb=\line@margin
4446 \ifnum\@l@tempcntb>1
4447 \advance\@l@tempcntb by\page@num
4448 \fi
4449 \ifodd\@l@tempcntb
4450 % #1\rlap{\rightlinenum}}%
4451 \xdef\rd@ta{\the\line@num}%

```



```

4452 \else
4453 \llap{{\leftlinenum}}\%#1%
4454 \fi
4455 \else
4456 \%#1%
4457 \fi
4458 \ifcase\@lock
4459 \or
4460 \global\@lock=2
4461 \or \or
4462 \global\@lock=0
4463 \fi
4464 \ifcase\sub@lock
4465 \or
4466 \global\sub@lock=2
4467 \or \or
4468 \global\sub@lock=0
4469 \fi}
4470
4471 \lineation{page}
4472 \linenummargin{right}
4473 \footparagraph{A}
4474 \footparagraph{B}
4475
4476 \renewcommand{\notenumfont}{\footnotesize}
4477 \newcommand{\notetextfont}{\footnotesize}
4478
4479 \let\Afootnoterule=\relax
4480 \count\Afootins=825
4481 \count\Bfootins=825
4482
4483 \newcommand{\Aparafootfmt}[3]{%
4484 \ledsetnormalparstuff
4485 \scriptsize
4486 \notenumfont\printlines#1\enspace
4487 % \lemmafont#1/#2\enskip
4488 \notetextfont
4489 #3\penalty-10\hskip 1em plus 4em minus.4em\relax}
4490
4491 \newcommand{\Bparafootfmt}[3]{%
4492 \ledsetnormalparstuff
4493 \scriptsize
4494 \notenumfont\printlines#1\enspace
4495 \select@lemmafont#1/#2\rbracket\enskip
4496 \notetextfont
4497 #3\penalty-10\hskip 1em plus 4em minus.4em\relax }
4498 \makeatother
4499
4500 \let\Afootfmt=\Aparafootfmt
4501 \let\Bfootfmt=\Bparafootfmt

```

```

4502 \def\lemmafont#1/#2/#3/#4/#5/#6/#7/{\scriptsize}
4503 \parindent=1em
4504
4505 \newcommand{\lmarpar}[1]{\edtext{}{\ffootnote{#1}}}
4506 \newcommand{\rmarpar}[1]{\edtext{}{\gfootnote{#1}}}
4507 \emergencystretch40pt
4508
4509 %%%
4510
4511 \begin{document}
4512
4513 \beginnumbering
4514 \pstart
4515 \rmarpar{741C}
4516 \noindent \edtext{Incipit Quartus $PERIFUSEWN$}{%
4517 \lemma{incipit\ .~.~\ $PERIFUSEWN$}\Bfootnote{\textit{om.\ R},
4518 incipit quartus \textit{M}}}}
4519 \pend
4520 \medskip
4521
4522 \pstart
4523 \noindent \edtext{NVTRITOR}{\lemma{$ANAKEFALIOSIS$}\Bfootnote{\textit{FJP, lege} $<anakefala'iwsis$>.\lmarpar{$ANAKEFALIOSIS$
4524 NATVRARVM} Prima nostrae
4525 \edtext{Physiologiae}{\lemma{physiologiae}\Bfootnote{phisiologiae
4526 \textit{P}, physeologiae \textit{R}}}}
4527 intentio praecipuaque mat\ -e\ -ria erat
4528 \edtext{quod}{\Bfootnote{\textit{p}.\ natura \textit{transp.\ MR}}}}
4529 \edtext{$UPEROUSIADES$}{\Bfootnote{\textit{codd.\ Vtrum}
4530 $<uperousi'wdhs$ (hoc est superessentialis) natura \textit{cum Gale
4531 (p.160) an} $<uperousi'oths$ (hoc est superessentialis natura)
4532 \textit{cum Floss (PL 122,741C) intelligendum sit, ambigitur}}}}
4533 (hoc est superessentialis) natura sit causa creatrix existentium et
4534 non existentium omnium, a nullo creata, unum principium, una
4535 origo, unus et uniuersalis uniuersorum fons, a nullo manans, dum
4536 ab eo manant omnia, trinitas coessentialis in tribus substantiis,
4537 $ANARQOS$ (hoc est sine principio), principium et finis, una
4538 bonitas, deus unus,
4539 \edtext{$OMOUSIOS$}{\Bfootnote{\textit{codd., lege} $<omoo'usios$}}
4540 \edtext{et}{\lemma{\textbf{et}}\Bfootnote{\textit{R}1, \textit{om.\ R}0}}}
4541 $UPEROUSIOS$ (id est coessentialis et superessentialis). Et, ut
4542 ait sanctus Epifanius, episcopus Constantiae Cypri, in
4543 \edtext{$AGKURATW$}{\Bfootnote{anchurato \textit{MR}}}}
4544 sermone
4545 \edtext{de fide}{\Bfootnote{Glo\Ma\langle\am ssa\Ma\rangle\am: Ita
4546 enim uocatur sermo eius de fide $AGKURATOS$, id est procuratus
4547 \textit{mg.\ add.\ FJP}}}:
4548 \begin{itshape}Tria sancta, tria consancta, tria
4549 \edtext{agentia}{\Bfootnote{actiua \textit{MR}}}},

```

```

4552 tria coagentia, tria
4553 \edtext{formantia}{\Bfootnote{formatiua \textit{MR}}},
4554 tria conformantia, tria
4555 \edtext{operantia}{\Bfootnote{operatiua \textit{MR}}},
4556 tria cooperantia, tria subsistentia, tria\rm arpar{742C}
4557 consubsistentia sibi inuicem coexistentia. Trinitas haec
4558 sancta uocatur: tria existentia, una consonantia, una deitas
4559 \edtext{eiusdem}{\Bfootnote{eiusdemque \textit{M}}}}
4560 essentiae,
4561 \edtext{eiusdem uirtutis, eiusdem
4562 \edtext{subsistentiae}{\Bfootnote{substantiae \textit{R}}}}{\%
4563 \Bfootnote{\textit{om. \ M}}},
4564 similia
4565 \edtext{similiter}{\Bfootnote{ex simili \textit{MR}}}}
4566 aequalitatem gratiae operantur patris et filii et sancti spiritus.
4567 Quomodo autem
4568 \edtext{sunt}{\Bfootnote{\textit{om. \ M}}},
4569 ipsis relinquitur docere:
4570 \edtext{'Nemo enim nouit patrem nisi filius, neque filium nisi pater,
4571 et cuicumque filius reuelauerit'}{\Afootnote{Matth.\ 11, 27}};
4572 reuelatur autem per spiritum sanctum. Non ergo haec tria existentia
4573 aut ex ipso aut per ipsum aut ad ipsum in unoquoque digne intelliguntur,
4574 \Ma\mid\! R, 264~{\rm r}\!\mid\!aM\ sicut ipsa reuelant:\end{itshape}
4575 $FWS, PUR, PNEUMA$
4576 \edtext{(hoc est lux, ignis, spiritus)}{\Afootnote{EPIPHANIVS,
4577 \textit{Ancoratus} 67; PG~43, 137C--140A; GCS 25, p.~82, 2--12}}.
4578 \pend
4579
4580 \pstart
4581 Haec, ut dixi, ab Epifanio tradita, ut quisquis interrogatus quae
4582 tria et quid unum in sancta trinitate debeat credere, sana fide
4583 \Ma\!\mid J, 1~{\rm v}\!\mid\!aM\ respondere ualeat, aut ad
4584 fidem accedens\rm arpar{743A} sic erudiatur. Et mihi uidetur
4585 spiritum pro calore posuisse, quasi dixisset in similitudine:
4586 lux, ignis, calor. Haec enim tria unius essentiae sunt. Sed cur
4587 lucem primo dixit, non est mirum. Nam et pater lux est et ignis
4588 et calor; et filius est lux, ignis, calor; et
4589 \edtext{spiritus sanctus}{\Bfootnote{sanctus spiritus \textit{R}}}}
4590 lux, ignis, calor. Illuminat enim pater, illuminat filius, illuminat
4591 spiritus sanctus: ex ipsis enim omnis scientia et sapientia donatur.
4592 \pend
4593 \endnumbering
4594
4595 \end{document}
4596
4597 \periph

```

---

## B.6 Nijmegen

This example, illustrated in Figure 6, was provided in 2004 by Dirk-Jan Dekker of the Department of Medieval History at the University of Nijmegen<sup>32</sup>. Unlike earlier examples, this was coded for LaTeX and ledmac from the start. I have reformatted the example to help it fit this document; any errors are those that I have inadvertently introduced. Note that repeated line numbers are eliminated from the footnotes.

---

```

4598 (*dekker)
4599 %% This is ledekker.tex, a sample critical text edition
4600 %% written in LaTeX2e with the ledmac package.
4601 %% (c) 2003--2004 by Dr. Dirk-Jan Dekker,
4602 %% University of Nijmegen (The Netherlands)
4603 %% (PRW) Modified slightly by PRW to fit the ledmac manual
4604
4605 \documentclass[10pt, letterpaper, onside]{article}
4606 \usepackage[latin]{babel}
4607 \usepackage{ledmac}
4608
4609 \lineation{section}
4610 \linenummargin{left}
4611 \sidenotemargin{outer}
4612
4613 \renewcommand{\notenumfont}{\footnotesize}
4614 \newcommand{\notetextfont}{\footnotesize}
4615
4616 %\let\Afootnoterule=\relax
4617 \let\Bfootnoterule=\relax
4618 \let\Cfootnoterule=\relax
4619
4620 \addtolength{\skip\Afootins}{1.5mm}
4621 %\addtolength{\skip\Bfootins}{1.5mm}
4622 %\addtolength{\skip\Cfootins}{1.5mm}
4623
4624 \makeatletter
4625
4626 \renewcommand*{\para@vfootnote}[2]{%
4627 \insert\cename #1footins\endcename
4628 \bgroup
4629 \notefontsetup
4630 \footsplitskips
4631 \l@dparsefootsspec #2\ledplinenumtrue % new from here
4632 \ifnum\@nameuse{previous@#1@number}=\l@dparsestartline\relax
4633 \ledplinenumfalse
4634 \fi
4635 \ifnum\previous@page=\l@dparsestartpage\relax
4636 \else \ledplinenumtrue \fi
4637 \ifnum\l@dparsestartline=\l@dparseendline\relax

```

---

<sup>32</sup>On 1st September 2004 the University changed its name to Radboud University.

```

4638 \else \ledplinenumtrue \fi
4639 \expandafter\edef\csname previous@#1@number\endcsname{\l@dparsedstartline}
4640 \xdef\previous@page{\l@dparsedstartpage} % to here
4641 \setbox0=\vbox{\hsize=\maxdimen
4642 \noindent\csname #1footfmt\endcsname#2}%
4643 \setbox0=\hbox{\unvvh0}%
4644 \dp0=0pt
4645 \ht0=\csname #1footfudgefactor\endcsname\wd0
4646 \box0
4647 \penalty0
4648 \egroup
4649 }
4650
4651 \newcommand*\previous@A@number}{-1}
4652 \newcommand*\previous@B@number}{-1}
4653 \newcommand*\previous@C@number}{-1}
4654 \newcommand*\previous@page}{-1}
4655
4656 \newcommand{\abb}[1]{#1%
4657 \let\rbracket\nobrak\relax}
4658 \newcommand{\nobrak}{\textnormal{}}
4659 \newcommand{\morenoexpands}{%
4660 \let\abb=0%
4661 }
4662
4663 \newcommand{\Aparafootfmt}[3]{%
4664 \ledsetnormalparstuff
4665 \scriptsize
4666 \notenumfont\printlines#1\enspace
4667 % \lemmafont#1/#2\enskip
4668 \notetextfont
4669 #3\penalty-10\hskip 1em plus 4em minus.4em\relax}
4670
4671 \newcommand{\Bparafootfmt}[3]{%
4672 \ledsetnormalparstuff
4673 \scriptsize
4674 \notenumfont\printlines#1%
4675 \ifledplinenum
4676 \enspace
4677 \else
4678 {\hskip 0em plus 0em minus .3em}
4679 \fi
4680 \select@lemmafont#1/#2\rbracket\enskip
4681 \notetextfont
4682 #3\penalty-10\hskip 1em plus 4em minus.4em\relax }
4683
4684 \newcommand{\Cparafootfmt}[3]{%
4685 \ledsetnormalparstuff
4686 \notenumfont\printlines#1\enspace
4687 % \lemmafont#1/#2\enskip

```

```

4688 \notetextfont
4689 #3\penalty-10\hskip 1em plus 4em minus.4em\relax}
4690
4691 \makeatother
4692
4693 \footparagraph{A}
4694 \footparagraph{B}
4695 \footparagraph{C}
4696
4697 \let\Afootfmt=\Aparafootfmt
4698 \let\Bfootfmt=\Bparafootfmt
4699 \let\Cfootfmt=\Cparafootfmt
4700
4701 \emergencystretch40pt
4702
4703 \author{Guillelmus de Berchen}
4704 \title{Chronicle of Guelders}
4705 \date{}
4706 \hyphenation{archi-epi-sco-po Huns-dis-brug li-be-ra No-vi-ma-gen-si}
4707 \begin{document}
4708 \maketitle
4709 \thispagestyle{empty}
4710
4711 \section*{St.\ Stephen's Church in Nijmegen}
4712 \beginnumbering
4713 \autopar
4714
4715 \noindent
4716 Nobilis itaque comes Otto imperio et dominio Novimagensi sibi,
4717 ut praeferetur, impignoratis et commissis
4718 \edtext{proinde}{\Bfootnote{primum D}} praeesse cupiens, anno
4719 \textsc{liiii}\ledsidenote{1254} superius descripto, mense
4720 Iu\edtext{}{\Afootnote{p.\ 227~R}}nio, una cum iudice, scabinis
4721 ceterisque civibus civitatis Novimagensis, pro ipsius et
4722 inhabitantium in ea necessitate,\edtext{}{\Afootnote{p.\ 97~N}}
4723 commodo et utilitate, ut
4724 \edtext{ecclesia eius}{\Bfootnote{ecclesia D: eius eius H}}
4725 parochialis
4726 \edtext{\abb{extra civitatem}}{\Bfootnote{\textit{om.}~H}} sita
4727 destrueretur et \edtext{infra}{\Bfootnote{intra D}} muros
4728 \edtext{transfer\edtext{}{\Afootnote{p.\ 129~D}}retur}%
4729 {\Bfootnote{transferreretur NH}}
4730 ac de novo construeretur, \edtext{a reverendo patre domino
4731 \edtext{Conrado de \edtext{Hofsteden}}%
4732 {\Bfootnote{Hoffstede D: Hoffsteden H}},
4733 archiepiscopo
4734 \edtext{Coloniensi}{\Bfootnote{Colononiensi H}}}%
4735 {\Cfootnote{Conrad of Hochstaden was archbishop of Cologne in
4736 1238--1261}}, licentiam){\Cfootnote{William is confusing two
4737 charters that are five years apart. Permission from St.\ Apostles'

```

4738 Church in Cologne had been obtained as early as 1249. Cf.\ Sloet,  
 4739 \textit{Oorkondenboek} nr.\ 707 (14 November 1249):  
 4740 ‘‘\ldots{}nos devotionis tue precibus annuentes, ut ipsam  
 4741 ecclesiam faciens demoliri transferas in locum alium competentem,  
 4742 tibi auctoritate presentium indulgemus\ldots{}’’}, et a  
 4743 venerabilibus \edtext{dominis}{\Bfootnote{viris H}} decano et  
 4744 capitulo sanctorum Apostolorum  
 4745 \edtext{Colonien}{\Bfootnote{Coloniae H}}, ipsius ecclesiae ab  
 4746 antiquo veris et pacificis patronis, consensum, citra tamen  
 4747 praeiudicium, damnum aut gravamen  
 4748 \edtext{iurium}{\Bfootnote{virium D}} et bonorum eorundem,  
 4749 impetravit.  
 4750  
 4751 \edtext{Et exinde \edtext{liberum}{\Bfootnote{librum H}} locum  
 4752 eiusdem civitatis \edtext{qui}{\Bfootnote{quae D}} dicitur  
 4753 \edtext{Hundisbrug}{\Bfootnote{Hundisburch D: Hunsdisbrug R}},  
 4754 de praelibati Wilhelmi Romanorum  
 4755 \edtext{regis}{\Bfootnote{imperatoris D}}, ipsius fundi  
 4756 do\edtext{}{\Afootnote{f.\ 72v~M}}mini, consensu, ad aedificandum  
 4757 \edtext{\abb{et consecrandum}}{\Bfootnote{\textit{om.}\ H}}  
 4758 ecclesi\edtext{}{\Afootnote{p.\ 228~R}}am et coemeterium,  
 4759 \edtext{eisdem}{\Bfootnote{eiusdem D}} decano et capitulo de  
 4760 expresso eiusdem civitatis assensu libera contradiderunt voluntate,  
 4761 obligantes se ipsi \edtext{comes}{\Bfootnote{comites D}} et civitas  
 4762 \edtext{\abb{dictis}}{\Bfootnote{\textit{om.}\ H}} decano et  
 4763 capitulo, quod in recompensationem illius areae infra castrum et  
 4764 portam, quae fuit dos ecclesiae, in qua plebanus habitare  
 4765 solebat---quae \edtext{tunc}{\Bfootnote{nunc H}} per novum fossatum  
 4766 civitatis est destructa---aliam aream competentem et ecclesiae  
 4767 novae,  
 4768 \edtext{ut praefertur, aedificandae}{\lemma{\abb{ut\ldots aedificandae}}}%  
 4769 \Bfootnote{\textit{om.}\ H}} satis  
 4770 \edtext{contiguam}{\Bfootnote{contiguum M}}, ipsi plebano darent  
 4771 et assignarent. Et desuper  
 4772 \edtext{\abb{apud}}{\Bfootnote{\textit{om.}\ H}} dictam ecclesiam  
 4773 sanctorum Apostolorum \edtext{est}{\Bfootnote{et H}}  
 4774 \edtext{littera}{\Bfootnote{litteram H}} sigillis ipsorum  
 4775 Ottonis\edtext{}{\Afootnote{p.\ 130~D}} comitis et civitatis  
 4776 \edtext{Novimagensis}{\Bfootnote{Novimagii D}}  
 4777 \edtext{sigillata}{\Bfootnote{sigillis communita H}}.}%  
 4778 {\Cfootnote{Cf.\ Sloet, \textit{Oorkondenboek} nr.\ 762 (June 1254)}}  
 4779  
 4780 % (PRW) the full document continues on after this point  
 4781 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
 4782 \endnumbering  
 4783 \end{document}  
 4784 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
 4785  
 4786 </dekker>

---

## B.7 Irish verse

This example, illustrated in Figure 7, is a somewhat modified and shortened version of Wayne Sullivan's example demonstration for EDSTANZA.

The stanza lines are numbered according to the source verse lines, not according to the printed lines. For example, the sixth ('f') line in the first stanza is printed as two lines as the source line was too long to fit on one printed line. Note that if you process this yourself you will get error reports about counters the first time through; this is because alphabetic counters, like roman numerals, have no notion of zero.

As is fairly typical of critical edition typesetting, some of ledmac's internal macros had to be modified to get the desired effects.

---

```

4787 (*braonain)
4788 %% This is ledbraonain.tex, a sample critical verse edition.
4789 %% Originally written for TeX processing with edmac and edstanza
4790 %% by Wayne Sullivan.
4791 %% Extensively modified by Peter Wilson for LaTeX and the ledmac package.
4792
4793 \documentclass{article}
4794 \usepackage{ledmac}
4795
4796 \setlength{\textheight}{40pc}
4797 \setlength{\textwidth}{24pc}
4798 \bigskipamount=12pt plus 6pt minus 6pt
4799 \newcommand*{\notetextfont}{\footnotesize}
4800
4801 %% Just one footnote series
4802 \footparagraph{C}
4803 \count\Cfootins=800
4804 \makeatletter
4805 %% but using two different formats
4806 \def\xparafootfmt#1#2#3{%
4807 \ledsetnormalparstuff
4808 {\notenumfont\printlines#1}\enspace
4809 %% {\select@lemmafонт#1/#2}\rbracket\enskip
4810 \notetextfont #3\penalty-10 }
4811 \def\yparafootfmt#1#2#3{%
4812 \ledsetnormalparstuff
4813 %% {\notenumfont\printlines#1}\enspace
4814 %% {\select@lemmafонт#1/#2}\rbracket\enskip
4815 \notetextfont #3\penalty-10 }
4816
4817 \let\Cfootfmt=\xparafootfmt
4818 \skip\Cfootins=\bigskipamount
4819 \makeatother
4820
4821 %% This is the default, but just to demonstrate...
4822 \setlength{\stanzaindentbase}{20pt}
4823

```



```

4824 %% MUST SET THE INDENTS
4825 %% indent multiples; first=hangindent.
4826 %% Must all be non-negative whole numbers
4827 \setstanzaindents{4,1,2,1,2,3,3,1,2,1}
4828
4829 %% Set stanza line penalties
4830 %% Must be nonnegative whole numbers.
4831 %% An initial zero indicates no penalties.
4832 \setstanzapenalties{1,5000,10500,5000,10500,5000,5000,5000,0}
4833 \%setstanzapenalties{0}% the default
4834
4835 %% Put some space between stanzas
4836 \let\endstanzaextra=\bigbreak % ==> \bigskip \penalty -200
4837
4838 %% (almost) force line break in foot paragraph
4839 \mathchardef\IMM=9999
4840 \def\lbreak{\hfil\penalty-\IMM}
4841
4842 %% Number each stanza in bold
4843 \newcounter{stanzanum}
4844 \setcounter{stanzanum}{0}
4845 \newcommand*{\numberit}{%
4846 \flagstanza[0.5\stanzaindentbase]{\textbf{\thestanzanum}}}
4847 %% Use the hook to insert the number (and counteract a new line)
4848 %% and reset the line number to zero
4849 \newcommand*{\startstanzahook}{\refstepcounter{stanzanum}%
4850 \numberit\vskip-\baselineskip%
4851 \setlinenum{0}}
4852
4853 %% Want to label the footnotes with the stanza and line number
4854 %% We'll use \linenum to replace the sub-line number
4855 %% with the stanza number, redefining \edtext to do this
4856 %% automatically for us.
4857 %%%
4858 \makeatletter
4859
4860 \renewcommand{\edtext}[2]{\leavevmode
4861 \begingroup
4862 \xdef\@tag{#1}%
4863 \set@line
4864 \global\insert@count=0
4865 \ignorespaces \linenum{\\the\c@stanzanum}#2\relax
4866 \flag@start
4867 \endgroup
4868 #1%
4869 \ifx\end@lemmas\empty \else
4870 \gl@p\end@lemmas\to\x@lemma
4871 \x@lemma
4872 \global\let\x@lemma=\relax
4873 \fi

```

```

4874 \flag@end}
4875
4876 %% We need only a very simple macro for footnote numbers,
4877 %% to produce the stanza number (sub-line) then the line number.
4878 \def\printstanzalines#1/#2/#3/#4/#5/#6/#7/{\begingroup
4879 #3\fullstop \linenumrep{#2}
4880 \endgroup}
4881 \let\oldprintlines\printlines
4882
4883 \makeatother
4884 %%%%%%%%%%%%%%%
4885
4886 \pagestyle{empty}
4887
4888 \begin{document}
4889
4890 \beginnumbering
4891
4892 \pstart \centering \textbf{22} \pend
4893
4894 \bigskip
4895 %% do not print line number beside heading
4896 \setcounter{firstlinenum}{1000}
4897 %% and heading footnotes use a different format
4898 \let\Cfootfmt=\yparafootfmt
4899
4900 \pstart
4901 \centerline{[Se\'an \'O Braon\'ain cct] chuim Tom\'ais U\'{i}
4902 \edtext{Dh\'unlaing}{\Cfootnote{\textbf{22} \textit{Teideal}: Dhuinnluing T,
4903 Se\'aghan Mac Domhnaill cct B\lbreak}}}
4904 \pend
4905
4906 \pstart
4907 \centerline{[Fonn: M\'airse\'ail U\'{i} Sh\'uilleabh\'ain (P\'ainseach
4908 na nUbh]}
4909 \pend
4910
4911 \bigskip
4912
4913 %% revert to the regular footnote format
4914 \let\Cfootfmt=\xparafootfmt
4915 %% but use our special number printing routine
4916 \let\printlines\printstanzalines
4917 %% Use letters for line numbering
4918 \linenumberstyle{alph}
4919 %% number lines from the second onwards
4920 \setcounter{firstlinenum}{2}
4921 \setcounter{linenumincrement}{1}
4922
4923 %% Each verse starts with \stanza.

```

```

4924 %% Lines end with &; the last line with \&.
4925
4926 \stanza
4927 A \edtext{dhuine}{\Cfootnote{dhuinne T}} gan ch\’eill do
4928 \edtext{mhaisligh}{\Cfootnote{mhaslaidh T, mhaslaig B}} an chl\’eir&
4929 is tharcaisnigh naomhscruipt na bhf\’aige,&
4930 na haitheanta \edtext{r\’eab}{\Cfootnote{raob T}} ’s an
4931 t-aifreann thr\’eig&
4932 \edtext{re}{\Cfootnote{le B}} taithneamh do chlaonchreideamh
4933 Mh\’artain,&
4934 c\’a rachair \edtext{’od}{\Cfootnote{dod B}} dh\’{\i}on ar
4935 \’Iosa Nasardha&
4936 nuair \edtext{chaithfimid}{\Cfootnote{chaithfam\’{\i}d T}} cruinn
4937 bheith ar \edtext{mhaoileann}{\Cfootnote{maoilinn B}} Josepha?&
4938 N\’{\i} caraoid Mac Crae chuim t’anama ’
4939 \edtext{phl\’e}{\Cfootnote{phleidl T}}&
4940 n\’a Calvin \edtext{bhiais}{\Cfootnote{bh\’{\i}os B}} taobh
4941 \edtext{ris}{\Cfootnote{leis B}} an l\’a sin.\&
4942
4943 \stanza
4944 N\’ach damanta an sc\’eal don chreachaire
4945 \edtext{chlaon}{\Cfootnote{claon B}}&
4946 ghlac baiste na cl\’eire ’na ph\’aiste&
4947 ’s do \edtext{glanadh}{\Cfootnote{glannuig T}} mar ghr\’ein
4948 \’on bpeaca r\’o-dhaor&
4949 tr\’{\i} \edtext{ainibhfios}{\Cfootnote{ainnibhfios T, ainnbhfios B}}
4950 \edtext{\’Eva}{\Cfootnote{\’Eabha B}} rinn \’Adam,&
4951 tuitim ar\’{\i}s f\’e chuing na haicme sin&
4952 tug atharrach br\’{\i} don scr\’{\i}bhinn bheannaithe,&
4953 d’aistrigh b\’easa \edtext{agus}{\Cfootnote{is B}} reachta na cl\’eire&
4954 ’s n\’ach \edtext{tugann}{\Cfootnote{tuigionn T}} aon
4955 gh\’eilleadh don Ph\’apa?\&
4956
4957 \stanza
4958 Gach \edtext{scolaire}{\Cfootnote{sgollaire T}} baoth, n\’{\i}
4959 \edtext{mholaim}{\Cfootnote{mholluim T}} a cheird&
4960 \edtext{’t\’a ag obair}{\Cfootnote{’t\’ag cobar T}}
4961 \edtext{le}{\Cfootnote{re B}} g\’eilleadh d\’a th\’aille&
4962 don \edtext{doirbhchoin chlaon}{\Cfootnote{dorbhchon daor B}}
4963 d\’a ngorthar Mac Crae,&
4964 deisceabal \edtext{straeigh}{\Cfootnote{straodhaig T}} as an
4965 gcoll\’aiste.&
4966 T\’a \edtext{\edtext{adaithe}}{\Cfootnote{fadaighthe B}}
4967 th\’{\i}os{\Cfootnote{fhadoghthe ts\’{\i}os T}} in
4968 \’{\i}ochtar ifrinn,&
4969 gan \edtext{solas}{\Cfootnote{sollus T}} gan soilse i
4970 dt\’{\i}orthaibh dorchu,&
4971 tuigsint an l\’einn, gach
4972 \edtext{cuirpeacht}{\Cfootnote{cuirripeacht T}} d\’ein&
4973 is \edtext{Lucifer}{\Cfootnote{Luicifer T, L\’ucifer B}} aosta

```

```
4974 'na \edtext{mh\'aistir}{\Cfootnote{mhaighistir T}}.\&
4975
4976 \endnumbering
4977
4978 \end{document}
4979
4980 </braonain>
```

---

## 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 LaTeX*. February 2003. (Available from CTAN in `macros/latex/contrib/footmisc`)
- [LW90] John Lavagnino and Dominik Wujastyk. ‘An overview of EDMAC: a PLAIN TeX format for critical editions’. *TUGboat*, **11**, 4, pp. 623–643, November 1990. (Code available from CTAN in `macros/plain/contrib/edmac`)
- [Lüc03] Uwe Lück. ‘ednotes — critical edition typesetting with LaTeX’. *TUGboat*, **24**, 2, pp. 224–236, 2003. (Code available from CTAN in `macros/latex/contrib/ednotes`)
- [Sul92] Wayne G. Sullivan. *The file edstanza.doc*. June 1992. (Available from CTAN in `macros/plain/contrib/edmac`)
- [Wil02] Peter Wilson. *The memoir class for configurable typesetting*. November 2002. (Available from CTAN in `macros/latex/contrib/memoir`)
- [Wil04] Peter Wilson. *Parallel typesetting for critical editions: the ledpar package*. December 2004. (Available from CTAN in `macros/latex/contrib/ledmmac`)
- [Wil05] Peter Wilson. *Critical editions and arabic typesetting: the ledarab and afoot packages*. February 2005. (Available from CTAN in `macros/latex/contrib/ledmmac`)

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

| Symbols                                                                                     |                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\&amp;</code> 18, 3147, 3151, 3152, 3201, 3210,<br>3216, 3218, 4924, 4941, 4955, 4974 | <code>\@ifnextchar</code> ..... 3048, 3321                                                                                                                                                                                                                                                                                                                             |
| <code>\-</code> ..... 4299, 4528                                                            | <code>\@ifpackageloaded</code> ..... 3096                                                                                                                                                                                                                                                                                                                              |
| <code>\@@line</code> ..... 1598                                                             | <code>\@iiiminipage</code> ..... 2959                                                                                                                                                                                                                                                                                                                                  |
| <code>\@@wrindexm@</code> .. 3068, 3073, 3076, 3100                                         | <code>\@iiiparbox</code> ..... 2986                                                                                                                                                                                                                                                                                                                                    |
| <code>\@EDROWFILL@</code> ..... 3439, <u>3646</u>                                           | <code>\@indexfile</code> ..... 3090                                                                                                                                                                                                                                                                                                                                    |
| <code>\@M</code> ..... 1598, 3195, 3205                                                     | <code>\@inputcheck</code> ..... 338                                                                                                                                                                                                                                                                                                                                    |
| <code>\@MM</code> ..... 1315                                                                | <code>\@insert</code> ..... 1106–1108, 1142–1144                                                                                                                                                                                                                                                                                                                       |
| <code>\@adv</code> ..... <u>451</u> , 612                                                   | <code>\@k</code> ..... <u>1596</u>                                                                                                                                                                                                                                                                                                                                     |
| <code>\@arabic</code> ..... 738                                                             | <code>\@kludgeins</code> ..... 2064, 2132                                                                                                                                                                                                                                                                                                                              |
| <code>\@aux</code> ..... 2237                                                               | <code>\@l</code> ..... <u>368</u> , 595                                                                                                                                                                                                                                                                                                                                |
| <code>\@auxout</code> . 2239, 3067, 3072, 3075, 3099                                        | <code>\@l@dttempcnta</code> ... <u>21</u> , 485, 487, 489,<br>490, 890, 891, 893, 895, 898,<br>899, 914, 950–954, 956, 963–<br>967, 969, 972, 975, 977, 981,<br>1011, 1015, 1019, 1026, 1030,<br>1034, 1115, 1119, 1123, 1126,<br>1129, 1132, 1133, 4394–4398,<br>4400, 4406, 4410, 4414, 4420–<br>4424, 4426, 4431, 4435, 4439, 4444                                  |
| <code>\@botlist</code> ..... 2190, 2192                                                     | <code>\@l@dttempcntb</code> <u>21</u> , 205, 206, 211, 215,<br>219, 223, 226, 249, 250, 257,<br>261, 265, 267, 275, 276, 948,<br>960, 981, 989–991, 993, 1011,<br>1015, 1019, 1026, 1030, 1034,<br>1064–1066, 1068, 1121, 1122,<br>2417, 2418, 2423, 2427, 2431,<br>2435, 2438, 2520–2522, 2524,<br>4392, 4406, 4410, 4414, 4418,<br>4431, 4435, 4439, 4444–4447, 4449 |
| <code>\@cclv</code> 2085, 2089, 2090, 2188, 2189, 2217                                      | <code>\@l@reg</code> ..... 368                                                                                                                                                                                                                                                                                                                                         |
| <code>\@checkend</code> ..... 3128                                                          | <code>\@lab</code> ..... 568, 2228, <u>2274</u>                                                                                                                                                                                                                                                                                                                        |
| <code>\@colht</code> ..... 2067, 2193, 2205                                                 | <code>\@latexerr</code> ..... 2196                                                                                                                                                                                                                                                                                                                                     |
| <code>\@colroom</code> ..... 2193                                                           | <code>\@led@extranofeet</code> .. <u>2157</u> , 2170, 2183                                                                                                                                                                                                                                                                                                             |
| <code>\@combinefloats</code> ..... 2062                                                     | <code>\@led@nofootfalse</code> ..... 2174–2182                                                                                                                                                                                                                                                                                                                         |
| <code>\@currenvir</code> ..... 3113, 3116, 3117                                             | <code>\@led@nofoottrue</code> ..... 2173                                                                                                                                                                                                                                                                                                                               |
| <code>\@currlist</code> ..... 2194, 2197                                                    | <code>\@led@testifnofoot</code> ..... <u>2172</u>                                                                                                                                                                                                                                                                                                                      |
| <code>\@dbldeferlist</code> ..... 2203, 2208, 2210                                          | <code>\@line@@num</code> ..... <u>3337</u> , 3451                                                                                                                                                                                                                                                                                                                      |
| <code>\@dblfloatplacement</code> ..... 2207                                                 | <code>\@listdepth</code> ..... 2972                                                                                                                                                                                                                                                                                                                                    |
| <code>\@dbltoplist</code> ..... 2203, 2204                                                  | <code>\@lock</code> ..... 134, <u>326</u> , 393,<br>395, 397, 410, 518, 519, 521,<br>522, 538, 539, 541, 823, 861,<br>920, 922, 923, 925, 1023, 1038,<br>1040, 1042, 4428, 4458, 4460, 4462                                                                                                                                                                            |
| <code>\@deferlist</code> ..... 2190, 2199, 2200                                             |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@docclearpage</code> ..... <u>2172</u>                                               |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@edrowfill@</code> ..... <u>3646</u>                                                 |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@ehb</code> ..... 2196                                                               |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@emptytoks</code> ..... <u>3104</u> , 3114                                           |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@footnotemark</code> ..... <u>2551</u>                                               |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@footnotemarkA</code> ..... 2584, 2596                                               |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@footnotemarkB</code> ..... 2869, 2880                                               |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@footnotemarkC</code> ..... 2887, 2896                                               |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@footnotetext</code> .....<br>.... <u>2547</u> , <u>2564</u> , 2970, 2996, 3023      |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@freelist</code> ..... 2060                                                          |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@gobble</code> 644, 645, 2359, 3125, 3408, 3425                                      |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@gobblethree</code> ..... <u>2351</u>                                                |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@h</code> ..... <u>1596</u>                                                          |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@hilfs@count</code> ..... <u>3341</u>                                                |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@idxfile</code> 3058, 3068, 3073, 3076, 3100                                         |                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\@ifclassloaded</code> .....<br>..... 23, 2135, 2160, 2546, 3045                      |                                                                                                                                                                                                                                                                                                                                                                        |

- \@lopL ..... 435
  - \@lopR ..... 435
  - \@makecol ..... 2139
  - \@makefcolumn .. 2199, 2200, 2208, 2210
  - \@makespecialcolbox ..... 2065
  - \@maxdepth ..... 2080, 2088
  - \@mem@extranofeet ..... 2161
  - \@mem@nofootfalse ..... 2162–2169
  - \@midlist ..... 2060, 2061
  - \@minipagefalse ..... 2983
  - \@minipagerestore ..... 2973
  - \@minus ..... 1470, 1475, 2689, 2695
  - \@mpargs ..... 2963, 2986
  - \@mpfn ..... 2969, 2995, 3022
  - \@mpfootins 2979, 2989, 2992, 3001, 3028
  - \@mpfootnotetext ... 2970, 2996, 3023
  - \@mplistdepth ..... 2972
  - \@nameuse ..... 282,  
284, 1320, 1321, 1451, 1453,  
1524, 1525, 1564, 1566, 1650,  
1652, 1697, 1699, 2146, 2147,  
2149, 2150, 2152, 2154, 2613,  
2617, 2619, 2623, 2626, 2627,  
2632, 2636, 2640, 2643, 2646,  
2651, 2654, 2656, 2657, 2661,  
2668, 2674, 2719, 2729, 2737,  
2739, 2764, 2774, 2782, 2784,  
2811, 2815, 2816, 2824, 2832,  
2833, 2837, 2847, 2857, 2859,  
2923, 2924, 2926, 2927, 2932, 4632
  - \@nobreakfalse ..... 745
  - \@nobreaktrue ..... 743, 747
  - \@nowrindex ..... 3057
  - \@oldnobreak ..... 743, 745, 784
  - \@opcol ..... 2200, 2218
  - \@opxtrafeetii ..... 2109, 2110, 2145
  - \@outputbox .....  
. 2067, 2069, 2070, 2085, 2087,  
2107, 2108, 2905, 2906, 2921, 2922
  - \@outputpage ..... 2209
  - \@page ..... 415
  - \@parboxrestore .... 1324, 2630, 2968
  - \@pboxswfalse ..... 2961
  - \@pend ..... 435
  - \@pendR ..... 435
  - \@plus ..... 1330,  
1470, 1475, 2689, 2695, 2725, 2770
  - \@ref ..... 553, 598
  - \@ref@reg ..... 555
  - \@reinsets ..... 2140
  - \@set ..... 466, 617
  - \@setminipage ..... 2974
  - \@showidx ..... 3065, 3098
  - \@tag ..... 652, 685, 713,  
2333, 2337, 2341, 2345, 2349, 4862
  - \@tempboxa .... 2188, 2189, 2964, 2986
  - \@tempdima ..... 2089, 2962, 2966
  - \@templ@d ..... 2511, 2512
  - \@textbottom ..... 2072
  - \@texttop ..... 2068
  - \@thefnmarkA ..... 2583, 2595
  - \@thefnmarkB ..... 2868, 2879
  - \@thefnmarkC ..... 2886, 2895
  - \@toksa ..... 309, 317
  - \@toksb ..... 309, 316–318
  - \@toplist ..... 2190, 2191
  - \@whilenum ..... 3620
  - \@whilesw ..... 2200, 2209
  - \@wredindex ..... 3087, 3089, 3328
  - \@x@sf 2540, 2543, 2554, 2560, 2603, 2609
  - \@xloop ..... 1140, 1147
  - \@xympar ..... 2405
  - \^ ..... 362, 3226
  - \\_ ... 1426, 3654, 3659, 3667, 4000,  
4035, 4158, 4161, 4517, 4529,  
4530, 4542, 4549, 4563, 4568,  
4571, 4574, 4583, 4711, 4720,  
4722, 4728, 4737–4739, 4756–  
4758, 4762, 4769, 4772, 4775, 4778
- A**
- \A@@footnote ... 3332, 3410, 3427, 3446
  - \abb ..... 4656,  
4660, 4726, 4757, 4762, 4768, 4772
  - \absline@num ... 131, 325, 373, 376,  
379, 480, 483, 492, 506, 528,  
550, 560, 851, 873, 874, 882, 1105
  - Abu Kamil Shuja' b. Aslam ..... 5
  - \actionlines@list .....  
..... 328, 345, 348, 355, 480,  
483, 492, 506, 528, 550, 903, 906
  - \actions@list .....  
. 328, 349, 356, 481, 490, 494,  
496, 508, 517, 530, 537, 551, 907
  - \add@inserts ..... 830, 1094
  - \add@inserts@next ..... 1094
  - \add@penalties ..... 1115
  - \addfootins ..... 2143

- \addfootinsX . . . . . 2918
  - \addtocounter . . . . . 785
  - \addtol@denvbody . . . 3108, 3129, 3131
  - \addtolength . . . . . 4620–4622
  - Adelard II . . . . . 4
  - \advancelabel@refs . . . . . 2234, 2243
  - \advanceline . . 10, 57, 60, 612, 635, 645
  - \advancepageno . . . . . 2055
  - \Aend . . . . . 2331, 2351
  - \Aendnote . . . . . 12, 2330
  - \affixline@num . . . . . 828, 942, 4390
  - \affixpstart@num . . . . . 829, 1053
  - \affixside@note . . . . . 830, 2510
  - \Afootfmt . . . . .
    - 4101, 4238, 4252, 4271, 4500, 4697
  - \Afootgroup . . . . . 2111
  - \Afootins . . . . . 1458, 2111, 2122,
    - 2162, 2175, 4102–4104, 4480, 4620
  - \Afootnote . . . . . 11,
    - 1171, 2938, 3332, 3402, 3410,
    - 3419, 3427, 3446, 3895, 3902,
    - 3988, 3991, 3993, 4024, 4026,
    - 4099, 4128, 4154, 4156–4158,
    - 4161, 4172, 4240, 4571, 4576,
    - 4720, 4722, 4728, 4756, 4758, 4775
  - \Afootnoterule . . . . . 4207, 4479, 4616
  - \Afootstart . . . . . 2111, 4105
  - \afterlemmaseparator . . . . . 15, 1912
  - \afterlemmaseparator@A . . . . . 1912
  - \afterlemmaseparator@B . . . . . 1913
  - \afterlemmaseparator@C . . . . . 1914
  - \afterlemmaseparator@D . . . . . 1915
  - \afterlemmaseparator@E . . . . . 1916
  - \afterlemmaseparator@Z . . . . . 1917
  - \afternote . . . . . 15, 1958
  - \afternote@A . . . . . 1958
  - \afternote@B . . . . . 1959
  - \afternote@C . . . . . 1960
  - \afternote@D . . . . . 1961
  - \afternote@E . . . . . 1962
  - \afternote@Z . . . . . 1963
  - \afternumberinfootnote . . . . . 14, 1791
  - \afternumberinfootnote@A . . . . . 1791, 1831
  - \afternumberinfootnote@B . . . . . 1792, 1832
  - \afternumberinfootnote@C . . . . . 1793, 1833
  - \afternumberinfootnote@D . . . . . 1794, 1834
  - \afternumberinfootnote@E . . . . . 1795, 1835
  - \afternumberinfootnote@Z . . . . . 1796, 1836
  - \aftersymmlinenumber . . . . . 14, 1831
  - \aftersymmlinenumber@A . . . . . 1831
  - \aftersymmlinenumber@B . . . . . 1832
  - \aftersymmlinenumber@C . . . . . 1833
  - \aftersymmlinenumber@D . . . . . 1834
  - \aftersymmlinenumber@E . . . . . 1835
  - \aftersymmlinenumber@Z . . . . . 1836
  - \allowbreak . . . . .
    - 1644, 1687, 2730, 2775, 3946, 3959
  - \alph . . . . . 3872
  - \alpha . . . . . 3993, 4028
  - \aM . . . . . 4348, 4547, 4574, 4583
  - \ampersand . . . . . 19, 3147, 3218
  - \Aparafootfmt . . . . . 4483, 4500, 4663, 4697
  - \AtBeginDocument . . . . . 2270, 3096
  - \author . . . . . 3879, 4703
  - \autopar . . . . . 7, 78, 792, 4713
  - \autoparfalse . . . . . 160, 793
  - \autopartrue . . . . . 806
- B**
- \B@@footnote . . . 3332, 3411, 3428, 3447
  - \ballast . . . . . 30
  - \ballast@count . . . 868, 871, 876, 1115
  - Beeton, Barbara Ann Neuhaus Friend . . . 7
  - \beforelemmaseparator . . . . . 14, 1892
  - \beforelemmaseparator@A . . . . . 1892
  - \beforelemmaseparator@B . . . . . 1893
  - \beforelemmaseparator@C . . . . . 1894
  - \beforelemmaseparator@D . . . . . 1895
  - \beforelemmaseparator@E . . . . . 1896
  - \beforelemmaseparator@Z . . . . . 1897
  - \beforenumberinfootnote . . . 14, 1771
  - \beforenumberinfootnote@A . . . . . 1771, 1811
  - \beforenumberinfootnote@B . . . . . 1772, 1812
  - \beforenumberinfootnote@C . . . . . 1773, 1813
  - \beforenumberinfootnote@D . . . . . 1774, 1814
  - \beforenumberinfootnote@E . . . . . 1775, 1815
  - \beforenumberinfootnote@Z . . . . . 1776, 1816
  - \beforesymmlinenumber . . . . . 14, 1811
  - \beforesymmlinenumber@A . . . . . 1811
  - \beforesymmlinenumber@B . . . . . 1812
  - \beforesymmlinenumber@C . . . . . 1813
  - \beforesymmlinenumber@D . . . . . 1814
  - \beforesymmlinenumber@E . . . . . 1815
  - \beforesymmlinenumber@Z . . . . . 1816
  - \beginnumbering . . . . .
    - . . 6, 115, 173, 750, 803, 3890,
    - 3984, 4126, 4254, 4513, 4712, 4890
  - \beginnumberingR . . . . . 798
  - \Bend . . . . . 2335, 2351
  - \Bendnote . . . . . 12, 2330



- \beta ..... 3993, 4028
  - \Bfootfmt ..... 4239, 4501, 4698
  - \Bfootgroup ..... 2112
  - \Bfootins ..... 1458,  
2112, 2123, 2163, 2176, 4481, 4621
  - \Bfootnote .. 11, 1183, 2939, 3333,  
3403, 3411, 3420, 3428, 3447,  
3896, 3995, 4028, 4030, 4038,  
4040, 4042, 4044, 4046, 4048,  
4050, 4052, 4054, 4241, 4517,  
4523, 4526, 4529, 4530, 4540,  
4541, 4545, 4547, 4551, 4553,  
4555, 4559, 4562, 4563, 4565,  
4568, 4589, 4718, 4724, 4726,  
4727, 4729, 4732, 4734, 4743,  
4745, 4748, 4751–4753, 4755,  
4757, 4759, 4761, 4762, 4765,  
4769, 4770, 4772–4774, 4776, 4777
  - \Bfootnoterule ..... 4207, 4617
  - \Bfootstart ..... 2112
  - \bfseries ..... 738, 3970
  - \bigbreak ..... 4836
  - \bigskip . 4263, 4269, 4836, 4894, 4911
  - \bigskipamount ..... 4798, 4818
  - \body ..... 1148, 1149, 3149, 3217
  - \bodyfootmarkA ..... 24
  - \box 842, 844, 1519, 1533, 1578, 1597,  
2085, 2189, 2217, 2828, 2841, 4646
  - \boxmaxdepth ..... 2088
  - \Bparafootfmt .. 4491, 4501, 4671, 4698
  - Bredon, Simon ..... 4
  - Breger, Herbert ..... 2, 5, 155
  - Brey, Gerhard ..... 4, 5
  - \brokenpenalty ..... 788
  - Burt, John ..... 3
  - Busard, Hubert L. L. .... 4
  - \bypage@false ..... 176, 189, 194
  - \bypage@true ..... 176, 184
  - \bypstart@false ..... 176, 185, 195
  - \bypstart@true ..... 176, 190
- C**
- \C@footnote ... 3332, 3412, 3429, 3448
  - \c@addcolcount ..... 3614
  - \c@ballast ..... 868, 876
  - \c@firstlinenum ... 232, 962, 964,  
967, 969, 4419, 4421, 4424, 4426
  - \c@firstsublinenum . 236, 949, 951,  
954, 956, 4393, 4395, 4398, 4400
  - \c@footnoteA ..... 2588
  - \c@footnoteB ..... 2872
  - \c@footnoteC ..... 2889
  - \c@labidx ..... 3034
  - \c@linenumincrement .....  
..... 232, 965, 966, 4422, 4423
  - \c@mpfootnote ..... 2969, 2995, 3022
  - \c@page ..... 595
  - \c@pstart ..... 738
  - \c@stanzanum ..... 4865
  - \c@sublinenumincrement .....  
..... 236, 952, 953, 4396, 4397
  - \Cend ..... 2339, 2351
  - \Cendnote ..... 12, 2330
  - \centering ..... 4892
  - \centerline ..... 3550,  
3555, 3561, 3566, 3572, 3577,  
3798, 3800, 4256, 4266, 4901, 4907
  - \Cfootfmt ..... 4699, 4817, 4898, 4914
  - \Cfootgroup ..... 2113
  - \Cfootins ..... 1459, 2113,  
2124, 2164, 2177, 4622, 4803, 4818
  - \Cfootnote ..... 11, 1183,  
2940, 3334, 3404, 3412, 3421,  
3429, 3448, 4001, 4036, 4735,  
4736, 4778, 4902, 4927, 4928,  
4930, 4932, 4934, 4936, 4937,  
4939–4941, 4945, 4947, 4949,  
4950, 4953, 4954, 4958–4962,  
4964, 4966, 4967, 4969, 4972–4974
  - \Cfootnoterule ..... 4618
  - \Cfootstart ..... 2113
  - \ch@ck@l@ck ..... 979, 1007
  - \ch@cksub@l@ck ..... 958, 1007
  - \char ..... 3147
  - \chardef ..... 2404, 3149, 3151
  - Chester, Robert of ..... 4
  - Claassens, Geert H. M. .... 5
  - class 1 feet ..... 117, 143
  - class 2 feet ..... 117, 118
  - \cleaders ..... 3609
  - \closeout ..... 587, 591, 2322
  - \clubpenalty ..... 788, 1119
  - \collation ..... 4240, 4258,  
4260, 4269, 4276, 4288, 4293,  
4295, 4305, 4314, 4316, 4322, 4331
  - \color@begingroup .....  
1325, 1529, 2092, 2631, 2837, 2965
  - \color@endgroup .....  
1326, 1529, 2096, 2632, 2837, 2984

- `\columnwidth` ..... 1323, 1500, 2629, 2967, 3010
- `\content` .. 1172, 1184, 1197, 1210, 1223, 1243, 1256, 1269, 1282, 1295, 2463, 2470, 2477, 4360, 4370
- Copernicus, Nicolaus ..... 4
- `\copy` ..... 4114
- `\count` ..... 1468, 1473, 1490, 1494, 1620, 1623, 1668, 1694, 2687, 2692, 2708, 2711, 2754, 2757, 2796, 2799, 4480, 4481, 4803
- `\countdef` ..... 2055
- `\Cparafootfmt` ..... 4684, 4699
- `\cr` ..... 1599, 1602
- `\CRITEXT` ..... 3312
- `\critext` ..... 34, 646, 655, 668, 3314, 3401, 3443
- `\cs` ..... 656, 661, 665, 1419, 1427, 1431, 1439–1441, 1749, 1936, 1982, 2013, 2243–2245
- `\csdef` 1735–1740, 1743, 1780–1785, 1788, 1800–1805, 1808, 1820–1825, 1828, 1840–1845, 1848, 1860–1865, 1868, 1881–1886, 1889, 1901–1906, 1909, 1921–1926, 1929, 1947–1952, 1955, 1967–1972, 1975, 1998–2003, 2006
- `\csexpandonce` ... 1176, 1179, 1188, 1192, 1201, 1205, 1214, 1218, 1227, 1231, 1247, 1260, 1273, 1286, 1299, 2465, 2472, 2479, 4362
- `\csexpandonce{@tag}` ..... 669
- `\csnumdef` ..... 770, 772
- `\csnumgdef` ..... 2049
- `\csuse` 1335–1337, 1544, 1551–1553, 1585, 1587–1589, 1591, 1640–1642, 1683–1685, 2017, 2023, 2030, 2034, 2036, 2039, 2043, 2047
- `\ctab` ..... 3232, 3757, 3848
- `\ctabtext` ..... 3236, 3767, 3852
- D**
- `\D@@footnote` ... 3332, 3413, 3430, 3449
- `\date` ..... 3880, 4705
- `\dcolerr` ..... 3288, 3300
- `\dcoli` ... 3258, 3290, 3291, 3658, 3663
- `\dcolii` ..... 3259, 3291
- `\dcoliii` ..... 3260, 3291
- `\dcoliv` ..... 3261, 3292
- `\dcolix` ..... 3266, 3293
- `\dcolv` ..... 3262, 3292
- `\dcolvi` ..... 3263, 3292
- `\dcolvii` ..... 3264, 3293
- `\dcolviii` ..... 3265, 3293
- `\dcolx` ..... 3267, 3293
- `\dcolxi` ..... 3268, 3294
- `\dcolxii` ..... 3269, 3294
- `\dcolxiii` ..... 3270, 3294
- `\dcolxiv` ..... 3271, 3295
- `\dcolxix` ..... 3276, 3296
- `\dcolxv` ..... 3272, 3295
- `\dcolxvi` ..... 3273, 3295
- `\dcolxvii` ..... 3274, 3296
- `\dcolxviii` ..... 3275, 3296
- `\dcolxx` ..... 3277, 3296
- `\dcolxxi` ..... 3278, 3297
- `\dcolxxii` ..... 3279, 3297
- `\dcolxxiii` ..... 3280, 3297
- `\dcolxxiv` ..... 3281, 3298
- `\dcolxxix` ..... 3286, 3299
- `\dcolxxv` ..... 3282, 3298
- `\dcolxxvi` ..... 3283, 3298
- `\dcolxxvii` ..... 3284, 3299
- `\dcolxxviii` ..... 3285, 3299
- `\dcolxxx` ..... 3287, 3299
- `\DeclareOption` ..... 6, 7
- Dekker, Dirk-Jan ..... 3, 32, 201
- `\Dend` ..... 2343, 2351
- `\Dendnote` ..... 12, 2330
- `\Dfootgroup` ..... 2114
- `\Dfootins` 1459, 2114, 2125, 2165, 2178
- `\Dfootnote` 11, 1183, 2941, 3335, 3405, 3413, 3422, 3430, 3449, 3998, 4033
- `\Dfootstart` ..... 2114
- `\dimen` 603, 604, 606–608, 610, 1469, 1474, 1498–1500, 1503, 1604–1606, 1621, 1624, 1669, 1695, 2688, 2693, 2694, 2709, 2712, 2755, 2758, 2803–2805, 2808, 4102
- `\dimen@` ..... 2069, 2071
- `\disablel@dtabfeet` ..... 3464, 3481, 3495, 3509, 3522, 3537, 3665, 3672, 3677, 3685, 3690, 3698, 3716, 3732, 3855
- `\displaystyle` ..... 3351, 3466, 3468, 3497, 3499, 3524, 3526, 3665, 3677, 3690, 3778, 3830, 3831
- `\displaywidowpenalty` ..... 789
- `\divide` ..... 952, 965, 1500, 1605, 2805, 4396, 4422

- \do@actions ..... 853, 880
  - \do@actions@fixedcode ..... 900, 913
  - \do@actions@next ..... 880
  - \do@ballast ..... 854, 868
  - \do@line ..... 773, 814
  - \do@linehook ..... 818, 835
  - \do@lockoff ..... 527
  - \do@lockoffL ..... 527
  - \do@lockon ..... 498
  - \do@lockonL ..... 498
  - \docsvlist ..... 1169
  - \documentclass ..... 3866,  
3922, 4062, 4186, 4340, 4605, 4793
  - \doedindexlabel 3039, 3059, 3084, 3325
  - \doendnotes ..... 21, 2397
  - \doreinextrafeeti ... 2118, 2904, 2925
  - \doreinextrafeetii .. 2119, 2121, 2148
  - \dosplits ..... 1596
  - \dots 3230, 4000, 4035, 4158, 4161, 4318
  - \doublecolon ..... 3949, 3958
  - Downes, Michael ..... 31, 97, 99
  - \doxtrafeet ..... 2102
  - \doxtrafeeti ..... 2103, 2904, 2920
  - \doxtrafeetii ..... 2104, 2106
  - \dp ..... 1316, 1517,  
1531, 2069, 2089, 2826, 2839, 4644
  - \dummy@edtext ..... 639, 647
  - \dummy@ref ..... 554, 564
  - \dummy@text ..... 638, 646
- E**
- \E@footnote ... 3332, 3414, 3431, 3450
  - \edaftertab .....  
.. 28, 169, 3238, 3705, 3739, 3757
  - edarrayc (environment) ..... 26, 3847
  - edarrayl (environment) ..... 26, 3847
  - edarrayr (environment) ..... 26, 3847
  - \EDATAB ..... 3797, 3805
  - \edatab ..... 3239, 3803
  - \edatabell ..... 3240, 3787
  - \edatleft ..... 28, 3241, 3589
  - \edatright ..... 28, 3242, 3597
  - \edbforetab .....  
.. 28, 169, 3237, 3705, 3739, 3757
  - \edfilldimen .....  
.... 3611, 3621, 3622, 3624, 3648
  - \edfont@info ..... 704, 707, 711
  - \EDINDEX ..... 3318
  - \edindex .... 25, 3046, 3082, 3318,  
3407, 3415, 3424, 3432, 3445,  
3457, 3474, 3490, 3504, 3517, 3532
  - \edindexlab ..... 25, 3034, 3040, 3043
  - \EDLABEL ..... 3316
  - \edlabel ..... 21, 644, 2227, 3040,  
3316, 3435, 3454, 3456, 3473,  
3489, 3503, 3516, 3531, 3664,  
3671, 3676, 3684, 3689, 3697, 3903
  - \edmakelabel ..... 22, 2317
  - \edpageref ..... 21, 2278
  - \edrowfill . 27, 3245, 3436, 3439, 3646
  - \EDTAB ..... 3793, 3829
  - \edtabcolsep ..... 27, 3380,  
3469, 3486, 3499, 3513, 3527,  
3542, 3622, 3679, 3692, 3701, 3815
  - \EDTABINDENT ..... 3810, 3823
  - \edtabindent ..... 3653,  
3657, 3662, 3673, 3686, 3699, 3819
  - \EDTABtext ..... 3837
  - edtabularc (environment) .... 26, 3851
  - edtabularl (environment) .... 26, 3851
  - edtabularr (environment) .... 26, 3851
  - \EDTEXT ..... 3312
  - \edtext ..... 11, 647, 683,  
2455–2457, 2567, 2672, 3312,  
3418, 3444, 3895, 3896, 3902,  
3987, 3990, 3992, 3994, 3996,  
3997, 4002, 4004, 4006, 4008,  
4010, 4012, 4014, 4016, 4018,  
4023, 4025, 4027, 4029, 4031,  
4032, 4037, 4039, 4041, 4043,  
4045, 4047, 4049, 4051, 4053,  
4128, 4154, 4156–4158, 4171,  
4256, 4259, 4267, 4275, 4281,  
4287, 4293, 4294, 4296, 4301,  
4304, 4307, 4313, 4315, 4321,  
4330, 4505, 4506, 4516, 4523,  
4526, 4529, 4530, 4540, 4541,  
4545, 4547, 4551, 4553, 4555,  
4559, 4561, 4562, 4565, 4568,  
4570, 4576, 4589, 4718, 4720,  
4722, 4724, 4726–4728, 4730,  
4731, 4734, 4743, 4745, 4748,  
4751–4753, 4755–4759, 4761,  
4762, 4765, 4768, 4770, 4772–  
4777, 4855, 4860, 4902, 4927,  
4928, 4930, 4932, 4934, 4936,  
4937, 4939–4941, 4945, 4947,

- 4949, 4950, 4953, 4954, 4958–  
4962, 4964, 4966, 4969, 4972–4974  
`\edvertdots` ..... 29, 3244, 3608  
`\edvertline` ..... 29, 3243, 3606  
`\Eend` ..... 2347, 2351  
`\Eendnote` ..... 12, 2330  
`\Efootgroup` ..... 2115  
`\Efootins` 1460, 2115, 2126, 2166, 2179  
`\Efootnote` ..... 11, 1183,  
2942, 3336, 3406, 3414, 3423,  
3431, 3450, 4003, 4005, 4007,  
4009, 4011, 4013, 4015, 4017, 4019  
`\Efootstart` ..... 2115  
`\emergencystretch` ..... 4507, 4701  
`\empty` 19, 106, 147, 150, 307, 308, 345,  
677, 692, 702, 716, 720, 726,  
757, 903, 961, 977, 1096–1098,  
1109, 1141, 2229, 3162, 4096, 4869  
`\enablel@dtabfeet` ..... 3713,  
3729, 3747, 3755, 3765, 3773, 3855  
`\end@lemmas` .....  
637, 677, 678, 692, 693, 4869, 4870  
`\endashchar` ..... 16, 1340, 1412, 2389  
`\endgraf` ..... 768, 808, 812, 4089  
`\endline@num` ..... 333, 571, 577  
`\endlock` 10, 627, 643, 3204, 3208, 3210  
`\endminipage` ..... 2976  
`\endnumbering` .....  
.. 6, 118, 140, 162, 172, 3905,  
4057, 4179, 4333, 4593, 4782, 4976  
`\endpage@num` ..... 333, 570, 577  
`\endprint` ..... 21, 2351, 2400  
`\endstanzaextra` ..... 19, 3197, 4836  
`\endsub` ..... 9, 603, 642, 4110  
`\endsubline@num` ..... 333, 572, 578  
`\enskip` 2352, 3933, 3945, 3958, 3965,  
4094, 4097, 4217, 4223, 4487,  
4495, 4667, 4680, 4687, 4809, 4814  
`\enspace` ..... 2352, 2636,  
2729, 2774, 2847, 3932, 3957,  
3964, 4194, 4222, 4231, 4486,  
4494, 4666, 4676, 4686, 4808, 4813  
environments:  
`edarrayc` ..... 26, 3847  
`edarrayl` ..... 26, 3847  
`edarrayr` ..... 26, 3847  
`edtabularc` ..... 26, 3851  
`edtabularl` ..... 26, 3851  
`edtabularr` ..... 26, 3851  
`ledgroup` ..... 20, 2994  
`ledgroupsize` ..... 20, 3007  
`minipage` ..... 20  
Euclid ..... 4  
`\ExecuteOptions` ..... 8  
`\exit` ..... 4196, 4324  
`\extensionchars` .... 29, 104, 124, 168
- F**
- `\f@encoding` ..... 711  
`\f@family` ..... 711  
`\f@series` ..... 711  
`\f@shape` ..... 711  
`\fx@l@cks` ..... 1002, 1007  
Fairbairns, Robin ..... 24  
`\ffootnote` ..... 4359, 4505  
`\finishstage` ..... 4109, 4110  
`\first@linenum@out@false` ... 582, 588  
`\first@linenum@out@true` ..... 582  
`\firstlinenum` ..... 7, 9, 241  
`\firstsublinenum` ..... 7, 9, 241  
`\fix@page` ..... 369, 422  
`\flag@end` ..... 596, 682, 697, 4874  
`\flag@start` ..... 596, 674, 689, 4866  
`\flagstanza` ..... 20, 3213, 4846  
`\floatingpenalty` ..... 1315  
`\flush@notes` ..... 778, 1139  
Folkerts, Menso ..... 4  
`\fontencoding` ..... 1155  
`\fontfamily` ..... 1155  
`\fontseries` ..... 1155  
`\fontshape` ..... 1155  
`\footfootmarkA` ..... 24, 2591  
`\footfudgefiddle` 31, 1496, 1500, 2805  
`\footgroupA` ..... 2907  
`\footgroupB` ..... 2908  
`\footgroupC` ..... 2909  
`\footins` .....  
1425, 2084, 2091, 2095, 2130, 2174  
`\footinsA` 2167, 2180, 2587, 2907, 2913  
`\footinsB` 2168, 2181, 2875, 2908, 2914  
`\footinsC` 2169, 2182, 2889, 2909, 2915  
`\footnormal` .....  
1456, 1478–1482, 2144, 3979, 3980  
`\footnormalX` 24, 2677, 2900–2902, 2919  
`\footnote` ..... 3900, 3910, 3989  
`\footnoteA` ..... 24, 2581, 2951  
`\footnoteB` . 24, 2866, 2952, 3911–3913  
`\footnoteC` ..... 24, 2884, 2953  
`\footnoteoptions@` .....  
..... 1158, 1173, 1181, 1185,

1190, 1194, 1198, 1203, 1207,  
1211, 1216, 1220, 1224, 1229,  
1233, 1244, 1249, 1253, 1257,  
1262, 1266, 1270, 1275, 1279,  
1283, 1288, 1292, 1296, 1301, 1305  
`\footnoterule` .. 1447, 2094, 2648, 2991  
`\footnotesize` ..... 1151, 2452,  
2453, 4476, 4477, 4613, 4614, 4799  
`\footparagraph` 13, 1484, 3977, 4100,  
4236, 4473, 4474, 4693–4695, 4802  
`\footparagraphX` ..... 24, 2789, 3873  
`\footsplitskips` ..... 1310,  
1313, 1513, 1527, 1629, 1672,  
2621, 2717, 2763, 2822, 2835, 4630  
`\footstartA` ..... 2907  
`\footstartB` ..... 2908  
`\footstartC` ..... 2909  
`\footthreecol` ..... 13, 1609, 3976  
`\footthreecolX` ..... 24, 2744  
`\foottwocol` ..... 13, 1657, 3975, 4237  
`\foottwocolX` ..... 24, 2698  
`\foottwocolX` ..... 2698  
`\fullstop` .....  
16, 297, 1340, 1409, 1411, 1413,  
1415, 2388, 2392, 4222, 4231, 4879

## G

`\g@addto@macro` ..... 2136, 2137,  
2145, 2148, 2151, 2153, 2161,  
2920, 2925, 2928, 2931, 3046, 3082  
Gädeke, Nora ..... 5  
`\get@linelistfile` ..... 341, 357  
`\getline@num` ..... 822, 849  
`\gfootnote` ..... 4369, 4506  
`\ggfootfmt` ..... 4087, 4101  
`\ggfootstart` ..... 4104, 4105  
`\gl@p` 319, 348, 349, 678, 693, 706, 906,  
907, 1102, 1106, 1142, 2232, 4870  
`\gl@poff` ..... 319, 320  
`\goodbreak` ..... 4233

## H

`\hangafter` .... 3193, 3942, 4108, 4193  
`\hangingsymbol` ..... 20, 3136, 3143  
`\hb@xt@` ..... 830,  
832, 842, 844, 3648, 3653, 3657,  
3662, 3673, 3686, 3699, 3776, 3778  
`\hfilneg` ..... 1598  
`\Hilfsbox` ..... 3253

`\hilfsbox` ..... 3253, 3308, 3309,  
3351, 3363, 3453, 3466, 3483,  
3497, 3511, 3524, 3539, 3664,  
3666, 3671, 3675, 3676, 3678,  
3684, 3688, 3689, 3691, 3697, 3700  
`\hilfscount` .... 3253, 3814–3816, 3822  
`\HILFSskip` ..... 3807  
`\Hilfsskip` ..... 3654,  
3658, 3659, 3663, 3666, 3667,  
3674, 3675, 3678–3680, 3687,  
3688, 3691–3693, 3700–3702,  
3807, 3813, 3815, 3821, 3825, 3826  
`\hilfsskip` .....  
. 3253, 3452, 3453, 3468, 3485,  
3499, 3513, 3526, 3541, 3824–3826  
`\hphantom` ..... 3228  
`\Hy@temp@A` ..... 3070, 3071  
`\HyInd@ParenLeft` ..... 3071  
`\hyphenation` ..... 4706  
`\hyphenpenalty` ..... 4070

## I

`\if@colmade` ..... 2200, 2209  
`\if@firstcolumn` 983, 1058, 2202, 2514  
`\if@led@nofoot` ..... 2157, 2187  
`\if@nobreak` ..... 742  
`\ifautopar` ..... 759, 792  
`\ifbypage@` 176, 416, 427, 885, 1372, 4077  
`\ifbypstart@` ..... 176, 774, 1397  
`\ifcsdef` ..... 1586, 2028  
`\ifcseempty` 1335, 1551, 1588, 1640, 1683  
`\ifcsstring` ..... 2032  
`\ifdim` 604, 606, 608, 610, 2539, 3308, 3814  
`\iffirst@linenum@out@` ..... 582, 586  
`\ifhbox` ..... 1577, 1582  
`\ifhmode` ..... 2553, 2560, 2602, 2609  
`\ifinserterhangingsymbol` ..... 3139  
`\ifinstanza` ..... 759, 809, 3136, 3142  
`\ifl@d@dash` ..... 1344, 1412, 2389  
`\ifl@d@elin` ..... 1344,  
1390, 1414, 1415, 2379, 2391, 2392  
`\ifl@d@esl` ..... 1344, 1415, 2392  
`\ifl@d@pnnum` .....  
1344, 1378, 1409, 1413, 2367, 2390  
`\ifl@d@ssub` ..... 1344, 1411, 2388  
`\ifl@dend@` ..... 2319, 2325  
`\ifl@dmemoir` ..... 22, 3319  
`\ifl@dpairing` ..... 108, 144, 1398  
`\ifl@dskipnumber` ..... 630, 944  
`\ifl@dstartendok` ..... 3618, 3628

- \ifledfinal ..... 5, 13, 29
  - \ifledplinenumber ..... 1723, 4675
  - \ifledRcol ..... 108, 795, 1399
  - \ifleftnoteup ..... 2491, 2505
  - \ifnolinenums ..... 4073, 4092
  - \ifnoteschanged@ ..... 154, 337
  - \ifnumberedpar@ ..... 731, 752, 764, 1174, 1186, 1199, 1212, 1225, 1245, 1258, 1271, 1284, 1297, 2332, 2333, 2336, 2337, 2340, 2341, 2344, 2345, 2348, 2349, 2407, 2464, 2471, 2478, 2566, 2572, 2660, 2671, 4361, 4371
  - \ifnumbering ..... 107, 116, 141, 164, 179, 748, 761, 801
  - \ifnumberingR ..... 108, 796
  - \ifnumberline ..... 699, 850, 855, 943
  - \ifnumberpstart ... 739, 759, 781, 809
  - \ifnumequal .... 775, 1585, 1587, 2030
  - \ifodd ..... 993, 1068, 2524, 4449
  - \ifpst@rtedL ..... 108
  - \ifpstartnum ..... 1079, 1082, 1087
  - \ifreportnoidxfile ..... 3052
  - \ifrightnoteup ..... 2460, 2500
  - \ifshowindexmark ..... 3065, 3098
  - \ifsidepstartnum ..... 759, 1053
  - \ifstrempy ..... 1712, 1758
  - \ifstrequal ..... 1160, 1733, 1778, 1798, 1818, 1838, 1858, 1879, 1899, 1919, 1945, 1965, 1996
  - \ifsublines@ ..... 295, 324, 405, 440, 445, 451, 466, 484, 493, 507, 529, 576, 578, 856, 892, 947, 2251, 2276, 4391
  - \iftoggle ..... 1335, 1551, 1640, 1683, 2016, 2021, 2026
  - \ifvbox ..... 771, 2064, 2132
  - \ifvmode ..... 2233
  - \ifvoid ..... 2084, 2111–2115, 2122–2126, 2130, 2146, 2149, 2154, 2162–2169, 2174–2182, 2907–2909, 2913–2915, 2923, 2926, 2932, 2944–2948, 2955–2957, 2979, 3001, 3028
  - \IMM ..... 4839, 4840
  - \indexentry ..... 3091
  - \initnumbering@reg ..... 115
  - \inplaceoflemmaseparator ... 15, 1944
  - \inplaceoflemmaseparator@A .... 1938
  - \inplaceoflemmaseparator@B .... 1939
  - \inplaceoflemmaseparator@C .... 1940
  - \inplaceoflemmaseparator@D .... 1941
  - \inplaceoflemmaseparator@E .... 1942
  - \inplaceoflemmaseparator@Z .... 1943
  - \inplaceofnumber ..... 14, 1851, 1935
  - \inplaceofnumber@A ..... 1851
  - \inplaceofnumber@B ..... 1852
  - \inplaceofnumber@C ..... 1853
  - \inplaceofnumber@D ..... 1854
  - \inplaceofnumber@E ..... 1855
  - \inplaceofnumber@Z ..... 1856
  - \InputIfFileExists ..... 358
  - \insert ..... 1308, 1510, 1627, 1670, 2122–2126, 2130, 2132, 2150, 2619, 2715, 2761, 2819, 2913–2915, 2927, 4627
  - \insert@count ..... 552, 553, 598, 672, 687, 1162, 1177, 1189, 1202, 1215, 1228, 1248, 1261, 1274, 1287, 1300, 2467, 2474, 2481, 2575, 2663, 4364, 4374, 4864
  - \insert@countR ..... 1166
  - \inserthangingsymbol ..... 830, 3140
  - \inserthangingsymbolfalse ..... 826
  - \inserthangingsymboltrue ..... 824
  - \inserthangingymbol ..... 3139
  - \insertlines@list ..... 147, 328, 354, 560, 1098, 1102
  - \insertparafootsep .. 1547, 1584, 2845
  - \inserts@list .... 756, 1093, 1096, 1106, 1141, 1142, 1161, 1176, 1188, 1201, 1214, 1227, 1247, 1260, 1273, 1286, 1299, 2466, 2473, 2480, 2574, 2662, 4363, 4373
  - \inserts@listR ..... 1165
  - \instanzafalse ..... 3138, 3210
  - \instanzatrue ..... 3199
  - \interfootnotelinepenalty ..... 1314
  - \interlinepenalty 789, 1126, 1314, 3204
  - \interparanoteglue ..... 14, 1980, 1985, 1987, 4204
  - \ipn@skip ..... 1981, 1984, 1986
  - \itshape ..... 4109, 4218, 4224, 4232
- J**
- Jayaditya ..... 5
- K**
- Kabelschacht, Alois ..... 84
  - Krukov, Alexej ..... 61

## L

- \l@d@@wrindexhyp ..... 3063, 3096
- \l@d@add ..... 721, 723, 727, 729
- \l@d@dashfalse ..... 1353, 1371, 2362
- \l@d@dashtrue .....
  - 1375, 1381, 1393, 2365, 2370, 2382
- \l@d@elinfalse ..... 1349, 1378, 2367
- \l@d@elintrue .. 1378, 1380, 2367, 2369
- \l@d@end .. 2319, 2321, 2322, 2328,
  - 2331, 2335, 2339, 2343, 2347, 2404
- \l@d@err@UnequalColumns ..... 3394
- \l@d@eslfalse .....
  - .... 1351, 1387, 1390, 2376, 2379
- \l@d@esltrue ... 1390, 1392, 2379, 2381
- \l@d@index ..... 3048, 3050, 3321
- \l@d@makecol ..... 2076, 2139, 2218
- \l@d@nums 671, 704, 707, 715, 716, 729,
  - 1176, 1188, 1201, 1214, 1227,
  - 1247, 1260, 1273, 1286, 1299,
  - 2332, 2336, 2340, 2344, 2348, 4362
- \l@d@pnumfalse ..... 1345, 1371, 2362
- \l@d@pnumtrue ..... 1374, 2364
- \l@d@reinserts ..... 2129, 2140
- \l@d@section ..... 2328, 2351
- \l@d@set ..... 473, 624
- \l@d@ssubfalse ..... 1347, 1383, 2372
- \l@d@ssubtrue ..... 1385, 2374
- \l@d@wrindexm@m ..... 3062, 3063
- \l@d@ampcount ..... 3248,
  - 3390, 3392, 3397, 3662, 3672,
  - 3673, 3685, 3686, 3721, 3737, 3775
- \l@d@begin@stack .... 3114, 3124–3126
- \l@d@bfnote ..... 2567, 2571
- \l@d@checkcols ..... 3347, 3359, 3387
- \l@d@checkstartend ..... 3617, 3628
- \l@d@chset@num ..... 372, 375, 473
- \l@d@colcount ..... 3248, 3290,
  - 3302, 3303, 3346, 3348, 3358,
  - 3360, 3388, 3392, 3397, 3458,
  - 3460, 3475, 3477, 3491, 3492,
  - 3505, 3506, 3518, 3519, 3533,
  - 3534, 3584, 3585, 3662, 3672,
  - 3673, 3685, 3686, 3717, 3719,
  - 3733, 3735, 3775, 3781, 3811, 3820
- \l@d@collect@tbody ..... 3116, 3123
- \l@d@collect@body .....
  - .... 3111, 3847–3849, 3851–3853
- \l@d@colwidth .... 3290, 3308, 3309,
  - 3452, 3583, 3648, 3674, 3687,
  - 3776, 3778, 3783, 3792, 3813, 3814
- \l@dcsnote ..... 2457, 2460
- \l@dcsnotetext ..... 836, 2486,
  - 2507, 2512, 2515, 2517, 2525, 2527
- \l@d@dodoreinextrafeet 2117, 2131, 2137
- \l@d@dofootinsert ..... 2077, 2082
- \l@d@dextrafeet ..... 2099, 2102, 2136
- \l@d@dedbeginmini .... 2151, 2934, 2937
- \l@d@dedendmini ..... 2153, 2935, 2937
- \l@d@emptyd@ta ..... 819, 836
- \l@d@dend@close ..... 2321, 2397
- \l@d@dend@false ..... 2319, 2322
- \l@d@dend@open ..... 2321, 2326
- \l@d@dend@stuff ... 125, 169, 2324, 2403
- \l@d@dend@true ..... 2319, 2321
- \l@d@envbody .... 3106, 3109, 3112–3114
- \l@d@fambeginmini ... 2928, 2934, 2950
- \l@d@famendmini ..... 2931, 2935, 2950
- \l@d@feetbeginmini .....
  - ..... 2934, 2971, 2997, 3024
- \l@d@feetendmini 2934, 2982, 3004, 3031
- \l@d@getline@margin ..... 202
- \l@d@getlock@disp ..... 246, 274
- \l@d@getref@num ..... 2278,
  - 2279, 2281, 2282, 2284, 2285, 2292
- \l@d@getsidenote@margin ..... 2414
- \l@d@gobblearg ..... 3338, 3419–3423
- \l@d@gobbledarg ..... 3338, 3402–3406
- \l@d@label@parse ..... 2298, 2301
- \l@d@dld@ta ... 830, 836, 982, 1059, 1071
- \l@d@dlp@rbox .... 842, 2445, 2490, 2496
- \l@d@dlsn@te ..... 831, 841
- \l@d@dlsnote ..... 2455, 2460
- \l@d@make@labels 2237, 2240, 2262, 2271
- \l@d@memoirfalse ..... 23
- \l@d@memoirtrue ..... 23
- \l@d@modforcritext ..... 3400, 3856
- \l@d@modforedtext ..... 3417, 3859
- \l@d@nullfills ..... 3434,
  - 3706, 3724, 3740, 3750, 3758, 3768
- \l@d@numpstartsL ..... 108, 130
- \l@d@dold@footnotetext .... 2547, 2549
- \l@d@dold@xympar ..... 2405
- \l@d@dold@dold@footnotetext . 2564, 2579
- \l@d@p@rsefootsspec ..... 1354
- \l@d@pairingfalse ..... 108
- \l@d@pairingtrue ..... 108
- \l@d@parsedendline ..... 1354, 4637
- \l@d@parsedendpage ..... 1354
- \l@d@parsedendsub ..... 1354

- \l@dparsestartline ..... 1354, 4632, 4637, 4639
- \l@dparsestartpage . 1354, 4635, 4640
- \l@dparsestartsub ..... 1354
- \l@parsefootspec ..... 1354, 4631
- \l@push@begins ..... 3120, 3124
- \l@drd@eta ... 832, 836, 982, 1061, 1069
- \l@dref@undefined ..... 2278, 2281, 2284, 2287
- \l@drestorefills ..... 3434, 3710, 3726, 3744, 3752, 3762, 3770
- \l@drestoreforcritext ... 3400, 3857
- \l@drestoreforedtext .... 3417, 3860
- \l@drp@rbox ..... 844, 2445, 2499
- \l@drsn@te ..... 833, 841
- \l@drsnote ..... 2456, 2460
- \l@dsetmaxcolwidth .. 3307, 3353, 3365
- \l@dskipnumberfalse ..... 630, 945
- \l@dskipnumbertrue ..... 630, 936
- \l@dstartendokfalse . 3632, 3636, 3640
- \l@dstartendoktrue ..... 3630
- \l@dtabaddcols ..... 3616, 3647
- \l@dtabnoexpands ..... 648, 3221
- \l@dunboxmpfoot 2980, 2988, 3002, 3029
- \l@dunhbox@line ..... 814
- \l@dzeropenalties ..... 767, 787
- Lück, Uwe ..... 3
- \label ..... 22, 1418, 1432, 3888
- \label@refs ..... 2230, 2232, 2237, 2240, 2248, 2249, 2253, 2255, 2257
- \labelref@list . 2223, 2229, 2232, 2276
- \labelrefsparseline ..... 2243
- \labelrefsparsesubline ..... 2243
- \last@page@num ..... 422
- \lastbox ... 807, 821, 1539, 1576, 1581
- \lastkern ..... 2539
- \lastskip ..... 603, 607
- Lavagnino, John ..... 2, 4
- \lbreak ..... 4840, 4903
- \ldots ..... 4740, 4742, 4768
- Leal, Jeronimo@Leal, Jerónimo ..... 3
- \led@err@AutoparNotNumbered .... 66, 797, 802
- \led@err@HighEndColumn .... 93, 3637
- \led@err@LineationInNumbered 43, 180
- \led@err@LowStartColumn ... 93, 3633
- \led@err@NumberingNotStarted 27, 158
- \led@err@NumberingShouldHaveStarted ..... 27, 171
- \led@err@NumberingStarted ... 27, 117
- \led@err@PendNoPstart ..... 66, 765
- \led@err@PendNotNumbered .... 66, 762
- \led@err@PstartInPstart .... 66, 753
- \led@err@PstartNotNumbered .. 66, 749
- \led@err@ReverseColumns ... 93, 3641
- \led@err@TooManyColumns ... 93, 3304
- \led@err@UnequalColumns ..... 93
- \led@mess@NotesChanged ..... 33, 155
- \led@mess@SectionContinued .. 41, 167
- \led@warn@BadAction ..... 80, 938
- \led@warn@BadAdvancelineLine 56, 460
- \led@warn@BadAdvancelineSubline ..... 56, 454
- \led@warn@BadLineation ..... 46, 197
- \led@warn@BadLinenummargin .. 46, 225
- \led@warn@BadLockdisp ..... 46, 252
- \led@warn@BadSetline ..... 62, 615
- \led@warn@BadSetlinenum .... 62, 622
- \led@warn@BadSidenotemargin 89, 2437
- \led@warn@BadSublockdisp .... 46, 278
- \led@warn@DuplicateLabel ... 82, 2265
- \led@warn@NoIndexFile ..... 91, 3053
- \led@warn@NoLineFile ..... 54, 363
- \led@warn@NoMarginpars .... 87, 2408
- \led@warn@RefUndefined .... 82, 2289
- \ledfinalfalse ..... 7
- \ledfinaltrue ..... 6
- \ledfootinsdim ..... 1456, 1469, 1474, 2688, 2694
- ledgroup (environment) ..... 20, 2994
- ledgroupsized (environment) . 20, 3007
- \ledleftnote ..... 23, 2455
- \ledlinenum ..... 290
- \ledllfill ..... 832, 846, 3011, 3015
- \ledlsnotefontsetup ... 24, 2448, 2489
- \ledlsnotesep ..... 24, 842, 2448
- \ledlsnotewidth ..... 23, 2448, 2489
- \Ledmac ..... 1438
- \ledmac@error ..... 26, 28, 30, 32, 44, 67, 70, 73, 76, 78, 94, 96, 99, 101, 103, 3642
- \ledmac@warning ..... 25, 47, 49, 51, 53, 55, 57, 60, 63, 65, 81, 83, 85, 88, 90, 92, 2248
- \ledplinenumfalse ..... 4633
- \ledplinenumtrue 1724, 4631, 4636, 4638
- \ledrightnote ..... 23, 2455, 3901
- \ledrlfill ..... 832, 846, 3012, 3019
- \ledrsnotefontsetup ... 24, 2448, 2498
- \ledrsnotesep ..... 24, 844, 2448



- `\ledrsnotewidth` . 23, 2448, 2498, 3877
  - `\ledsetnormalparstuff` . . . . . 1328, 1548, 2635, 2846, 3931, 3963, 4215, 4221, 4484, 4492, 4664, 4672, 4685, 4807, 4812
  - `\ledsidenote` . . . . . 23, 2455, 4719
  - `\left` . . . . . 3591, 3594, 3599, 3602
  - `\leftctab` . . . . . 3661, 3759
  - `\lefthyphenmin` . . . . . 4345
  - `\leftlinenum` 9, 290, 984, 996, 4082, 4453
  - `\leftltab` . . . . . 3652, 3741
  - `\leftnoteuptrue` . . . . . 2506
  - `\leftpstartnum` . . . . . 1053
  - `\lefttrtab` . . . . . 3656, 3707
  - `Leibniz` . . . . . 5
  - `\lemma` . . . . . 12, 713, 4000, 4035, 4158, 4161, 4257, 4268, 4293, 4318, 4517, 4523, 4526, 4541, 4768
  - `\lemmafnt` . . . . 4487, 4502, 4667, 4687
  - `\lemmaseparator` . . . . . 14, 1871, 1932
  - `\lemmaseparator@A` . . . . . 1871
  - `\lemmaseparator@B` . . . . . 1872
  - `\lemmaseparator@C` . . . . . 1873
  - `\lemmaseparator@D` . . . . . 1874
  - `\lemmaseparator@E` . . . . . 1875, 1876
  - `\lemmaseparator@Z` . . . . . 1877
  - `\letsforverteilen` . . . . . 3442, 3467, 3484, 3498, 3512, 3525, 3540
  - `Levy, Silvio` . . . . . 195
  - `\line` . . . . 2247, 2249, 2250, 2255, 2257
  - `\line@list` 150, 328, 353, 578, 702, 706
  - `\line@list@stuff` . . . . 124, 168, 584
  - `\line@margin` . . . . 202, 989, 1064, 4445
  - `\line@num` 132, 294, 322, 377, 411, 417, 428, 458, 459, 461, 469, 474, 475, 487, 571, 575, 862, 886, 895, 960, 962, 963, 972, 973, 1587, 2030, 2049, 2275, 4077–4079, 4082, 4209, 4418–4420, 4451
  - `\line@set` . . . . . 717, 718
  - `\lineation` . . . . . 8, 44, 47, 178, 3971, 4085, 4242, 4471, 4609
  - `\linenum` . . . . . 12, 714, 2314, 3337, 3408, 3425, 3451, 4854, 4865
  - `\linenum@out` . . . 581, 587, 589, 591, 592, 595, 597, 600, 605, 609, 612, 617, 624, 627, 628, 634, 2228
  - `\linenumberlist` . . . . 9, 19, 961, 973
  - `\linenumberstyle` . . . . 10, 281, 4918
  - `\linenumincrement` . . . . . 7, 9, 241
  - `\linenummargin` . . . . . 9, 49, 202, 3972, 4084, 4243, 4472, 4610
  - `\linenumr@p` . . . . . 281
  - `\linenumrep` . . . . . 281, 294, 1410, 1414, 2275, 2387, 2391, 4879
  - `\linenums` . . . . . 4075
  - `\linenumsep` . . . . . 9, 290, 1083, 1088, 2450, 2451, 4190, 4201, 4202
  - `\lineref` . . . . . 21, 2281, 3043, 3915
  - `\linewidth` . . . . . 830
  - `\list@clear` . . . . . 308, 353–356, 756
  - `\list@clearing@reg` . . . . . 340, 352
  - `\list@create` . . . . . 307, 328–331, 637, 1093, 2223
  - `\lmarpar` . . . . . 4505, 4524
  - `\lock@disp` . . . . . 246, 1025, 1029, 1033, 4430, 4434, 4438
  - `\lock@off` . . . . . 500, 501, 527, 628
  - `\lock@on` . . . . . 498, 627
  - `\lockdisp` . . . . . 10, 51, 246
  - `Lorch, Richard` . . . . . 5
  - `\lp@rbox` . . . . . 4357, 4382
  - `\ltab` . . . . . 3233, 3739, 3847
  - `\ltabtext` . . . . . 3235, 3749, 3851
  - `Luecking, Dan` . . . . . 37
- M**
- `\m@m@makecolfloats` . . . . . 2059, 2078
  - `\m@m@makecolintro` . . . . . 2059
  - `\m@m@makecoltext` . . . . . 2059, 2079
  - `\m@mdodoreinextrafeet` . . . . . 2137
  - `\m@mdoextrafeet` . . . . . 2136
  - `\m@mmf@check` . . . . . 2538, 2555, 2604
  - `\m@mmf@prepare` . . . . . 2535, 2550, 2559, 2585, 2597, 2608, 2870, 2881, 2888, 2897
  - `\m@th` . . . . . 3609
  - `\Ma` . . . . . 4347, 4547, 4574, 4583
  - `\makehboxofhboxes` . . . . . 1558, 1568, 1573, 2852, 2861
  - `\makeindex` . . . . . 3046, 3082
  - `\makememindexhook` . . . . . 3046
  - `\maketitle` . . . . . 3882, 4708
  - `\managestanza@modulo` . . . . 3168, 3185
  - `\marg` . . . . . 1936
  - `\marginparwidth` . . . . . 2448, 2449
  - `\mathchardef` . . . . . 3163, 4839
  - `\maxdepth` . . . . . 2080
  - `\maxdimen` 1514, 1528, 2823, 2836, 4641
  - `Mayer, Gyula` . . . . . 5

- \measurebody . . . 3709, 3715, 3743, 3761
  - \measuremcell . . . . . 3345, 3371
  - \measuremrow . . . . . 3369, 3720
  - \measuretbody . . . 3725, 3731, 3751, 3769
  - \measuretcell . . . . . 3357, 3376
  - \measuretrow . . . . . 3374, 3736
  - \medskip . . . . . 4520
  - \message . . . . . 42, 123
  - \mid . . . . . 4574, 4583
  - Middleton, Thomas . . . . . 5, 49
  - minipage (environment) . . . . . 20
  - Mittelbach, Frank . . . . . 4
  - \morenoexpands . . . 31, 640, 4353, 4659
  - \moveleft . . . . . 3654, 3659, 3667
  - \moveright . . . . . 3680, 3693, 3702
  - \mpAfootgroup . . . . . 2944
  - \mpAfootins . . . . . 1236, 2944
  - \mpAfootnote . . . . . 1242, 2938
  - \mpBfootgroup . . . . . 2945
  - \mpBfootins . . . . . 1236, 2945
  - \mpBfootnote . . . . . 1242, 2939
  - \mpCfootgroup . . . . . 2946
  - \mpCfootins . . . . . 1236, 2946
  - \mpCfootnote . . . . . 1242, 2940
  - \mpDfootgroup . . . . . 2947
  - \mpDfootins . . . . . 1236, 2947
  - \mpDfootnote . . . . . 1242, 2941
  - \mpEfootgroup . . . . . 2948
  - \mpEfootins . . . . . 1236, 2948
  - \mpEfootnote . . . . . 1242, 2942
  - \mpfootgroupA . . . . . 2955
  - \mpfootgroupB . . . . . 2956
  - \mpfootgroupC . . . . . 2957
  - \mpfootinsA . . . . . 2593, 2955
  - \mpfootinsB . . . . . 2877, 2956
  - \mpfootinsC . . . . . 2893, 2957
  - \mpfootnoteA . . . . . 2593, 2951
  - \mpfootnoteB . . . . . 2877, 2952
  - \mpfootnoteC . . . . . 2893, 2953
  - \mpnormalfootgroup . . . . . 1450, 1472
  - \mpnormalfootgroupX . . . . . 2653, 2691
  - \mpnormalvfootnote . . . . .
    - . . . . . 1319, 1471, 1614, 1662
    - . . . . . 2625, 2690, 2703, 2749
  - \mppara@footgroup . . . . . 1493, 1563
  - \mppara@footgroupX . . . . . 2798, 2850
  - \mppara@vfootnote . . . . . 1492, 1523
  - \mppara@vfootnoteX . . . . . 2797, 2818
  - \mpthreecolfootgroup . . . . . 1615, 1649
  - \mpthreecolfootgroupX . . . . . 2750, 2777
  - \mpthreecolfootsetup . . . . . 1616, 1622
  - \mpthreecolfootsetupX . . . . . 2751, 2753
  - \mptwocolfootgroup . . . . . 1663, 1693
  - \mptwocolfootgroupX . . . . . 2704, 2732
  - \mptwocolfootsetup . . . . . 1664, 1693
  - \mptwocolfootsetupX . . . . . 2705, 2707
  - \mpvAfootnote . . . . . 1246, 1251
  - \mpvBfootnote . . . . . 1259, 1264
  - \mpvCfootnote . . . . . 1272, 1277
  - \mpvDfootnote . . . . . 1285, 1290
  - \mpvEfootnote . . . . . 1298, 1303
  - \mpvfootnoteA . . . . . 2597
  - \mpvfootnoteB . . . . . 2881
  - \mpvfootnoteC . . . . . 2897
  - \multfootsep . . . . . 24, 2532, 2542
  - \multiplefootnotemarker . . . . .
    - . . . . . 2532, 2536, 2537, 2539
- N
- \n@num . . . . . 548, 634
  - \n@num@reg . . . . . 548
  - \NeedsTeXFormat . . . . . 2
  - \new@line . . . . . 595, 832
  - \newbox . . . . . 731, 734, 2445,
    - 2446, 3253, 3255, 3844, 3845, 4357
  - \newcommandx . . . . .
    - . . . . . 1158, 1331, 1537, 1546, 1631,
    - 1674, 1711, 1732, 1757, 1777,
    - 1797, 1817, 1837, 1857, 1878,
    - 1898, 1918, 1932, 1944, 1964, 1995
  - \newcounter . . . . . 232,
    - 234, 236, 238, 737, 869, 2588,
    - 2872, 2889, 3036, 3171, 3614, 4843
  - \newif . . . . . 5, 22, 107, 108, 110,
    - 113, 114, 176, 177, 324, 337,
    - 582, 630, 699, 732, 739, 792,
    - 1054, 1079, 1344, 1346, 1348,
    - 1350, 1352, 1723, 2157, 2320,
    - 2460, 2505, 3137, 3139, 3628, 4073
  - \newinsert . . . . . 1236–1240, 1458–1460,
    - 2587, 2598, 2875, 2882, 2891, 2898
  - \newlength . . . . . 290, 3155
  - \newlinechar . . . . .
    - . . . . . 2330, 2334, 2338, 2342, 2346
  - \newparafootfmt . . . . . 4220, 4238, 4271
  - \newread . . . . . 338
  - \newtoggle . . . . .
    - 1704–1709, 1751–1756, 1769, 1770
  - \newtwocolfootfmt . . . . . 4226, 4239

- \newwrite ..... 581, 2319
- \NEXT ..... 3341,
  - 3346, 3349, 3354, 3355, 3358,
  - 3361, 3366, 3367, 3370, 3372,
  - 3373, 3375, 3377, 3378, 3383,
  - 3546, 3549, 3551, 3552, 3554,
  - 3556, 3557, 3560, 3562, 3563,
  - 3565, 3567, 3568, 3571, 3573,
  - 3574, 3576, 3578, 3579, 3584,
  - 3586, 3587, 3781, 3784, 3785,
  - 3790, 3794, 3795, 3811, 3817, 3818
- \Next ..... 3383, 3459,
  - 3461, 3470, 3471, 3476, 3478,
  - 3487, 3488, 3491, 3493, 3500,
  - 3501, 3505, 3507, 3514, 3515,
  - 3518, 3520, 3528, 3529, 3533,
  - 3535, 3543, 3544, 3799, 3801, 3802
- \next@action ..... 81, 349,
  - 875, 883, 884, 889, 890, 898, 907
- \next@actionline .....
  - ..... 346, 348, 874, 882, 904, 906
- \next@insert .....
  - 757, 1097, 1100, 1102, 1105, 1109
- \next@page@num .....
  - ..... 137, 380, 382, 420, 432, 481
- \no@expands ... 640, 670, 685, 713, 4351
- \noalign ..... 1601
- \nobrak ..... 4657, 4658
- \noendnotes ..... 21,
  - 2403, 3875, 3925, 4065, 4199, 4343
- \noindent ..... 809, 1051, 1515,
  - 1529, 1561, 1571, 2824, 2837,
  - 2855, 2864, 4516, 4523, 4642, 4715
- \nolemmaseparator ..... 15, 1932
- \nolinenums ..... 4072, 4074
- \nolinenumsfalse ..... 4075
- \nolinenumstrue ..... 4074
- \nonum@ ..... 1769
- \nonumberinfootnote ... 14, 1748, 1757
- \nonumparafootfmt .. 4213, 4214, 4252
- \normal@footnotemarkX ... 2611, 2679
- \normal@pars .....
  - ... 143, 758, 812, 1329, 1632,
  - 1675, 2722, 2767, 3937, 3951, 4227
- \normalbfnoteX ..... 2659, 2672
- \normalbodyfootmarkX .... 2616, 2680
- \normalcolor 1452, 1565, 1651, 1698,
  - 2093, 2655, 2738, 2783, 2858, 2990
- \normalfont 292, 1152, 2533, 2617, 4233
- \normalfootfmt . 1328, 1464, 3930, 3978
- \normalfootfmtX ..... 2634, 2683
- \normalfootfootmarkX .... 2639, 2684
- \normalfootgroup ..... 1448, 1465
- \normalfootgroupX ..... 2650, 2685
- \normalfootnoterule .... 1447, 1467
- \normalfootnoteruleX 2648, 2686, 2795
- \normalfootstart ... 1417, 1443, 1462
- \normalfootstartX ..... 2642, 2678
- \normalvfootnote ..... 1307, 1463
- \normalvfootnoteX ..... 2618, 2681
- \nosep@ ..... 1770
- \nospeak ..... 4119, 4130
- \nospeaker ..... 4118
- \notblank ..... 1169
- \note 4241, 4282, 4297, 4302, 4308, 4319
- \notefontsetup ..... 16, 1151,
  - 1309, 1322, 1497, 1512, 1526,
  - 1560, 1570, 1628, 1645, 1671,
  - 1688, 1986, 2351, 2620, 2628,
  - 2716, 2732, 2762, 2777, 2802,
  - 2821, 2834, 2854, 2863, 4088, 4629
- \notenumfont ..... 16,
  - 1152, 2036, 2039, 2043, 2047,
  - 2351, 2636, 2729, 2774, 2847,
  - 3932, 3944, 3957, 3964, 3970,
  - 4222, 4231, 4476, 4486, 4494,
  - 4613, 4666, 4674, 4686, 4808, 4813
- \noteschanged@false ..... 337, 359
- \noteschanged@true .....
  - .... 148, 151, 337, 364, 703, 1099
- \notetextfont 4477, 4488, 4496, 4614,
  - 4668, 4681, 4688, 4799, 4810, 4815
- \nulledindex .... 3318, 3407, 3424,
  - 3457, 3474, 3490, 3504, 3517, 3532
- \nullsetzen ..... 3581, 3718, 3734
- \num@lines . 731, 768, 1116, 1122, 1125
- \numberedpar@false ..... 731
- \numberedpar@true ..... 731, 760
- \numberingfalse ..... 107, 142
- \numberingtrue ..... 107, 120, 162
- \numberit ..... 4845, 4850
- \numberlinefalse ..... 8
- \numberlinetrue ..... 8, 700
- \numberonlyfirstinline .... 14, 1704
- \numberpstartfalse ..... 8, 736
- \numberpstarttrue ..... 8, 736
- \numlabfont .. 16, 290, 4079, 4082, 4209

## O

- \oarg ..... 1749, 1936

- \oldprintlines ..... 4881
  - \one@line ..... 731, 820, 821, 832
  - \openout ..... 589, 592, 2321
  - \os ..... 4068, 4093, 4162, 4172
  - \overfullrule ..... 4344
- P**
- \PackageError ..... 26
  - \PackageWarning ..... 25
  - \page@action ..... 381, 479, 565
  - \page@num ..... 333,  
344, 419, 430, 570, 575, 884,  
991, 1066, 1585, 1594, 2522, 4447
  - \page@start ..... 601, 2083
  - \pagelinesep ..... 25, 3034, 3043
  - \pageno ..... 83, 85, 2055
  - \pageparbreak ..... 31, 1051
  - \pageref ..... 22
  - \pagestyle ..... 4249, 4886
  - \par@line .....  
. 731, 769, 1117, 1118, 1121, 1125
  - \para@footgroup ..... 1489, 1556
  - \para@footgroupX ..... 2794, 2850
  - \para@footsetup ..... 1491, 1497
  - \para@footsetupX ..... 2800, 2802
  - \para@vfootnote .... 1487, 1509, 4626
  - \para@vfootnoteX ..... 2792, 2818
  - \parafootfmt ..... 1488, 1546, 3962
  - \parafootfmtX ..... 2793, 2844
  - \parafootftmsep .... 1988, 1989–1994
  - \parafootsep ..... 15, 1989
  - \parafootsep@A ..... 1989
  - \parafootsep@B ..... 1990
  - \parafootsep@C ..... 1991
  - \parafootsep@D ..... 1992
  - \parafootsep@E ..... 1993
  - \parafootsep@Z ..... 1994
  - \parafootstart ..... 1486, 1505
  - \parafootstartX ..... 2791, 2810
  - \pausenumbering ..... 8, 161
  - \pend . 6, 73, 76, 754, 761, 810, 1051,  
3208, 3210, 3904, 4020, 4055,  
4110, 4118, 4137, 4150, 4178,  
4262, 4269, 4290, 4325, 4332,  
4519, 4578, 4592, 4892, 4904, 4909
  - \phantom ..... 3227
  - \phantomsection ..... 1418, 1432
  - Pigman, III<sup>rd</sup>, G. W. .... 189
  - Plato of Tivoli ..... 4
  - \postbodyfootmark ..... 2600, 2614
  - \postdisplaypenalty ..... 790
  - \prebodyfootmark ..... 2600, 2612
  - \predisplaypenalty ..... 789
  - \prevgraf ..... 768
  - \previous@A@number ..... 4651
  - \previous@B@number ..... 4652
  - \previous@C@number ..... 4653
  - \previous@page ..... 4635, 4640, 4654
  - \prevline ..... 1586, 2029
  - \prevpage@num ..... 1584
  - \printendlines ..... 2351, 2385
  - \printlinefootnote .....  
1333, 1549, 1638, 1681, 2012, 2015
  - \printlines 1396, 2039, 2043, 2047,  
3932, 3944, 3957, 3964, 4093,  
4222, 4231, 4486, 4494, 4666,  
4674, 4686, 4808, 4813, 4881, 4916
  - \printnpnum ..... 21, 2387, 2390, 2395
  - \printstanzalines ..... 4878, 4916
  - \processl@denvbody .....  
..... 3113, 3117, 3118, 3133
  - \ProcessOptions ..... 9
  - \protected@write .....  
2239, 3067, 3072, 3075, 3090, 3099
  - \protected@xdef .....  
2583, 2595, 2868, 2879, 2886, 2895
  - \ProvidesPackage ..... 3
  - \pst@rtedLfalse ..... 108, 129, 145
  - \pst@rtedLtrue ..... 108, 165
  - \pstart ..... 6, 67, 70, 71,  
76, 736, 809, 1051, 3189, 3891,  
3986, 4022, 4107, 4112, 4116,  
4118, 4119, 4193, 4255, 4265,  
4514, 4522, 4580, 4892, 4900, 4906
  - \pstartline ..... 772, 775
  - \pstartnum ..... 1053
  - \pstartnumfalse ..... 1084, 1091
  - \pstartnumtrue ..... 782, 1080
- Q**
- \quad ..... 4305, 4322
- R**
- \raggedright ..... 1636,  
1679, 2453, 2727, 2772, 3941, 3955
  - \raw@text ..... 731, 759, 771, 820
  - \rbracket ..... 16, 1340,  
1871–1877, 3945, 4217, 4223,  
4232, 4495, 4657, 4680, 4809, 4814
  - \rd@ta ..... 4386, 4451

- `\read@linelist` ..... 338, 585  
`\ref` ..... 22, 3916  
`\refstepcounter` ..... 4849  
`\Relax` ..... 3341, 3798, 3805  
`\rem@inder` ..... 973, 975–977  
`\removeboxes` .....  
     .... 1559, 1569, 1573, 2853, 2862  
`\removeatstart` ..... 3458, 3475  
`\RequirePackage` ..... 11, 12  
`\resumenumbering` ..... 8, 161  
`\right` ..... 3592, 3595, 3600, 3603  
`\rightctab` ..... 3670, 3760  
`\rightlinenum` .....  
     9, 290, 986, 994, 4077, 4209, 4450  
`\rightlftab` ..... 3683, 3742  
`\rightnoteuptrue` ..... 2461  
`\rightpstartnum` .... 1061, 1069, 1086  
`\rightrtab` ..... 3696, 3708  
`\rightstartnum` ..... 1053  
`\rigidbalance` ... 1596, 1648, 1655,  
     1691, 1702, 2735, 2742, 2780, 2787  
`\rlap` .. 986, 994, 1061, 1069, 4190, 4450  
`\rmarrpar` ..... 4506, 4515, 4556, 4584  
`\rmfamily` ..... 3929, 3949, 4096  
Robinson, Peter ..... 3  
`\roman` ..... 3615  
`\rtab` ..... 3231, 3705, 3849  
`\rtabtext` ..... 3234, 3723, 3853
- S**
- Sacrobosco ..... 5  
`\savel@dcnote` ..... 2507  
`\sc@n@list` ..... 974, 976  
`\scf` ..... 4206,  
     4303, 4308, 4311, 4312, 4319, 4320  
Schöpf, Rainer ..... 4  
`\section@num` .....  
     104, 121, 123, 124, 166–168, 2328  
`\select@lemmfont` ..... 641, 1153  
`\select@lemmfont` ..... 17,  
     1153, 1334, 1550, 1639, 1682,  
     2352, 3933, 3945, 3958, 3965,  
     4217, 4223, 4495, 4680, 4809, 4814  
`\sen` ..... 4111,  
     4153, 4154, 4156, 4158, 4163–4168  
`\senspeak` ..... 4112, 4152  
`\set@line` ..... 671, 686, 701, 4863  
`\set@line@action` 374, 464, 471, 482, 567  
`\setl@dlp@rbox` . 2484, 2488, 2515, 2527  
`\setl@drp@rbox` . 2485, 2497, 2517, 2525  
`\setl@drpr@box` ..... 2488  
`\setline` ... 10, 63, 613, 645, 775, 4273  
`\setlinenum` ..... 10, 65, 620, 4851  
`\setlp@rbox` ..... 4379, 4384  
`\setmcellcenter` ..... 3516, 3572  
`\setmcellleft` ..... 3489, 3561  
`\setmcellright` ..... 3456, 3550  
`\setmrowcenter` ..... 3570, 3764  
`\setmrowleft` ..... 3559, 3746  
`\setmrowright` ..... 3548, 3712  
`\setprintendlines` ..... 2361, 2386  
`\setprintlines` ..... 1370, 1408  
`\setstanzaindents` .... 18, 3168, 4827  
`\setstanzapenalties` .....  
     .... 19, 3168, 4832, 4833  
`\setstanzavalues` 3158, 3168, 3169, 3219  
`\settcclcenter` ..... 3531, 3577  
`\settcclleft` ..... 3503, 3566  
`\settcclright` ..... 3473, 3555  
`\settoggle` ..... 1161, 1165,  
     1713–1717, 1720, 1759–1763, 1766  
`\setthrowcenter` ..... 3575, 3772  
`\setthrowleft` ..... 3564, 3754  
`\setthrowright` ..... 3553, 3728  
`\Setzen` ..... 3789, 3798, 3800  
Shakespeare, William ..... 192  
`\showlemma` ..... 13, 29, 676, 691  
`\sidenote@margin` ..... 2414, 2520  
`\sidenotemargin` ..... 23, 2414, 4611  
`\sidenotemmargint` ..... 90  
`\skip` ..... 1425, 1444, 1451, 1470,  
     1475, 1507, 1564, 1650, 1697,  
     2091, 2643, 2654, 2689, 2695,  
     2737, 2782, 2811, 2815, 2857,  
     2989, 4103, 4104, 4620–4622, 4818  
`\skip@lockoff` ..... 501, 527  
`\skipnumbering` ..... 11, 630  
`\skipnumbering@reg` ..... 630  
`\spacedcolon` ..... 3929, 3933, 3965  
`\spacefactor` .....  
     2540, 2543, 2554, 2560, 2603, 2609  
`\spaceskip` ..... 1311, 2622, 2718  
`\speak` ..... 4116, 4139  
`\speaker` ..... 4193, 4275, 4292, 4327  
`\splitmaxdepth` ..... 1316  
`\splitoff` ..... 1596  
`\splittopskip` .... 817, 1316, 1598,  
     1646, 1648, 1653, 1655, 1689,  
     1691, 1700, 1702, 2733, 2735,  
     2740, 2742, 2778, 2780, 2785, 2787

- `\spreadmath` ..... 27, [3777](#)
  - `\spreadtext` ..... 27, [3775](#)
  - `\ss` ..... 3222
  - `\stage` ..... 4107, 4128, 4190, 4196
  - `\stanza` 18, [3197](#), 4923, 4926, 4943, 4957
  - `\stanza@count` ..... [3149](#), 3160, 3163, 3165, 3182, 3194, 3201, 3209
  - `\stanza@hang` ..... [3180](#), 3203
  - `\stanza@line` ..... [3180](#), 3209, 3211
  - `\stanza@modulo` ..... 3172, 3175–3177, 3186, 3201
  - `\stanzaindentbase` ..... 18, [3149](#), 3183, 3187, 3192, 3213, 4822, 4846
  - `\startlock` ..... 10, [627](#), 643, 3190
  - `\startstanzahook` ..... 20, [3197](#), 4849
  - `\startsub` ..... 9, 603, 642, 4107
  - `\stepcounter` ..... 2582, 2594, 2867, 2878, 2885, 2894, 3039, 3623
  - `\stepl@dc@colcount` .... [3302](#), 3352, 3364, 3465, 3482, 3496, 3510, 3523, 3538, 3582, 3782, 3791, 3812
  - `\strip@pt` ..... 1503, 2808
  - `\strip@szacnt` ..... [3158](#)
  - `\sub@action` ..... 390, [491](#), 566
  - `\sub@change` ..... 138, 384, 385, 391, 441, 443, 446, 448
  - `\sub@clock` ..... 135, [326](#), 399, 401, 403, 406, 509, 510, 512, 513, 531, 532, 534, 857, 928, 930, 931, 933, 1008, 1044, 1046, 1048, 4403, 4464, 4466, 4468
  - `\sub@off` ..... [440](#), 609
  - `\sub@on` ..... [440](#), 605
  - `\subline` ..... 2252–2255
  - `\subline@num` ..... 133, 296, 297, [323](#), 407, 411, 417, 428, 452, 453, 455, 467, 485, 572, 576, 858, 863, 886, 893, 948–950, 2276, 4080, 4083, 4392–4394
  - `\sublinenumberstyle` ..... 10, [281](#)
  - `\sublinenumincrement` ..... 7, 9, [241](#)
  - `\sublinenumr@p` ..... [281](#)
  - `\sublinenumrep` ..... [281](#), 297, 1411, 1415, 2276, 2388, 2392
  - `\sublineref` ..... 21, [2284](#)
  - `\sublines@false` ... 136, [324](#), 388, 918
  - `\sublines@true` ..... [324](#), 386, 916
  - `\sublock@disp` ..... [272](#), 1010, 1014, 1018, 4405, 4409, 4413
  - `\sublockdisp` ..... 53, [272](#)
  - `\subsection` ..... 3888
  - `\subsubsection` ..... 2011
  - Sullivan, Wayne ..... 4, 5, 18, 30, 37, 42, 98, 99, 121, 152, 195
  - `\symlinenum` ..... 14, [1726](#)
  - `\symlinenum@A` ..... 1726
  - `\symlinenum@B` ..... 1727
  - `\symlinenum@C` ..... 1728
  - `\symlinenum@D` ..... 1729
  - `\symlinenum@E` ..... 1730
  - `\symlinenum@Z` ..... 1731
  - `\symplinenum` ..... [1723](#), 1726–1731
  - `\sza@penalty` ..... [3180](#), 3207, 3208
- T**
- `\tabellzwischen` ..... [3780](#), 3788
  - `\tabelskip` 3792, 3832–3834, 3840–3842
  - `\tabHilfbox` ..... 3831, 3833, 3835, 3839, 3841, 3843, [3844](#)
  - `\tabhilfbox` ..... 3830, 3832, 3834, 3838, 3840, 3842, [3844](#)
  - `\tableofcontents` ..... 3883
  - Tapp, Christian ..... 3
  - `\textbf` ..... 4541, 4846, 4892, 4902
  - `\textheight` ..... 2205, 4796
  - `\textnormal` ..... 1340–1342, 4258, 4276, 4295, 4305, 4306, 4314, 4316, 4318, 4322, 4331, 4658
  - `\textrm` ..... 4111, 4112, 4303
  - `\textsc` ..... 4719
  - `\textsf` ..... 3885, 3892
  - `\textsuperscript` ..... 2533, 2591, 2617, 2640, 4542
  - `\textwidth` ..... 2967, 3009, 4797
  - `\thanks` ..... 3879
  - `\theadcolcount` .... [3614](#), 3621, 3624
  - `\thefootnoteA` 24, 2583, [2588](#), 2591, 2595
  - `\thefootnoteB` .. 2868, [2872](#), 2879, 3872
  - `\thefootnoteC` ..... 2886, [2889](#), 2895
  - `\thelabidx` ..... 3040, 3043
  - `\thempfn` ..... 2969, 2995, 3022
  - `\thempfootnote` ..... 2969, 2995, 3022
  - Theodosius ..... 5
  - `\thepage` ..... 595, 2240, 3043
  - `\thepageline` ..... 3042, 3068, 3073, 3076, 3091, 3100
  - `\thepstart` ..... 8, [736](#), 738, 759, 809, 1082, 1089, 1405
  - `\thepstartL` ..... 1402
  - `\thepstartR` ..... 1400

- `\thestanzanum` ..... 4846  
`\thinspace` ..... 1342, 3929, 3949  
`\thr@@` ..... 223, 512, 521, 532,  
     539, 923, 931, 1621, 1624, 1648,  
     1655, 2435, 2755, 2758, 2780, 2787  
`\threecolfootfmt` ... 1611, 1631, 3936  
`\threecolfootfmtX` ..... 2746, 2766  
`\threecolfootgroup` ..... 1612, 1645  
`\threecolfootgroupX` ..... 2747, 2777  
`\threecolfootsetup` ..... 1613, 1619  
`\threecolfootsetupX` ..... 2748, 2753  
`\threecolvfootnote` ..... 1610, 1626  
`\threecolvfootnoteX` ..... 2745, 2760  
`\tiny` ..... 4206  
`\title` ..... 3878, 4704  
`\tolerance` ..... 1635,  
     1678, 2726, 2771, 3940, 3954, 4229  
`\twocolfootfmt` ..... 1659, 1667, 3950  
`\twocolfootfmtX` ..... 2700, 2721  
`\twocolfootgroup` ..... 1660, 1667  
`\twocolfootgroupX` ..... 2701, 2732  
`\twocolfootsetup` ..... 1661, 1667  
`\twocolfootsetupX` ..... 2702, 2707  
`\twocolvfootnote` ..... 1658, 1667  
`\twocolvfootnoteX` ..... 2699, 2714
- U**
- `\underbrace` ..... 3229  
`\unhbox` 814, 1540, 1559, 1561, 1569,  
     1571, 1578, 1582, 2853, 2855,  
     2862, 2864, 3834, 3835, 3842, 3843  
`\unkern` ..... 2541  
`\unpenalty` ..... 1543, 1575  
`\unvbox` ..... 821,  
     1321, 1448, 1454, 1525, 1538,  
     1557, 1567, 1607, 2070, 2090,  
     2095, 2108, 2122–2126, 2130,  
     2132, 2150, 2188, 2627, 2651,  
     2657, 2833, 2851, 2860, 2906,  
     2913–2915, 2922, 2927, 2986, 2992  
`\unvxx` 1516, 1530, 1537, 2825, 2838, 4643  
`\usepackage` ..... 3867, 3923, 4063,  
     4187, 4341, 4349, 4606, 4607, 4794  
`\usingcritext` ..... 3855  
`\usingdtext` ..... 3855
- V**
- `\vAfootnote` ..... 1175, 1179  
`\valign` ..... 1599  
`\value` ..... 3176, 3181, 3620  
`Vamana` ..... 5  
`\variab` ..... 3385, 3711,  
     3727, 3745, 3753, 3763, 3771, 3804  
`\vbadness` ..... 816, 1598  
`\vbfnoteX` ..... 2661, 2666  
`\vBfootnote` ..... 1187, 1192  
`\vbox` ..... 759, 1320,  
     1514, 1524, 1528, 1538, 2067,  
     2087, 2107, 2213, 2217, 2492,  
     2494, 2496, 2501, 2503, 2626,  
     2823, 2832, 2836, 2905, 2921,  
     2964, 3591, 3594, 3599, 3602,  
     3606, 3608, 3609, 3653, 3657,  
     3662, 3673, 3686, 3699, 4382, 4641  
`\vCfootnote` ..... 1200, 1205  
`\vDfootnote` ..... 1213, 1218  
`\vEfootnote` ..... 1226, 1231  
`\vffootnote` ..... 4362, 4366, 4384  
`\vfil` ..... 1599, 2217,  
     3592, 3595, 3600, 3603, 3606, 3609  
`\vfootnoteA` ..... 2585  
`\vfootnoteB` ..... 2870  
`\vfootnoteC` ..... 2888  
`\vgfootnote` ..... 4372, 4376, 4386  
`\vl@dbfnote` ..... 2571  
`\vl@dcnote` ..... 2479, 2484  
`\vl@dlsnote` ..... 2465, 2484  
`\vl@drsnote` ..... 2472, 2484  
`\vnumfootnoteX` ..... 2670, 2682  
`\vrule` ... 3591, 3594, 3599, 3602, 3606  
`\vsize` ..... 1456, 2693, 4102  
`\vsplit` ..... 820, 1606, 2188
- W**
- `\wd` ... 809, 832, 1518, 1532, 2827,  
     2840, 3308, 3309, 3453, 3666,  
     3675, 3678, 3688, 3691, 3700,  
     3832, 3833, 3840, 3841, 4114, 4645  
 Whitney, Ron ..... 4  
`\widowpenalty` ..... 790, 1123  
 Wujastyk, Dominik ..... 2, 4
- X**
- `\x@lemma` . 678–680, 693–695, 4870–4872  
`\xcritext` ..... 3312, 3443  
`\xedindex` ..... 3318, 3415, 3432, 3445  
`\xedlabel` ..... 3316, 3454  
`\xedtext` ..... 3312, 3444  
`\xleft@appenditem` ..... 315  
`\xlineref` ..... 22, 2281

|                                           |                                                                                                                                                                                                                                                     |                                      |                                         |
|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------|
| <code>\xpageref</code> . . . . .          | 22, <a href="#">2278</a>                                                                                                                                                                                                                            | <code>\xspaceskip</code> . . . . .   | 1311, 2622, 2718                        |
| <code>\xparafootfmt</code> . . . . .      | 4806, 4817, 4914                                                                                                                                                                                                                                    | <code>\xsublineref</code> . . . . .  | 22, <a href="#">2284</a>                |
| <code>\xright@appenditem</code> . . . . . |                                                                                                                                                                                                                                                     | <code>\xxref</code> . . . . .        | 22, <a href="#">2309</a>                |
| . . . . .                                 | <a href="#">309</a> , 480, 481, 483, 490,<br>492, 494, 496, 506, 508, 517,<br>528, 530, 537, 550, 551, 560,<br>574, 1161, 1165, 1175, 1187,<br>1200, 1213, 1226, 1246, 1259,<br>1272, 1285, 1298, 2274, 2465,<br>2472, 2479, 2573, 2661, 4362, 4372 |                                      |                                         |
|                                           |                                                                                                                                                                                                                                                     | <b>Y</b>                             |                                         |
|                                           |                                                                                                                                                                                                                                                     | <code>\yparafootfmt</code> . . . . . | 4811, 4898                              |
|                                           |                                                                                                                                                                                                                                                     | <b>Z</b>                             |                                         |
|                                           |                                                                                                                                                                                                                                                     | <code>\z@skip</code> . . . . .       | 1311, 1317, 2622, 2718                  |
|                                           |                                                                                                                                                                                                                                                     | <code>\zz@@@</code> . . . . .        | <a href="#">2224</a> , 2230, 2311, 2313 |

## Change History

|         |                                                                                                                                                                                 |    |         |                                                                                                                                                                                             |       |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| v0.1    | General: First public release . . . . .                                                                                                                                         | 1  | v0.13   | General: New stanzaindent repetition counter to repeat stanza indents every $n$ verses. . . . .                                                                                             | 1, 19 |
| v0.10   | General: Corrections to <code>\section</code> and other titles in numbered sections . . . . .                                                                                   | 1  |         | <code>\managestanza@modulo</code> : New stanzaindent repetition counter to repeat stanza indents every $n$ verses. . . . .                                                                  | 154   |
| v0.11   | General: Makes it possible to add a symbol on each verse's hanging, as in French typography. Redefines the command <code>\hangingsymbol</code> to define the character. . . . . | 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     |
| v0.12   | General: For compatibility with ledpar, possibility to use <code>\autopar</code> on the right side. . . . .                                                                     | 1  | v0.14   | General: Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph. . . . .                                                                   | 1     |
|         | Possibility to number the pstart with the commands <code>\numberpstarttrue</code> . . . . .                                                                                     | 1  |         | <code>\edlabel</code> : Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph. . . . .                                                    | 121   |
|         | Possibility to number <code>\pstart</code> . . . . .                                                                                                                            | 8  | v0.15   | General: Line numbering can be reset at each pstart. . . . .                                                                                                                                | 43    |
|         | <code>\ifledRcol</code> : Added <code>\ifledRcol</code> and <code>\ifnumberingR</code> for/from ledpar . . . . .                                                                | 41 |         | New management of hangingsymbol insertion, preventing undesirable insertions. . . . .                                                                                                       | 152   |
| v0.12.1 | General: Don't number <code>\pstarts</code> of stanza. . . . .                                                                                                                  | 1  |         | Possibility to print <code>\pstart</code> number in side. . . . .                                                                                                                           | 8     |
|         | The numbering of <code>\pstarts</code> restarts on each <code>\beginnumbering</code> . . . . .                                                                                  | 1  |         |                                                                                                                                                                                             |       |



|                                                                                                                                                   |         |                                                                                                                                                                                                                    |     |
|---------------------------------------------------------------------------------------------------------------------------------------------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <code>\affixline@num</code> : Line numbering can be disabled. . . . .                                                                             | 78      | Major changes to insert code when memoir is loaded . . . .                                                                                                                                                         | 118 |
| <code>\printlines</code> : Line numbering can be reset at each pstart. . . . .                                                                    | 93      | <code>\doxtrafeet</code> : Renamed <code>\doxtrafeet</code> to <code>\l@ddoxtrafeet</code> . . . . .                                                                                                               | 117 |
| v0.17                                                                                                                                             |         | <code>\edlabel</code> : Tweaked <code>\edlabel</code> to get correct page numbers . . .                                                                                                                            | 121 |
| General: New new management of hangingsymbol insertion, preventing undesirable insertions.                                                        | 152     | <code>\l@d@makecol</code> : Rewrote <code>\@makecol</code> , calling it <code>\l@d@makecol</code> . . . .                                                                                                          | 116 |
| v0.18                                                                                                                                             |         | <code>\l@ddodoreinxtrafeet</code> : Renamed <code>\dodoreinxtrafeet</code> to <code>\l@ddodoreinxtrafeet</code> . . . .                                                                                            | 118 |
| General: Debug in lineation command . . . . .                                                                                                     | 8       | <code>\l@ddofootinsert</code> : Renamed <code>\dofootinsert</code> as <code>\l@ddofootinsert</code> . . . . .                                                                                                      | 117 |
| New generic commands to customize footnote display. . .                                                                                           | 14, 105 | <code>\m@m@makecolintro</code> : Added <code>\m@m@makecolfloats</code> , <code>\m@m@makecoltext</code> and <code>\m@m@makecolintro</code> . . .                                                                    | 116 |
| v0.18.1                                                                                                                                           |         | <code>\morenoexpands</code> : Removed some <code>\lets</code> from <code>\no@expands</code> . These were in EDMAC but I feel that they should not have been as they disabled page/line refs in footnotes . . . . . | 66  |
| General: <code>\inplaceoflemmaseparator</code> added. . . . .                                                                                     | 15      | <code>\zz@@@</code> : Minor change to <code>\zz@@@</code> .                                                                                                                                                        | 121 |
| <code>\lemma</code> can contain commands. . . .                                                                                                   | 12      | v0.2.2                                                                                                                                                                                                             |     |
| <code>\nosum</code> deleted. . . . .                                                                                                              | 14      | General: Added the Dekker example . . . . .                                                                                                                                                                        | 201 |
| Options nonum and noseup in <code>\Xfootnote</code> . . . . .                                                                                     | 12      | Improved paragraph footnotes . .                                                                                                                                                                                   | 1   |
| Options nonum and noseup in <code>\Xfootnotes</code> . . . . .                                                                                    | 86      | New Dekker example . . . . .                                                                                                                                                                                       | 1   |
| v0.18.2                                                                                                                                           |         | <code>\footfudgefiddle</code> : Added <code>\footfudgefiddle</code> . . . . .                                                                                                                                      | 97  |
| General: Code refactoring in options of <code>\Xfootnotes</code> . . . . .                                                                        | 86      | <code>\l@d@section</code> : Used <code>\providecommand</code> for <code>\@gobblethree</code> to avoid clash with the amsfonts package . . . . .                                                                    | 127 |
| Debug nexted <code>\edtext</code> with v0.18.1 functions. . . . .                                                                                 | 11      | <code>\line@list@stuff</code> : Added initial write of page number in <code>\line@list@stuff</code> . . . . .                                                                                                      | 61  |
| Possibility to have commands in sidenotes. . . . .                                                                                                | 23      | <code>\para@footsetup</code> : Added <code>\footfudgefiddle</code> to <code>\para@footsetup</code> . . . . .                                                                                                       | 97  |
| <code>\morenoexpands</code> : Change to be compatible with 0.18.1 changes, and to have formatting commands in nexted <code>\edtext</code> . . . . | 66      | <code>\para@footsetupX</code> : Added <code>\footfudgefiddle</code> to <code>\para@footsetupX</code> . . . . .                                                                                                     | 140 |
| v0.2                                                                                                                                              |         | v0.3                                                                                                                                                                                                               |     |
| General: Added tabmac code, and extended indexing . . . . .                                                                                       | 1       | <code>\@l@reg</code> : Added a bunch of code to <code>\@l</code> for handling <code>\setlinenum</code> . .                                                                                                         | 54  |
| <code>\ifl@dmemoir</code> : Added <code>\ifl@dmemoir</code> for memoir class having been used . . . . .                                           | 37      | <code>\@lab</code> : Replaced <code>\the\line@num</code> by <code>\linenumr@p\line@num</code> in <code>\@lab</code> , and similar for sub-lines                                                                    | 123 |
| <code>\ledmac@error</code> : Added <code>\ledmac@error</code> and replaced error messages . .                                                     | 38      |                                                                                                                                                                                                                    |     |
| <code>\morenoexpands</code> : Added <code>\l@dtabnoexpands</code> to <code>\no@expands</code> . . . . .                                           | 66      |                                                                                                                                                                                                                    |     |
| v0.2.1                                                                                                                                            |         |                                                                                                                                                                                                                    |     |
| <code>\@lab</code> : Removed page setting from <code>\@lab</code> . . . . .                                                                       | 123     |                                                                                                                                                                                                                    |     |
| General: Added text about normal labeling . . . . .                                                                                               | 22      |                                                                                                                                                                                                                    |     |
| Bug fixes and match with mempatch v1.8 . . . . .                                                                                                  | 1       |                                                                                                                                                                                                                    |     |

- General: Added the Braonain example . . . . . 205  
Includes edstanza and more . . . 1
- `\ledlinenum`: Added `\linenumr@p` and `\sublinenum@rep` to `\leftlinenum` and `\rightlinenum` . . . . . 47
- `\linenumberlist`: Added `\linenumberlist` mechanism . 37
- `\printendlines`: Added `\linenumr@p` and `\sublinenumr@p` to `\printendlines` . . . . . 128
- `\printlines`: Added `\linenumr@p` and `\sublinenumr@p` to `\printlines` . . . . . 93
- `\sublinenumr@p`: Added `\linenumberstyle` and `\sublinenumberstyle` . . . 46
- v0.3.1  
General: Not released. Added remarks about the parallel package . . . . . 1
- v0.31  
General: Added remarks about ledmac/parallel package incompatibility . . . . . 32
- v0.4  
`\@iiiminipage`: Modified kernel `\@iiiminipage` and `\endminipage` to cater for critical footnotes . . . . . 144  
General: Added `\showlemma` to `\edtext` (and `\critext`) . . . 68  
Added minipage, etc., support . 1
- `ledgroup`: Added ledgroup environment . . . . . 145
- `ledgroupsize`: Added ledgroup-sized environment . . . . . 146
- `\footnormal`: Added minpage footnote setup to `\footnormal` . . 95
- `\ifledfinal`: Added final/draft options . . . . . 37
- `\l@dfeetendmini`: Added `\l@dfeetbeginmini`, `\l@dfeetendmini` and all their supporting code 144
- `\mpEfootins`: Added `\mpAfootins` and friends . . . . . 87
- `\mpEfootnote`: Added `\mpAfootnote` and friends . . . . . 88
- `\mpfootinsA`: Familiar footnotes extended for minipages . . . . 134
- `\mpfootinsB`: Familiar footnotes extended for minipages . . . . 142
- `\mpfootinsC`: Familiar footnotes extended for minipages . . . . 143
- `\mpnormalfootgroup`: Added `\mpnormalfootgroup` . . . . . 94
- `\mpnormalvfootnote`: Added `\mpnormalvfootnote` . . . . . 90
- `\showlemma`: Added `\showlemma` . 37
- v0.4.1  
`\@opxtrafeetii`: Added `\@opxtrafeetii` . . . . . 117  
General: Added code for changing `\@docclearpage` . . . . . 119  
Let ledmac take advantage of memoir's indexing . . . . . 147  
Not released. Minor editorial improvements and code tweaks . . 1  
Only change `\@footnotetext` and `\@footnotemark` if memoir not used . . . . . 132
- `\addfootins`: Added `\addfootins` 119
- `\addfootinsX`: Added minpage setup to `\addfootinsX` . . . . 143
- `\doxtrafeetii`: Changed `\doxtrafeetii` code for easier extensions . . . . . 117
- `\ledfootinsdim`: Added `\ledfootinsdim` . . . . . 95
- v0.5  
`\@footnotetext`: Enabled regular `\footnote` in numbered text 133
- `\@xympar`: Eliminated `\marginpar` disturbance . . . . . 129
- General: Added left and right side notes . . . . . 129  
Added sidenotes, familiar footnotes in numbered text . . . . . 1
- v0.5.1  
General: Added moveable side note 129  
Fixed right line numbers killed in v0.5 . . . . . 1
- `\affixline@num`: Changed `\affixline@num` to cater for sidenotes . . . . . 77
- `ledgroupsize`: Only change `\hsize` in ledgroupsize environment otherwise page number can be in wrong place . . . . 146

|                                                                                                                                                                          |     |                                                                                                                                                                  |     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <code>\l@dgetsidenote@margin:</code> Added<br><code>\sidenotemargin</code> and<br><code>\sidenote@margin</code> . . . . .                                                | 129 | <code>\do@linehook:</code> Added <code>\do@linehook</code><br>for use in <code>\do@line</code> . . . . .                                                         | 74  |
| v0.6                                                                                                                                                                     |     | <code>\endnumbering:</code> Changed<br><code>\endnumbering</code> for <code>ledpar</code> . . .                                                                  | 42  |
| <code>\@l@reg:</code> Added <code>\fix@page</code> to <code>\@l</code> . . . . .                                                                                         | 54  | <code>\f@x@l@cks:</code> Added <code>\ch@cksub@l@ck</code> ,<br><code>\ch@ck@l@ck</code> and <code>\f@x@l@cks</code> . . . . .                                   | 80  |
| Extended <code>\@l</code> to include the page<br>number . . . . .                                                                                                        | 54  | <code>\footsplitskips:</code> Added<br><code>\footsplitskips</code> for use in<br>many footnote styles . . . . .                                                 | 90  |
| <code>\@lopR:</code> Added <code>\@pend</code> , <code>\@pendR</code> ,<br><code>\@lopL</code> and <code>\@lopR</code> in anticipa-<br>tion of parallel processing . . . | 56  | <code>\get@linelistfile:</code> Added<br><code>\get@linelistfile</code> . . . . .                                                                                | 53  |
| General: Changed version of the<br>Dekker example . . . . .                                                                                                              | 201 | <code>\ifledRcol:</code> Added <code>\l@dnumstartL</code> ,<br><code>\ifl@dpairing</code> and <code>\ifpst@rted</code><br>for/from <code>ledpar</code> . . . . . | 41  |
| Fixed long paragraphs looping . .                                                                                                                                        | 1   | <code>\initnumbering@reg:</code> Added<br><code>\initnumbering@reg</code> . . . . .                                                                              | 41  |
| Fixed minor typos . . . . .                                                                                                                                              | 1   | <code>\l@dcsnotetext:</code> Added <code>\l@emptyd@ta</code><br>. . . . .                                                                                        | 74  |
| Prepared for <code>ledpar</code> package . . .                                                                                                                           | 1   | <code>\l@ddofootinsert:</code> Deleted<br><code>\page@start</code> from <code>\l@ddofootinsert</code><br>. . . . .                                               | 117 |
| <code>\fix@page:</code> Added <code>\last@page@num</code><br>and <code>\fix@page</code> . . . . .                                                                        | 55  | <code>\l@dgetline@margin:</code> Added<br><code>\l@dgetline@margin</code> . . . . .                                                                              | 44  |
| <code>\footnoteA:</code> Modified <code>\footnoteA</code><br>and friends to include<br><code>\@thefnmarkA</code> etc . . . . .                                           | 134 | <code>\l@dgetlock@disp:</code> Added<br><code>\l@dgetlock@disp</code> . . . . .                                                                                  | 45  |
| <code>\new@line:</code> Extended <code>\new@line</code> to<br>output page numbers . . . . .                                                                              | 61  | <code>\l@dgetsidenote@margin:</code> Added<br><code>\l@dgetsidenote@margin</code> . . . . .                                                                      | 129 |
| <code>\page@start:</code> Made <code>\page@start</code> a<br>no-op . . . . .                                                                                             | 61  | <code>\l@drrsn@te:</code> Added <code>\l@drrsn@te</code><br>and <code>\l@drrsn@te</code> for use in<br><code>\do@line</code> . . . . .                           | 74  |
| <code>\vl@dbfnote:</code> Changed <code>\l@dbfnote</code><br>and <code>\vl@dbfnote</code> as originals<br>could give incorrect markers in<br>the footnotes . . . . .     | 133 | <code>\l@dunboxmpfoot:</code> Added<br><code>\l@dunboxmpfoot</code> containing<br>some common code . . . . .                                                     | 145 |
| v0.7                                                                                                                                                                     |     | <code>\l@dzeropenalties:</code> Added<br><code>\l@dzeropenalties</code> . . . . .                                                                                | 72  |
| <code>\@l@reg:</code> Added <code>\@l@reg</code> . . . . .                                                                                                               | 54  | <code>\ledlinenum:</code> Added <code>\ledlinenum</code><br>for use by <code>\leftlinenum</code> and<br><code>\rightlinenum</code> . . . . .                     | 47  |
| <code>\@ref@reg:</code> Added <code>\@ref@reg</code> . . .                                                                                                               | 59  | <code>\line@list@stuff:</code> Deleted<br><code>\page@start</code> from <code>\line@list@stuff</code><br>. . . . .                                               | 61  |
| General: Added bits about <code>ledpar</code><br>package . . . . .                                                                                                       | 32  | <code>\list@clearing@reg:</code> Added<br><code>\list@clearing@reg</code> . . . . .                                                                              | 52  |
| <code>ledmac</code> having been available for<br>2 years, deleted the commented<br>out original <code>edmac</code> texts . . . . .                                       | 1   | <code>\n@num@reg:</code> Added <code>\n@num</code> . . . .                                                                                                       | 59  |
| Maïeul Rouquette new main-<br>tainer . . . . .                                                                                                                           | 1   | <code>\normalbfnoteX:</code> Removed<br>extraneous space from<br><code>\normalbfnoteX</code> . . . . .                                                           | 136 |
| Made macros of all messages . .                                                                                                                                          | 38  | <code>\resumenumbering:</code> Changed<br><code>\resumenumbering</code> for <code>ledpar</code> . .                                                              | 42  |
| Replaced all <code>\interAfootnotelinepenalty</code> ,<br><code>\line@list@stuff</code> ,<br>etc., by just <code>\interfootnotelinepenalty</code><br>. . . . .           | 1   |                                                                                                                                                                  |     |
| Tidying up for <code>ledpar</code> and<br><code>ledarab</code> packages . . . . .                                                                                        | 1   |                                                                                                                                                                  |     |
| <code>\affixline@num:</code> Added skipnum-<br>mering to <code>\affixline@num</code> . . .                                                                               | 77  |                                                                                                                                                                  |     |
| <code>\do@actions@fixedcode:</code> Added<br><code>\do@actions@fixedcode</code> . . . .                                                                                  | 76  |                                                                                                                                                                  |     |
| <code>\do@actions@next:</code> Added number<br>skipping to <code>\do@actions</code> . . . .                                                                              | 75  |                                                                                                                                                                  |     |

|                                                                |         |        |                                                             |
|----------------------------------------------------------------|---------|--------|-------------------------------------------------------------|
| <code>\setprintendlines:</code>                                | Added   | v0.8   |                                                             |
| <code>\setprintendlines</code> for use by                      |         |        | General: Bug on endnotes fixed :                            |
| <code>\printendlines</code> . . . . .                          | 127     |        | in a // text, all endnotes will                             |
| <code>\setprintlines:</code> Added <code>\setprintlines</code> |         |        | print and be placed at the ends                             |
| for use by <code>\printlines</code> . . . . .                  | 92      |        | of columns () . . . . .                                     |
| <code>\skipnumbering@reg:</code>                               | Added   | v0.8.1 |                                                             |
| <code>\skipnumbering</code> and supports                       | 63      |        | General: Bug on <code>\edtext</code> ; <code>\critex</code> |
| <code>\sublinenumincrement:</code>                             | Added   |        | ; <code>\lemma</code> fixed : we can now us                 |
| <code>\firstlinenum</code> , <code>\linenumincrement</code> ,  |         |        | non switching commands . . . . .                            |
| <code>\firstsublinenum</code> and                              |         |        |                                                             |
| <code>\linenumincrement</code> . . . . .                       | 45      | v0.9   |                                                             |
| <code>\sublinenumr@p:</code> Using <code>\linenumrep</code>    |         |        | General: No more ledpatch. All                              |
| instead of <code>\linenumr@p</code> . . . . .                  | 46      |        | patches are now in the main                                 |
| Using <code>\sublinenumrep</code> instead of                   |         |        | file. . . . .                                               |
| <code>\sublinenumr@p</code> . . . . .                          | 46      | v0.9.1 |                                                             |
| <code>\vnumfootnoteX:</code>                                   | Removed |        | General: Fix some bugs linked to in-                        |
| extraneous space from                                          |         |        | tegrating ledpatch on the main                              |
| <code>\vnumfootnoteX</code> . . . . .                          | 137     |        | file. . . . .                                               |