

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 v0.18.1, last revised 2012/09/02.

[†]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	87
22.1 Fonts	88
22.2 Outer-level footnote commands	88
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	141

27.2.3 Paragraped footnotes	143
27.3 Other series footnotes	145
28 Minipages and such	146
29 Indexing	149
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
B Examples	179
B.1 Version 0.18.1	179
B.2 Simple example	188
B.3 General example of features	189
B.4 Gascoigne	192
B.5 Shakespeare	195
B.6 Classical text edition	198
B.7 Nijmegen	204
B.8 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:

I saw my friend <code>\edtext{Smith}{</code>	1 I saw my friend
<code>\Afootnote{Jones C, D.}}</code>	2 Smith on Tuesday.
on Tuesday.	<u>2 Smith]</u> Jones C, D.

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

The `<lemma>` may contain further `\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:

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

However, `\edtext` cannot handle overlapping but unnested notes—for example, one note covering lines 10–15, and another covering 12–18; 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.

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

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

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

To specify these multiples one invokes, for example

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

The numerical entries must be whole numbers, 0 or greater, separated by commas without embedded spaces. The first entry gives the hanging indentation to be used if the stanza line requires more than one print line. If it is known that each stanza line will fit on a single print line, then this first entry should be 0; T_EX 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 the `\edlabel{<lab>}` command occurred.
`\sublineref`

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

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

`\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:

`\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}[2012/09/02 v0.18.1 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 `\resumenumbers` 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 `\resumenumbers` `\ifnumbering` flag set to true, to show that numbering continues across the gap.<sup>20</sup>

```

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

```

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

```

163 \newcommand*{\resumenumbers}{%
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 specify the current lineation system:

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

`\bypstart@true`  
`\bypstart@false`  
`\ifbypage@`  
`\bypage@true`  
`\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 }
```

```

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@dttempcntb>\m@ne
276 \global\sublock@disp=\l@dttempcntb
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-lineated text, and those sub-line numbers should not return to zero until the next change in the major line number. This is common in the typesetting of English Renaissance verse drama, in which stage directions are given sub-line numbers: a single line of verse may be interrupted by several stage directions. |
| <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>         | <ul style="list-style-type: none"> <li>• <code>\line@list</code>: the page and line numbers for every lemma marked by <code>\edtext</code>. There are seven pieces of information, separated by vertical bars:</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>\actions@list</code>             | <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 instructions to the line-list file. When sub-lineation is in effect, the line number



counter is frozen and the sub-line counter advances instead. If one of these commands appears in the middle of a line, it 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—`TeX` seems to get confused about the difference between alignment parameters and macro parameters; those that use temporary control sequences; and those that look carefully at what the current font is.

(The `\copyright` macro defined in `PLAIN TeX` has this sort of problem as well, but 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*{\no@expands}{\let\rm=0\let\it=0\let\sl=0\let\bf=0\let\tt=0%
641 \let\b=0\let\c=0\let\d=0\let\t=0%
642 \let\select@lemmafnt=0%
643 \def\protect{\noexpand\protect\noexpand}%
644 \let\startsub=\relax \let\endsub=\relax
645 \let\startlock=\relax \let\endlock=\relax
646 \let\edlabel=\@gobble
647 % \let\edpageref=\@gobble
648 % \let\lineref=\@gobble
649 % \let\sublineref=\@gobble
650 \let\setline=\@gobble \let\advanceline=\@gobble
651 \let\critext=\dummy@text
652 \let\edtext=\dummy@edtext
653 \l@dtabnoexpands
654 \morenoexpands}
655 \let\morenoexpands=\relax
656
```

`\critext` Now we begin `\critext` itself. The definition requires a / after the arguments: this eliminates the possibility of problems about knowing where #2 ends. This

also changes the handling of spaces following an invocation of the macro: normally such spaces are skipped, but in this case they're significant because #2 is a 'delimited parameter'. Since `\critext` is always used in running text, it seems more appropriate to pay attention to spaces than to skip them.

When executed, `\critext` first ensures that we're in horizontal mode.

```
657 \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.

```
658 \begingroup
659 \noexpands
660 \global\newcommand{\@tag}{#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`.

```
661 \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`.

```
662 \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.

```
663 \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.

```
664 \flag@start
665 \endgroup
666 \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.

```
667 \ifx\end@lemmas\empty \else
```

```

668 \glp\end@lemmas\to\x@lemma
669 \x@lemma
670 \global\let\x@lemma=\relax
671 \fi
672 \flag@end}

```

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

```

\edtext
673 \newcommand{\edtext}[2]{\leavevmode
674 \begingroup
675 \global\newcommand{\@tag}{#1}%
676 \set@line
677 \global\insert@count=0
678 \ignorespaces #2\relax
679 \flag@start
680 \endgroup
681 \showlemma{#1}%
682 \ifx\end@lemmas\empty \else
683 \glp\end@lemmas\to\x@lemma
684 \x@lemma
685 \global\let\x@lemma=\relax
686 \fi
687 \flag@end}
688

```

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

```

689 \newif\ifnumberline
690 \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.

```

691 \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.

```

692 \ifx\line@list\empty
693 \global\noteschanged@true
694 \xdef\l@d@nums{000|000|000|000|000|000|000|\edfont@info}%
695 \else
696 \glp\line@list\to\@tempb
697 \xdef\l@d@nums{\@tempb|\edfont@info}%
698 \global\let\@tempb=\undefined
699 \fi}
700

```

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

```
701 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
702
```

## 20.2 Substitute lemma

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

```
703 \newcommand*{\lemma}[1]{\global\renewcommand{\@tag}{#1}\ignorespaces}
```

## 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.

```
704 \newcommand*{\linenum}[1]{%
705 \xdef\@tempa{#1|||||\\noexpand\\l@d@nums}%
706 \global\let\l@d@nums=\empty
707 \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.

```
708 \def\line@set#1|#2|#3|#4\\{%
709 \gdef\@tempb{#1}%
710 \ifx\@tempb\empty
711 \l@d@add{#3}%
712 \else
713 \l@d@add{#1}%
714 \fi
715 \gdef\@tempb{#4}%
716 \ifx\@tempb\empty\else
717 \l@d@add{|}\line@set#2\\#4\\%
718 \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`.

```
719 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
720
```



## 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`

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

```

721 \newbox\raw@text
722 \newif\ifnumberedpar@
723 \newcount\num@lines
724 \newbox\one@line
725 \newcount\par@line

```

|                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\pstart</code><br><code>\numberpstarttrue</code><br><code>\numberpstartfalse</code><br><code>thepstart</code> | <p><code>\pstart</code> starts the paragraph by clearing the <code>\inserts@list</code> list and other relevant variables, and then arranges for the subsequent text to go into the <code>\raw@text</code> box. <code>\pstart</code> needs to appear at the start of every paragraph that's to be numbered; the <code>\autopar</code> command below may be used to insert these commands automatically.</p> |
|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

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 numbering of `\pstarts`, insert

```
\setcounter{pstart}{0}
```

```

726
727 \newcounter{pstart}
728 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
729 \newif\ifnumberpstart
730 \numberpstartfalse
731 \newcommand*\pstart{
732 \if@nbreak
733 \let\@oldnbreak\@nbreaktrue

```

```

734 \else
735 \let\@oldnbreak\@nbreakfalse
736 \fi
737 \@nbreaktrue
738 \ifnumbering \else
739 \led@err@PstartNotNumbered
740 \beginnumbering
741 \fi
742 \ifnumberedpar@
743 \led@err@PstartInPstart
744 \pend
745 \fi
746 \list@clear{\inserts@list}%
747 \global\let\next@insert=\empty
748 \begingroup\normal@pars
749 \global\setbox\raw@text=\vbox\bgroup\ifautopar\else\ifnumberpstart\ifinstanza\else\ifsi
750 \numberedpar@true}

```

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

```

751 \newcommand*\pend{\ifnumbering \else
752 \led@err@PendNotNumbered
753 \fi
754 \ifnumberedpar@ \else
755 \led@err@PendNoPstart
756 \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.

```

757 \l@dzzeropenalties
758 \endgraf\global\num@lines=\prevgraf\egroup
759 \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`.

```

760 \csnumdef{pstartline}{0}
761 \loop\ifvbox\raw@text
762 \csnumdef{pstartline}{\pstartline+1}%
763 \do@line
764 \ifbypstart@%
765 \ifnumequal{\pstartline}{1}{\setline{1}}{}%
766 \fi
767 \repeat

```

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

```

768 \flush@notes
769 \endgroup
770 \ignorespaces
771 \ifnumberpstart
772 \pstartnumtrue
773 \fi
774 \@oldnobreak
775 \addtocounter{pstart}{1}}
776

```

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

```

777 \newcommand*{\l@dzeropenalties}{%
778 \brokenpenalty \z@ \clubpenalty \z@
779 \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
780 \postdisplaypenalty \z@ \widowpenalty \z@}
781

```

`\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.

```

782 \newif\ifautopar
783 \autoparfalse
784 \newcommand*{\autopar}{
785 \ifledRcol
786 \ifnumberingR \else

```

```

787 \led@err@AutoparNotNumbered
788 \beginnumberingR
789 \fi
790 \else
791 \ifnumbering \else
792 \led@err@AutoparNotNumbered
793 \beginnumbering
794 \fi
795 \fi
796 \autopartrue
797 \everypar={\setbox0=\lastbox
798 \endgraf \vskip-\parskip
799 \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\fi\fi
800 \let\par=\pend}%
801 \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.

```

802 \newcommand*{\normal@pars}{\everypar={}\let\par\endgraf}
803

```

## 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
804 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
805 \newcommand*{\do@line}{%
806 {\vbadness=10000
807 \splittopskip=\z@
808 \do@linehook
809 \l@emptyd@ta
810 \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
811 \unvbox\one@line \global\setbox\one@line=\lastbox
812 \getline@num
813 \ifnum\@lock>\@ne
814 \inserthangingsymboltrue
815 \else
816 \inserthangingsymbolfalse
817 \fi
818 \affixline@num
819 \affixpstart@num
820 \hb@xt@ \linewidth{\inserthangingsymbol\l@dld@ta\add@inserts\affixside@note
821 \l@dlsn@te
822 {\ledllfill\hb@xt@ \wd\one@line{\new@line\l@dunhbox@line{\one@line}}\ledrlfill\l@drd@
823 \l@drrs@te
824 }}}%

```

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

```

825 \newcommand*{\do@linehook}{}

\l@emptyd@ta Nulls the \. . .d@ta, which may later hold line numbers. Similarly for \l@dcstetext
\l@dld@ta for the text of a sidenote.
\l@drd@ta 826 \newcommand*{\l@emptyd@ta}{%
\l@dcstetext 827 \gdef\l@dld@ta{}%
 828 \gdef\l@drd@ta{}%
 829 \gdef\l@dcstetext{}}
 830

\l@dlsn@te Zero width boxes of the left and right side notes, together with their kerns.
\l@drsn@te 831 \newcommand{\l@dlsn@te}{%
 832 \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep}}
 833 \newcommand{\l@drsn@te}{%
 834 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
 835

\ledllfill These macros are called at the left (\ledllfill) and the right (\ledllfill) of each
\ledrlfill numbered line. The initial definitions correspond to the original code for \do@line.
 836 \newcommand*{\ledllfill}{\hfil}
 837 \newcommand*{\ledrlfill}{\hfil}
 838

```

### 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.

```

839 \newcommand*{\getline@num}{%
840 \ifnumberline
841 \global\advance\absline@num \@ne
842 \fi
843 \do@actions
844 \do@ballast
845 \ifnumberline
846 \ifsublines@
847 \ifnum\sub@lock<\tw@
848 \global\advance\subline@num \@ne
849 \fi
850 \else
851 \ifnum\@lock<\tw@
852 \global\advance\line@num \@ne
853 \global\subline@num \z@
854 \fi
855 \fi
856 \fi
857 }

```

`\do@ballast` The real work in the macro above is done in `\do@actions`, but before we plunge into that, let's get `\do@ballast` out of the way. This macro looks to see if there

is an action to be performed on the *next* line, and if it is going to be a page break action, `\do@ballast` decreases the count `\ballast@count` counter by the amount of `ballast`. This means, in practice, that when `\add@penalties` assigns penalties at this point, T<sub>E</sub>X 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.

```
858 \newcount\ballast@count
859 \newcounter{ballast}
860 \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.

```
861 \newcommand*{\do@ballast}{\global\ballast@count \z@
862 \begingroup
863 \advance\absline@num \@ne
864 \ifnum\next@actionline=\absline@num
865 \ifnum\next@action>-1001\relax
866 \global\advance\ballast@count by -\c@ballast
867 \fi
868 \fi
869 \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 T<sub>E</sub>X'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`.

```
870 \newcommand*{\do@actions}{%
871 \global\let\do@actions@next=\relax
872 \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.

```
873 \ifnum\next@action>-1001
874 \global\page@num=\next@action
875 \ifbypage@
876 \global\line@num=\z@ \global\subline@num=\z@
877 \fi
```

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`.)

```
878 \else
879 \ifnum\next@action<-4999
880 \@l@tempcnta=-\next@action
```

```

881 \advance\@l@dtmpcnta by -5001
882 \ifsublines@
883 \global\subline@num=\@l@dtmpcnta
884 \else
885 \global\line@num=\@l@dtmpcnta
886 \fi

```

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

```

887 \else
888 \@l@dtmpcnta=-\next@action
889 \advance\@l@dtmpcnta by -1000
890 \do@actions@fixedcode
891 \fi
892 \fi

```

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.

```

893 \ifx\actionlines@list\empty
894 \gdef\next@actionline{1000000}%
895 \else
896 \gl@p\actionlines@list\to\next@actionline
897 \gl@p\actions@list\to\next@action
898 \global\let\do@actions@next=\do@actions
899 \fi
900 \fi

```

Make the recursive call, if necessary.

```

901 \do@actions@next}
902

```

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

```

903 \newcommand*{\do@actions@fixedcode}{%
904 \ifcase\@l@dtmpcnta
905 \or% % 1001
906 \global\sublines@true
907 \or% % 1002
908 \global\sublines@false
909 \or% % 1003
910 \global\@lock=\@ne
911 \or% % 1004
912 \ifnum\@lock=\tw@
913 \global\@lock=\thr@@
914 \else
915 \global\@lock=\z@
916 \fi

```

```

917 \or% % 1005
918 \global\sub@lock=\@ne
919 \or% % 1006
920 \ifnum\sub@lock=\tw@
921 \global\sub@lock=\thr@@
922 \else
923 \global\sub@lock=\z@
924 \fi
925 \or% % 1007
926 \l@dskipnumbertrue
927 \else
928 \led@warn@BadAction
929 \fi}
930
931

```

## 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@dtempcnta`, 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@dtempcntb` for comparison.

Remember that some counts are now counters!

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

```

932 \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).
933 \ifnumberline
934 \ifl@dskipnumber
935 \global\l@dskipnumberfalse
936 \else
937 \ifsublines@
938 \l@dtempcntb=\subline@num

```



```

939 \ifnum\subline@num>\c@firstsublinenum
940 \l@dttempcnta=\subline@num
941 \advance\l@dttempcnta by-\c@firstsublinenum
942 \divide\l@dttempcnta by\c@sublinenumincrement
943 \multiply\l@dttempcnta by\c@sublinenumincrement
944 \advance\l@dttempcnta by\c@firstsublinenum
945 \else
946 \l@dttempcnta=\c@firstsublinenum
947 \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.

```

948 \ch@cksub@l@ck

```

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

```

949 \else
950 \l@dttempcntb=\line@num

```

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

```

951 \ifx\linenumberlist\empty
952 \ifnum\line@num>\c@firstlinenum
953 \l@dttempcnta=\line@num
954 \advance\l@dttempcnta by-\c@firstlinenum
955 \divide\l@dttempcnta by\c@linenumincrement
956 \multiply\l@dttempcnta by\c@linenumincrement
957 \advance\l@dttempcnta by\c@firstlinenum
958 \else
959 \l@dttempcnta=\c@firstlinenum
960 \fi
961 \else

```

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

```

962 \l@dttempcnta=\line@num
963 \edef\rem@inder{\linenumberlist,\number\line@num,}%
964 \edef\sc@n@list{\def\noexpand\sc@n@list
965 ###1,\number\l@dttempcnta,###2|{\def\noexpand\rem@inder{###2}}}%
966 \sc@n@list\expandafter\sc@n@list\rem@inder|
967 \ifx\rem@inder\empty\advance\l@dttempcnta\@ne\fi
968 \fi

```

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

```

969 \ch@ck@l@ck
970 \fi

```

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

```

971 \ifnum\l@dttempcnta=\l@dttempcntb

```

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 972 \if@twocolumn
 973 \if@firstcolumn
 974 \gdef\l@dld@ta{\llap{\leftlinenum}}}%
 975 \else
 976 \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
 977 \fi
 978 \else
```

Continuing the original code ...

```
979 \@l@tempcntb=\line@margin
980 \ifnum\@l@tempcntb>\@ne
981 \advance\@l@tempcntb \page@num
982 \fi
```

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

```
983 \ifodd\@l@tempcntb
984 \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
985 \else
986 \gdef\l@dld@ta{\llap{\leftlinenum}}}%
987 \fi
988 \fi
989 \else
```

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

```
990 %% #1%
991 \fi
```

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

```
992 \f@x@l@cks
993 \fi
994 \fi
995 }
996
```

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

```

997 \newcommand*{\ch@cksub@l@ck}{%
998 \ifcase\sub@lock
999 \or
1000 \ifnum\sublock@disp=\@ne
1001 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1002 \fi
1003 \or
1004 \ifnum\sublock@disp=\tw@ \else
1005 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1006 \fi
1007 \or
1008 \ifnum\sublock@disp=\z@
1009 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1010 \fi
1011 \fi}

```

Similarly for line numbers.

```

1012 \newcommand*{\ch@ck@l@ck}{%
1013 \ifcase\@lock
1014 \or
1015 \ifnum\lock@disp=\@ne
1016 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1017 \fi
1018 \or
1019 \ifnum\lock@disp=\tw@ \else
1020 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1021 \fi
1022 \or
1023 \ifnum\lock@disp=\z@
1024 \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1025 \fi
1026 \fi}

```

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

```

1027 \newcommand*{\f@x@l@cks}{%
1028 \ifcase\@lock
1029 \or
1030 \global\@lock=\tw@
1031 \or \or
1032 \global\@lock=\z@
1033 \fi
1034 \ifcase\sub@lock
1035 \or
1036 \global\sub@lock=\tw@
1037 \or \or
1038 \global\sub@lock=\z@
1039 \fi}
1040

```

\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.

```
1041 \newcommand{\pageparbreak}{\pend\newpage\pstart\noindent}
1042
```

## 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 1043
\ifsidepstartnum 1044 \newif\ifsidepstartnum
1045 \newcommand*{\affixpstart@num}{%
1046 \ifsidepstartnum
1047 \if@twocolumn
1048 \if@firstcolumn
1049 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
1050 \else
1051 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
1052 \fi
1053 \else
1054 \l@dttempcntb=\line@margin
1055 \ifnum\l@dttempcntb>\@ne
1056 \advance\l@dttempcntb \page@num
1057 \fi
1058 \ifodd\l@dttempcntb
1059 \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
1060 \else
1061 \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
1062 \fi
1063 \fi
1064 \fi
1065 }
1066 %
1067
1068
1069 \newif\ifpstartnum
```

```

1070 \pstartnumtrue
1071 \newcommand*{\leftpstartnum}{
1072 \ifpstartnum\thepstart
1073 \kern\linenumsep\fi
1074 \global\pstartnumfalse
1075 }
1076 \newcommand*{\rightpstartnum}{
1077 \ifpstartnum
1078 \kern\linenumsep
1079 \thepstart
1080 \fi
1081 \global\pstartnumfalse
1082 }

```

## 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.

```

1083 \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`.

```

1084 \newcommand*{\add@inserts}{%
1085 \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.

```

1086 \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.

```

1087 \ifx\next@insert\empty
1088 \ifx\insertlines@list\empty
1089 \global\noteschanged@true
1090 \gdef\next@insert{100000}%
1091 \else
1092 \gl@p\insertlines@list\to\next@insert
1093 \fi
1094 \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.

```

1095 \ifnum\next@insert=\absline@num
1096 \gl@p\inserts@list\to\@insert
1097 \@insert
1098 \global\let\@insert=\undefined
1099 \global\let\next@insert=\empty
1100 \global\let\add@inserts@next=\add@inserts
1101 \fi
1102 \fi

```

Make the recursive call, if necessary.

```

1103 \add@inserts@next}
1104

```

## 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. 77). Finally, the penalty is checked to see that it doesn't go below  $-10000$ .

```

1105 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
1106 \ifnum\num@lines>\@ne
1107 \global\advance\par@line \@ne
1108 \ifnum\par@line=\@ne
1109 \advance\@l@tempcnta \clubpenalty
1110 \fi
1111 \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
1112 \ifnum\@l@tempcntb=\num@lines
1113 \advance\@l@tempcnta \widowpenalty
1114 \fi
1115 \ifnum\par@line<\num@lines
1116 \advance\@l@tempcnta \interlinepenalty
1117 \fi
1118 \fi
1119 \ifnum\@l@tempcnta=\z@
1120 \relax
1121 \else
1122 \ifnum\@l@tempcnta>-10000
1123 \penalty\@l@tempcnta
1124 \else
1125 \penalty -10000

```

```

1126 \fi
1127 \fi}
1128

```

## 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 `TEX`, 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.

```

1129 \newcommand*{\flush@notes}{%
1130 \@xloop
1131 \ifx\inserts@list\empty \else
1132 \gl@p\inserts@list\to\@insert
1133 \@insert
1134 \global\let\@insert=\undefined
1135 \repeat}
1136

```

`\@xloop` `\@xloop` is a variant of the PLAIN `TEX` `\loop` macro, useful when it's hard to construct a positive test using the `TEX` `\if` commands—as in `\flush@notes` above. One 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 `TEX` `\loop` is used, too, so we could just call it `\loop`; but it seems preferable not to change the definitions of any of the standard macros.)

This variant of `\loop` was introduced by Alois KABELSCHACHT in *TUGboat* 8 (1987), pp. 184–5.

```

1137 \def\@xloop#1\repeat{%
1138 \def\body{#1\expandafter\body\fi}%
1139 \body}
1140

```

## 22 Footnotes

The footnote macros are adapted from those in PLAIN `TEX`, but they differ in these respects: the outer-level commands must add other commands to a list macro rather than doing insertions immediately; there are 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.

```
1141 \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`).

```
1142 \newcommand*{\notenumfont}{\normalfont}
```

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

`\select@@lemmafont` 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.

```
1143 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@@lemmafont#7|}%
1144 \def\select@@lemmafont#1/#2/#3/#4|{%
1145 {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
1146 \selectfont}%
1147
```

## 22.2 Outer-level footnote commands

`\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.

```
1148 \newcommand*{\Afootnote}[2][\relax]{%
1149 \newcommand{\content}{#2}%
1150 \forcsvlist{\listadd\options}{#1}% Made a list of options
```



```

1151 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1152 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1153 \ifnumberedpar@
1154 \xright@appenditem{\noexpand\vAfootnote{A}%
1155 {{\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}}\to\inserts@list
1156 \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.

```

1157 \else
1158 \vAfootnote{A}{{0|0|0|0|0|0|0|0}}{\csexpandonce{content}}}%
1159 \fi%
1160 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1161 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1162 \ignorespaces}

```

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

```

\Bfootnote \newcommand*{\Bfootnote}[2][\relax]{%
\Bfootnote \newcommand{\content}{#2}%
\Bfootnote \forcsvlist{\listadd\options}{#1}% Made a list of options
1166 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1167 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1168 \ifnumberedpar@
1169 \xright@appenditem{\noexpand\vBfootnote{B}%
1170 {{\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}}\to\inserts@list
1171 \global\advance\insert@count \@ne
1172 \else
1173 \vBfootnote{B}{{0|0|0|0|0|0|0|0}}{\csexpandonce{content}}}%
1174 \fi%
1175 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1176 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1177 \ignorespaces}

1178 \newcommand*{\Cfootnote}[2][\relax]{%
1179 \newcommand{\content}{#2}%
1180 \forcsvlist{\listadd\options}{#1}% Made a list of options
1181 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1182 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1183 \ifnumberedpar@
1184 \xright@appenditem{\noexpand\vCfootnote{C}%
1185 {{\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}}\to\inserts@list
1186 \global\advance\insert@count \@ne
1187 \else
1188 \vCfootnote{C}{{0|0|0|0|0|0|0|0}}{\csexpandonce{content}}}%
1189 \fi%
1190 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1191 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1192 \ignorespaces}

```

```

1193 \newcommand*{\Dfootnote}[2][\relax]{%

```

```
1238 \global\advance\insert@count \@ne
```

```

1239 \else
1240 \mpvAfootnote{A}{\{0|0|0|0|0|0|0\}}{\#1}}%
1241 \fi%
1242 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1243 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1244 \ignorespaces}
1245 \newcommand*\mpBfootnote}[2][\relax]{%
1246 \newcommand{\content}{#2}%
1247 \forcsvlist{\listadd\options}{#1}% Made a list of options
1248 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1249 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1250 \ifnumberedpar@
1251 \xright@appenditem{\noexpand\mpvBfootnote{B}%
1252 {\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}\to\inserts@list
1253 \global\advance\insert@count \@ne
1254 \else
1255 \mpvBfootnote{B}{\{0|0|0|0|0|0|0\}}{\#1}}%
1256 \fi%
1257 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1258 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1259 \ignorespaces}
1260 \newcommand*\mpCfootnote}[2][\relax]{%
1261 \newcommand{\content}{#2}%
1262 \forcsvlist{\listadd\options}{#1}% Made a list of options
1263 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1264 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1265 \ifnumberedpar@
1266 \xright@appenditem{\noexpand\mpvCfootnote{C}%
1267 {\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}\to\inserts@list
1268 \global\advance\insert@count \@ne
1269 \else
1270 \mpvCfootnote{C}{\{0|0|0|0|0|0|0\}}{\#1}}%
1271 \fi%
1272 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1273 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1274 \ignorespaces}
1275 \newcommand*\mpDfootnote}[2][\relax]{%
1276 \newcommand{\content}{#2}%
1277 \forcsvlist{\listadd\options}{#1}% Made a list of options
1278 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1279 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog
1280 \ifnumberedpar@
1281 \xright@appenditem{\noexpand\mpvDfootnote{D}%
1282 {\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}}}\to\inserts@list
1283 \global\advance\insert@count \@ne
1284 \else
1285 \mpvDfootnote{D}{\{0|0|0|0|0|0|0\}}{\#1}}%
1286 \fi%
1287 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\noexpand\tog

```

```

1288 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\
1289 \ignorespaces}

1290 \newcommand*{\mpEfootnote}[2][\relax]{%
1291 \newcommand{\content}{#2}%
1292 \forcsvlist{\listadd\options}{#1}% Made a list of options
1293 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\
1294 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\
1295 \ifnumberedpar@
1296 \xright@appenditem{\noexpand\mpvEfootnote{E}%
1297 {\l@d@nums}{\csexpandonce{tag}}{\csexpandonce{content}}}}\to\inserts
1298 \global\advance\insert@count \@ne
1299 \else
1300 \mpvEfootnote{E}{\{0|0|0|0|0|0|0\}}{#1}}%
1301 \fi%
1302 \ifinlist{nonum}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\
1303 \ifinlist{nosep}{\options}{\global\advance\insert@count \@ne\xright@appenditem{\global\
1304 \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.

```

1305 \newcommand*{\normalvfootnote}[2]{%
1306 \insert\csname #1footins\endcsname\bgroup
1307 \notefontsetup
1308 \footsplitskips
1309 \spaceskip=\z@skip \xspaceskip=\z@skip
1310 \csname #1footfmt\endcsname #2[#1]\egroup}

```

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

```

1311 \newcommand*{\footssplitskips}{%
1312 \interlinepenalty=\interfootnotelinepenalty
1313 \floatingpenalty=\@MM
1314 \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
1315 \leftskip=\z@skip \rightskip=\z@skip}
1316

```

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

```

1317 \newcommand*{\mpnormalvfootnote}[2]{%
1318 \global\setbox\@nameuse{mp#1footins}\vbox{%
1319 \unvbox\@nameuse{mp#1footins}
1320 \notefontsetup
1321 \hsize\columnwidth
1322 \@parboxrestore
1323 \color@begingroup
1324 \csname #1footfmt\endcsname #2[#1]\color@endgroup}}
1325

```

`\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).

```

1326 \newcommand*{\ledsetnormalparstuff}{%
1327 \normal@pars
1328 \parindent \z@ \parfillskip \z@ \@plus 1fil}
1329 \newcommandx*{\normalfootfmt}[4][4=Z]{% 4th arg is optional, for ascendant compatibility
1330 \ledsetnormalparstuff%
1331 \strut{\printlinefootnote{#1}{#4}}%
1332 {\select@lemmafnt#1|#2}%
1333 \iftoggle{nolemmaseparatorthisnote@}{\hskip\csuse{inplaceoflemmaseparator@#4}}{\ifcsemt{lemmasepa
1334 {\hskip\csuse{inplaceoflemmaseparator@#4}}%
1335 {\nobreak\hskip\csuse{beforelemmaseparator@#4}\csuse{lemmaseparator@#4}\hskip\csuse{afterlemmasep
1336 }}%
1337 #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.

`\fullstop`  
`\rbracket`

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.

```
1338 \def\endashchar{\textnormal{--}}
1339 \newcommand*\fullstop{\textnormal{.}}
1340 \newcommand*\rbracket{\textnormal{\thinspace]]}}
1341
```

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:

- `\@pnun` 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@pnun
\ifl@d@ssub 1342 \newif\ifl@d@pnun
\ifl@d@elin 1343 \l@d@pnunfalse
\ifl@d@esl 1344 \newif\ifl@d@ssub
\ifl@d@dash 1345 \l@d@ssubfalse
1346 \newif\ifl@d@elin
1347 \l@d@elinfalse
1348 \newif\ifl@d@esl
1349 \l@d@eslfalse
1350 \newif\ifl@d@dash
1351 \l@d@dashfalse

\l@dparsefootspec \l@dparsefootspec{<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@dparsestartpage page number and lemma font specifier in \l@d@nums style format. The real work
\l@dparsestartline is done by \l@dp@rsefootspec which defines macros holding the numeric values.
\l@dparsestartsub 1352 \newcommand*\l@dparsefootspec{[3]{\l@dp@rsefootspec#1}}
\l@dparseendpage
\l@dparseendline
\l@dparseendsub
```

```

1353 \def\l@dp@rsefootspec#1|#2|#3|#4|#5|#6|#7|{%
1354 \gdef\l@dparsedstartpage{#1}%
1355 \gdef\l@dparsedstartline{#2}%
1356 \gdef\l@dparsedstartsub{#3}%
1357 \gdef\l@dparsedendpage{#4}%
1358 \gdef\l@dparsedendline{#5}%
1359 \gdef\l@dparsedendsub{#6}%
1360 }

```

Initialise the several number value macros.

```

1361 \def\l@dparsedstartpage{0}%
1362 \def\l@dparsedstartline{0}%
1363 \def\l@dparsedstartsub{0}%
1364 \def\l@dparsedendpage{0}%
1365 \def\l@dparsedendline{0}%
1366 \def\l@dparsedendsub{0}%
1367

```

`\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`.

```

1368 \newcommand*\setprintlines}[6]{%
1369 \l@dpnumfalse \l@ddashfalse
1370 \ifbypage@
1371 \ifnum#4=#1 \else
1372 \l@dpnumtrue
1373 \l@ddashtrue
1374 \fi
1375 \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.

```

1376 \ifl@dpnum \l@d@elintrue \else \l@d@elinfalse \fi
1377 \ifnum#2=#5 \else
1378 \l@d@elintrue
1379 \l@ddashtrue
1380 \fi

```

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

```

1381 \l@d@ssubfalse
1382 \ifnum#3=0 \else
1383 \l@d@ssubtrue
1384 \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.

```

1385 \l@d@eslfalse
1386 \ifnum#6=0 \else
1387 \ifnum#6=#3
1388 \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
1389 \else
1390 \l@d@esltrue
1391 \l@d@dashtrue
1392 \fi
1393 \fi}

```

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

```

1394 \def\printlines#1|#2|#3|#4|#5|#6|#7|{\begingroup
1395 \ifbypstart%
1396 \ifl@d@pairing%
1397 \ifledRcol%
1398 \thepstartR%
1399 \else%
1400 \thepstartL%
1401 \fi%
1402 \else%
1403 \thepstart%
1404 \fi%
1405 \fi%
1406 \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).

```

1407 \ifl@d@pnum #1\fullstop\fi
1408 \linenumrep{#2}

1409 \ifl@d@ssub \fullstop \sublinenumrep{#3}\fi
1410 \ifl@d@dash \endashchar\fi
1411 \ifl@d@pnum #4\fullstop\fi
1412 \ifl@d@elin \linenumrep{#5}\fi
1413 \ifl@d@esl \ifl@d@elin \fullstop\fi \sublinenumrep{#6}\fi
1414 \endgroup}
1415 % \begin{macro}{\normalfootstart}
1416 % \phantomsection\label{normalvalues}\relax
1417 % \cs{normalfootstart}
1418 % is a standard footnote-starting macro, called in the output
1419 % routine whenever there are footnotes of this series to be
1420 % printed: it skips a bit and then draws a rule.
1421 %
1422 % Any \verb"footstart" macro must put onto the page something that
1423 % takes up space exactly equal to the \verb"\skip\footins" value for
1424 % the associated series of notes. \TeX\ makes page computations based
1425 % on that \cs{skip} value, and the output pages will suffer from
1426 % spacing problems if what you add takes up a different amount of

```



```

1427 % space.
1428 %
1429 % The \cs{leftskip} and \cs{rightskip} values
1430 % are both zeroed here.\phantomsection\label{footskips}
1431 % Similarly, these skips are cancelled in the \verb"vfootnote" macros
1432 % for the various types of notes. Strictly speaking,
1433 % this is necessary only
1434 % if you are using paragraphed footnotes, but we have put it here
1435 % and in the other \verb"vfootnote" macros too so that the behavior of
1436 % \Ledmac{} in this respect is general across all footnote types
1437 % (you can change this). What this means is that any \cs{leftskip}
1438 % and \cs{rightskip} you specify applies to the main text, but not
1439 % the footnotes. The footnotes continue to be of width \cs{hsize}.
1440 % \begin{macrocode}
1441 \newcommand*{\normalfootstart}[1]{%
1442 \vskip\skip\csname #1footins\endcsname
1443 \leftskip0pt \rightskip0pt
1444 \csname #1footnoterule\endcsname}

\normalfootnoterule \norrmalfootnoterule is a standard footnote-rule macro, for use by a footstart
macro: just the same as the PLAIN TEX footnote rule.
1445 \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.
1446 \newcommand*{\normalfootgroup}[1]{\unvbox\csname #1footins\endcsname}
1447

\mpnormalfootgroup A somewhat different version for minipages.
1448 \newcommand*{\mpnormalfootgroup}[1]{%
1449 \vskip\skip\@nameuse{mp#1footins}
1450 \normalcolor
1451 \@nameuse{#1footnoterule}
1452 \unvbox\csname mp#1footins\endcsname}%
1453

```

## 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`

```
1454 \newcommand*{\ledfootinsdim}{0.8\vsiz}
1455
```

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

```
1456 \newinsert\Afootins \newinsert\Bfootins
1457 \newinsert\Cfootins \newinsert\Dfootins
1458 \newinsert\Efootins
```

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

```
1459 \newcommand*{\footnormal}[1]{%
1460 \expandafter\let\csname #1footstart\endcsname=\normalfootstart
1461 \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
1462 \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
1463 \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
1464 \expandafter\let\csname #1footnoterule\endcsname=%
1465 \normalfootnoterule
1466 \count\csname #1footins\endcsname=1000
1467 \dimen\csname #1footins\endcsname=\ledfootinsdim
1468 \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).

```
1469 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
1470 \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
1471 \count\csname mp#1footins\endcsname=1000
1472 \dimen\csname mp#1footins\endcsname=\ledfootinsdim
1473 \skip\csname mp#1footins\endcsname=1.2em \@plus .6em \@minus .6em
1474 }
1475
```

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  $\TeX$  to make.

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

```
1476 \footnormal{A}
1477 \footnormal{B}
1478 \footnormal{C}
1479 \footnormal{D}
1480 \footnormal{E}
1481
```

## 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  $\TeX$  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  $\TeX$  will try to put too many or too few of these notes on each page. If you need to change the `\hsize` within the document, call `\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.

```
1482 \newcommand*{\footparagraph}[1]{%
1483 \expandafter\newcount\csname prevpage#1@num\endcsname
1484 \expandafter\let\csname #1footstart\endcsname=\parafootstart
1485 \expandafter\let\csname v#1footnote\endcsname=\para@vfootnote
1486 \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
1487 \expandafter\let\csname #1footgroup\endcsname=\para@footgroup
1488 \count\csname #1footins\endcsname=1000
1489 \para@footsetup{#1}
```

And the extra setup for minipages.

```
1490 \expandafter\let\csname mpv#1footnote\endcsname=\mppara@vfootnote
1491 \expandafter\let\csname mp#1footgroup\endcsname=\mppara@footgroup
1492 \count\csname mp#1footins\endcsname=1000
1493 }
```

`\footfudgefiddle` For paraphrased footnotes  $\TeX$  has to estimate the amount of space required. If it underestimates this then the notes may get too long and run off the bottom of the

text block. `\footfudgefiddle` can be increased from its default 64 (say to 70) to increase the estimate.

```
1494 \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 9pt) 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`.

```
1495 \newcommand*{\para@footsetup}[1]{\notefontsetup
1496 \dimen0=\baselineskip
1497 \multiply\dimen0 by 1024
1498 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
1499 \expandafter
1500 \xdef\csname #1footfudgefactor\endcsname{%
1501 \expandafter\strip@pt\dimen0 }}
1502
```

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.

```
1503 \newcommand*{\parafootstart}[1]{%
1504 \rightskip=0pt \leftskip=0pt \parindent=0pt
1505 \vskip\skip\csname #1footins\endcsname
1506 \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 T<sub>E</sub>X does not expect to have to break lines, it does not insert certain items

like `\discretionary`s. If you later unbox these hboxes and stick them together, as the *TeXbook* macros do to make these footnotes, you lose the ability to hyphenate after an explicit hyphen. This can lead to overfull `\hboxes` when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.<sup>22</sup>

Wayne Sullivan pointed out to us another subtle problem that arises from the same cause: `TeX` also leaves the `\language` whatsit nodes out of the horizontal list.<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 `\hbox` in the first place, but instead to collect it in a `\vbox` whose width is (virtually) infinite. The text is therefore typeset in unrestricted horizontal mode, as a paragraph consisting of a single long line. Later, there is an extra level of unboxing to be done: we have to unpack the `\vbox`, as well as the hboxes inside it, but that'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 `\unvxh`: `unvbox`, extract the last line, and `unhbox` it.

Doing things this way has an important consequence: as Michael pointed out, you really can't put an explicit line-break into a note built in a `\vbox` the way we are doing.<sup>25</sup> In other words, be very careful not to say `\break`, or `\penalty-10000`, or any equivalent inside your para-footnote. If you do, most of the note will probably disappear. You *are* allowed to make strong suggestions; in fact `\penalty-9999` will be quite okay. Just 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 `ledmac` is quite baroque enough already. If you think you are having this problem, look up Michael's solutions.

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

So, finally, here is the modified foot-paragraph code, which sets the footnote in vertical mode so that language and discretionary nodes are included.

```
1507 \newcommand*{\para@vfootnote}[2]{%
1508 \insert\csname #1footins\endcsname
1509 \bgroup
1510 \notefontsetup
1511 \footplitskips
1512 \setbox0=\vbox{\hsize=\maxdimen
```

<sup>22</sup>Michael Downes, 'Line Breaking in `\unboxed` 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 `\unvxh` macro since it is publicly documented.

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

```

1513 \noindent\csname #1footfmt\endcsname#2[#1]]%
1514 \setbox0=\hbox{\unvxh0[#1]]}%
1515 \dp0=0pt
1516 \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.

```

1517 \box0
1518 \penalty0
1519 \egroup}
1520

```

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.

```

1521 \newcommand*{\mppara@vfootnote}[2]{%
1522 \global\setbox\@nameuse{mp#1footins}\vbox{%
1523 \unvbox\@nameuse{mp#1footins}%
1524 \notefontsetup
1525 \footplitskips
1526 \setbox0=\vbox{\hsize=\maxdimen
1527 \noindent\color@begingroup\csname #1footfmt\endcsname #2[#1]\color@endgroup}%
1528 \setbox0=\hbox{\unvxh0[#1]]}%
1529 \dp0=\z@
1530 \ht0=\csname #1footfudgefactor\endcsname\wd0
1531 \box0
1532 \penalty0
1533 }}
1534

```

`\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`.

```

1535 \newcommand*{\unvxh}[2][2=Z]{% 2th is optional for retro-compatibility
1536 \setbox0=\vbox{\unvbox#1%
1537 \global\setbox1=\lastbox}%

```

```

1538 \unhbox1
1539 \unskip % remove \rightskip,
1540 \unskip % remove \parfillskip,
1541 \unpenalty % remove \penalty of 10000,
1542 \hskip\csuse{afternote@#2}} % but add the glue to go between the notes
1543

```

`\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).

```

1544 \newcommand*{\parafootfmt}[4][4=Z]{%
1545 \insertparafootsep{#4}%
1546 \ledsetnormalparstuff%
1547 \printlinefootnote{#1}{#4}%
1548 {\select@lemmafont#1|#2}%
1549 \iftoggle{nolemmaseparatorthisnote@}{\hskip\csuse{inplaceoflemmaseparator@#4}}{\ifcsemtyp{lemmasepa
1550 {\hskip\csuse{inplaceoflemmaseparator@#4}}}%
1551 {\nobreak\hskip\csuse{beforelemmaseparator@#4}\csuse{lemmaseparator@#4}\hskip\csuse{afterlemmasep
1552 }}%
1553 #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.

```

1554 \newcommand*{\para@footgroup}[1]{%
1555 \unvbox\csname #1footins\endcsname
1556 \makehboxofhboxes
1557 \setbox0=\hbox{\unhbox0 \removehboxes}%
1558 \notefontsetup
1559 \noindent\unhbox0\par}
1560

```

`\mppara@footgroup` The minipage version.

```

1561 \newcommand*{\mppara@footgroup}[1]{%
1562 \vskip\skip\@nameuse{mp#1footins}
1563 \normalcolor
1564 \@nameuse{#1footnoterule}%
1565 \unvbox\csname mp#1footins\endcsname
1566 \makehboxofhboxes
1567 \setbox0=\hbox{\unhbox0 \removehboxes}%

```

```

1568 \notefontsetup
1569 \noindent\unhbox0\par}}
1570

```

```

\makehboxofhboxes
\removehboxes 1571 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
1572 \loop
1573 \unpenalty
1574 \setbox2=\lastbox
1575 \ifhbox2
1576 \setbox0=\hbox{\box2\unhbox0}%
1577 \repeat}
1578
1579 \newcommand*{\removehboxes}{\setbox0=\lastbox
1580 \ifhbox0{\removehboxes}\unhbox0 \fi}
1581

```

### 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 1582 \newcommand{\insertparafootsep}[1]{%
1583 \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
1584 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
1585 {\ifnumequal{\csuse{prevline#1}}{\line@num}%
1586 {\ifcsempy{sympmlinenum}{\csuse{parafootsep@#1}}{}}}%
1587 {\csuse{parafootsep@#1}}}%
1588 }%
1589 {\csuse{parafootsep@#1}}}%
1590 }%
1591 {}%
1592 \global\csname prevpage#1@num\endcsname=\page@num%
1593 }

```

## 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`.

```

1594 \newcount\@k \newdimen\@h

```



```

1595 \newcommand*{\rigidbalance}[3]{\setbox0=\box#1 \@k=#2 \@h=#3
1596 \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
1597 \valign{##\vfil\cr\dosplits}}}}
1598
1599 \newcommand*{\dosplits}{\ifnum\@k>0 \noalign{\hfil}\splitoff
1600 \global\advance\@k-1\cr\dosplits\fi}
1601
1602 \newcommand*{\splitoff}{\dimen0=\ht0
1603 \divide\dimen0 by\@k \advance\dimen0 by\@h
1604 \setbox2 \vsplit0 to \dimen0
1605 \unvbox2 }
1606

```

### 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.

```

1607 \newcommand*{\footthreecol}[1]{%
1608 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
1609 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
1610 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
1611 \threecolfootsetup{#1}

```

The additional setup for minipages.

```

1612 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
1613 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
1614 \mpthreecolfootsetup{#1}
1615 }
1616

```

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 T<sub>E</sub>X is accumulating material for the page and checking that limit, it doesn't apply the `\count` scaling.

```

1617 \newcommand*{\threecolfootsetup}[1]{%
1618 \count\csname #1footins\endcsname 333

```

```
1619 \multiply\dimen\csname #1footins\endcsname \thr@@}
```

`\mpthreecolfootsetup` The setup for minipages.

```
1620 \newcommand*{\mpthreecolfootsetup}[1]{%
1621 \count\csname mp#1footins\endcsname 333
1622 \multiply\dimen\csname mp#1footins\endcsname \thr@@}
1623
```

`\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).

```
1624 \newcommand*{\threecolvfootnote}[2]{%
1625 \insert\csname #1footins\endcsname\bgroup
1626 \notefontsetup
1627 \footsplitskips
1628 \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.

```
1629 \newcommandx*{\threecolfootfmt}[4][4=Z]{%
1630 \normal@pars
1631 \hsize .3\hsize
1632 \parindent=0pt
1633 \tolerance=5000
1634 \raggedright
1635 \leavevmode
1636 \strut{\printlinefootnote{#1}{#4}}%
1637 {\select@lemmafont#1|#2}%
1638 \iftoggle{nolemmaseparatorthisnote@}{\hskip\csuse{inplaceoflemmaseparator@#4}}{\ifcsemp
1639 {\hskip\csuse{inplaceoflemmaseparator@#4}}}%
1640 {\nobreak\hskip\csuse{beforelemmaseparator@#4}\csuse{lemmaseparator@#4}\hskip\csuse{a
1641 }}%
1642 #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.

```
1643 \newcommand*\threecolfootgroup}[1]{\notefontsetup
1644 \splittopskip=\ht\strutbox
1645 \expandafter
1646 \rigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}}
```

`\mpthreecolfootgroup` The setup for minipages.

```
1647 \newcommand*\mpthreecolfootgroup}[1]{%
1648 \vskip\skip\@nameuse{mp#1footins}
1649 \normalcolor
1650 \@nameuse{#1footnoterule}
1651 \splittopskip=\ht\strutbox
1652 \expandafter
1653 \rigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
1654
```

## 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.

```
1655 \newcommand*\foottwocol}[1]{%
1656 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
1657 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
1658 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
1659 \twocolfootsetup{#1}
```

The additional setup for minipages.

```
1660 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
1661 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
1662 \mptwocolfootsetup{#1}
1663 }
1664
```

`\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`.

```
1665 \newcommand*\twocolfootsetup}[1]{%
1666 \count\csname #1footins\endcsname 500
1667 \multiply\dimen\csname #1footins\endcsname \tw@}

1668 \newcommand*\twocolvfootnote}[2]{\insert\csname #1footins\endcsname\bgroup
1669 \notefontsetup
1670 \footsplitskips
1671 \csname #1footfmt\endcsname #2[#1]\egroup}
```

```

1672 \newcommandx*{\twocolfootfmt}[4][4=Z]{% 4th arg is optional, for ascendant compatibility
1673 \normal@pars
1674 \hsize .45\hsize
1675 \parindent=0pt
1676 \tolerance=5000
1677 \raggedright
1678 \leavevmode
1679 \strut{\printlinefootnote{#1}{#4}}%
1680 {\select@lemmafont#1|#2}%
1681 \iftoggle{nolemmaseparatorthisnote@}{\hskip\csuse{inplaceoflemmaseparator@#4}}{\ifcsemp
1682 {\hskip\csuse{inplaceoflemmaseparator@#4}}%
1683 {\nobreak\hskip\csuse{beforelemmaseparator@#4}\csuse{lemmaseparator@#4}\hskip\csuse{a
1684 }}%
1685 #3\strut\par\allowbreak}

1686 \newcommand*{\twocolfootgroup}[1]{\notefontsetup
1687 \splittopskip=\ht\strutbox
1688 \expandafter
1689 \rigidbalance\csname #1footins\endcsname \tw@ \splittopskip}}
1690

```

`\mptwocolfootsetup` The versions for minipages.

```

\mptwocolfootgroup 1691 \newcommand*{\mptwocolfootsetup}[1]{%
1692 \count\csname mp#1footins\endcsname 500
1693 \multiply\dimen\csname mp#1footins\endcsname \tw@}

1694 \newcommand*{\mptwocolfootgroup}[1]{%
1695 \vskip\skip\@nameuse{mp#1footins}
1696 \normalcolor
1697 \@nameuse{#1footnoterule}
1698 \splittopskip=\ht\strutbox
1699 \expandafter
1700 \rigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
1701

```

## 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`

```

1702 \newtoggle{numberonlyfirstinline@A}
1703 \newtoggle{numberonlyfirstinline@B}
1704 \newtoggle{numberonlyfirstinline@C}
1705 \newtoggle{numberonlyfirstinline@D}
1706 \newtoggle{numberonlyfirstinline@E}
1707 \newtoggle{numberonlyfirstinline@Z}
1708
1709 \newcommandx{\numberonlyfirstinline}[2][1,2={true}]{%

```

```

1710 \ifstrempy{#1}{
1711 \settoggle{numberonlyfirstinline@A}{#2}
1712 \settoggle{numberonlyfirstinline@B}{#2}
1713 \settoggle{numberonlyfirstinline@C}{#2}
1714 \settoggle{numberonlyfirstinline@D}{#2}
1715 \settoggle{numberonlyfirstinline@E}{#2}
1716 }
1717 {
1718 \settoggle{numberonlyfirstinline@#1}{#2}
1719 }
1720 }

```

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

```

1721 \newif\ifledplinenum
1722 \ledplinenumtrue
1723 \newcommand*{\symlinenum}{}

```

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

```

1724 \def\symlinenum@A{\symlinenum}
1725 \def\symlinenum@B{\symlinenum}
1726 \def\symlinenum@C{\symlinenum}
1727 \def\symlinenum@D{\symlinenum}
1728 \def\symlinenum@E{\symlinenum}
1729 \def\symlinenum@Z{\symlinenum}
1730 \newcommandx{\symlinenum}[2][1]{%
1731 \ifstrequal{#1}{}% If arg 1 is empty
1732 {%
1733 \csdef{symlinenum@A}{#2}%
1734 \csdef{symlinenum@B}{#2}%
1735 \csdef{symlinenum@C}{#2}%
1736 \csdef{symlinenum@D}{#2}%
1737 \csdef{symlinenum@E}{#2}%
1738 \csdef{symlinenum@Z}{#2}%
1739 }%
1740 {%
1741 \csdef{symlinenum@#1}{#2}%
1742 }
1743 }

```

```

1744 % \end{macrocode}

```

```

1745 % \end{macro}

```

```

1746 % \begin{macro}{\nonumberinfootnote}

```

```

1747 % \cs{nonumberinfootnote}\oarg{series}\oarg{position} is used to disable line number printing in foot

```

```

1748 % \begin{macrocode}

```

```

1749 \newtoggle{nonumberinfootnote@A}

```

```

1750 \newtoggle{nonumberinfootnote@B}

```

```

1751 \newtoggle{nonumberinfootnote@C}

```

```

1752 \newtoggle{nonumberinfootnote@D}

```

```

1753 \newtoggle{nonumberinfootnote@E}
1754 \newtoggle{nonumberinfootnote@Z}
1755 \newcommandx{\nonumberinfootnote}[2][1,2={true}]{%
1756 \ifstrepty{#1}{
1757 \settoggle{nonumberinfootnote@A}{#2}
1758 \settoggle{nonumberinfootnote@B}{#2}
1759 \settoggle{nonumberinfootnote@C}{#2}
1760 \settoggle{nonumberinfootnote@D}{#2}
1761 \settoggle{nonumberinfootnote@E}{#2}
1762 }
1763 {
1764 \settoggle{nonumberinfootnote@#1}{#2}
1765 }
1766 }

```

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

```
1767 \newtoggle{nonumberthisnote@}
```

`\nolemmaseparatorthisnote@` `\nonumberthisnote@` toggle is used to disable lemma separator in a particular footnote.

```
1768 \newtoggle{nolemmaseparatorthisnote@}
```

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

```

1769 \def\beforenumberinfootnote@A{0pt}
1770 \def\beforenumberinfootnote@B{0pt}
1771 \def\beforenumberinfootnote@C{0pt}
1772 \def\beforenumberinfootnote@D{0pt}
1773 \def\beforenumberinfootnote@E{0pt}
1774 \def\beforenumberinfootnote@Z{0pt}
1775 \newcommandx*\beforenumberinfootnote[2][1,usedefault]{%
1776 \ifstrequal{#1}{}% If arg 1 is empty
1777 {%
1778 \csdef{beforenumberinfootnote@A}{#2}%
1779 \csdef{beforenumberinfootnote@B}{#2}%
1780 \csdef{beforenumberinfootnote@C}{#2}%
1781 \csdef{beforenumberinfootnote@D}{#2}%
1782 \csdef{beforenumberinfootnote@E}{#2}%
1783 \csdef{beforenumberinfootnote@Z}{#2}%
1784 }%
1785 {%
1786 \csdef{beforenumberinfootnote@#1}{#2}%
1787 }%
1788 }

```

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

```
1789 \def\afternumberinfootnote@A{0.5em}
```

```

1790 \def\afternumberinfootnote@B{0.5em}
1791 \def\afternumberinfootnote@C{0.5em}
1792 \def\afternumberinfootnote@D{0.5em}
1793 \def\afternumberinfootnote@E{0.5em}
1794 \def\afternumberinfootnote@Z{0.5em}
1795 \newcommandx*\afternumberinfootnote}[2][1,usedefault]{%
1796 \ifstrequal{#1}{}% If arg 1 is empty
1797 {%
1798 \csdef{afternumberinfootnote@A}{#2}%
1799 \csdef{afternumberinfootnote@B}{#2}%
1800 \csdef{afternumberinfootnote@C}{#2}%
1801 \csdef{afternumberinfootnote@D}{#2}%
1802 \csdef{afternumberinfootnote@E}{#2}%
1803 \csdef{afternumberinfootnote@Z}{#2}%
1804 }%
1805 {%
1806 \csdef{afternumberinfootnote@#1}{#2}%
1807 }%
1808 }

```

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

```

1809 \def\beforesymmlinenum@A{\beforenumberinfootnote@A}
1810 \def\beforesymmlinenum@B{\beforenumberinfootnote@B}
1811 \def\beforesymmlinenum@C{\beforenumberinfootnote@C}
1812 \def\beforesymmlinenum@D{\beforenumberinfootnote@D}
1813 \def\beforesymmlinenum@E{\beforenumberinfootnote@E}
1814 \def\beforesymmlinenum@Z{\beforenumberinfootnote@Z}
1815 \newcommandx*\beforesymmlinenum}[2][1,usedefault]{%
1816 \ifstrequal{#1}{}% If arg 1 is empty
1817 {%
1818 \csdef{beforesymmlinenum@A}{#2}%
1819 \csdef{beforesymmlinenum@B}{#2}%
1820 \csdef{beforesymmlinenum@C}{#2}%
1821 \csdef{beforesymmlinenum@D}{#2}%
1822 \csdef{beforesymmlinenum@E}{#2}%
1823 \csdef{beforesymmlinenum@Z}{#2}%
1824 }%
1825 {%
1826 \csdef{beforesymmlinenum@#1}{#2}%
1827 }%
1828 }

```

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

```

1829 \def\aftersymmlinenum@A{\afternumberinfootnote@A}
1830 \def\aftersymmlinenum@B{\afternumberinfootnote@B}
1831 \def\aftersymmlinenum@C{\afternumberinfootnote@C}
1832 \def\aftersymmlinenum@D{\afternumberinfootnote@D}
1833 \def\aftersymmlinenum@E{\afternumberinfootnote@E}
1834 \def\aftersymmlinenum@Z{\afternumberinfootnote@Z}
1835 \newcommandx*\aftersymmlinenum}[2][1,usedefault]{%

```

```

1836 \ifstrequal{#1}{}% If arg 1 is empty
1837 {%
1838 \csdef{aftersymmlinenumber@A}{#2}%
1839 \csdef{aftersymmlinenumber@B}{#2}%
1840 \csdef{aftersymmlinenumber@C}{#2}%
1841 \csdef{aftersymmlinenumber@D}{#2}%
1842 \csdef{aftersymmlinenumber@E}{#2}%
1843 \csdef{aftersymmlinenumber@Z}{#2}%
1844 }%
1845 {%
1846 \csdef{aftersymmlinenumber@#1}{#2}%
1847 }%
1848 }

```

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

```

1849 \def\inplaceofnumber@A{1em}
1850 \def\inplaceofnumber@B{1em}
1851 \def\inplaceofnumber@C{1em}
1852 \def\inplaceofnumber@D{1em}
1853 \def\inplaceofnumber@E{1em}
1854 \def\inplaceofnumber@Z{1em}
1855 \newcommandx*{\inplaceofnumber}[2][1,usedefault]{%
1856 \ifstrequal{#1}{}% If arg 1 is empty
1857 {%
1858 \csdef{inplaceofnumber@A}{#2}%
1859 \csdef{inplaceofnumber@B}{#2}%
1860 \csdef{inplaceofnumber@C}{#2}%
1861 \csdef{inplaceofnumber@D}{#2}%
1862 \csdef{inplaceofnumber@E}{#2}%
1863 \csdef{inplaceofnumber@Z}{#2}%
1864 }%
1865 {%
1866 \csdef{inplaceofnumber@#1}{#2}%
1867 }%
1868 }

```

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

```

1869 \let\lemmaseparator@A\rbracket
1870 \let\lemmaseparator@B\rbracket
1871 \let\lemmaseparator@C\rbracket
1872 \let\lemmaseparator@D\rbracket
1873 \let\lemmaseparator@E\rbracket
1874 \let\lemmaseparator@F\rbracket
1875 \let\lemmaseparator@Z\rbracket
1876 \newcommandx*{\lemmaseparator}[2][1,usedefault]{%
1877 \ifstrequal{#1}{}% If arg 1 is empty
1878 {%
1879 \csdef{lemmaseparator@A}{#2}%
1880 \csdef{lemmaseparator@B}{#2}%

```



```

1881 \csdef{lemmaseparator@C}{#2}%
1882 \csdef{lemmaseparator@D}{#2}%
1883 \csdef{lemmaseparator@E}{#2}%
1884 \csdef{lemmaseparator@Z}{#2}%
1885 }%
1886 {%
1887 \csdef{lemmaseparator@#1}{#2}%
1888 }%
1889 }

```

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

```

1890 \def\beforelemmaseparator@A{0em}
1891 \def\beforelemmaseparator@B{0em}
1892 \def\beforelemmaseparator@C{0em}
1893 \def\beforelemmaseparator@D{0em}
1894 \def\beforelemmaseparator@E{0em}
1895 \def\beforelemmaseparator@Z{0em}
1896 \newcommandx*{\beforelemmaseparator}[2][1,usedefault]{%
1897 \ifstrequal{#1}{}% If arg 1 is empty
1898 {%
1899 \csdef{beforelemmaseparator@A}{#2}%
1900 \csdef{beforelemmaseparator@B}{#2}%
1901 \csdef{beforelemmaseparator@C}{#2}%
1902 \csdef{beforelemmaseparator@D}{#2}%
1903 \csdef{beforelemmaseparator@E}{#2}%
1904 \csdef{beforelemmaseparator@Z}{#2}%
1905 }%
1906 {%
1907 \csdef{beforelemmaseparator@#1}{#2}%
1908 }%
1909 }

```

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

```

1910 \def\afterlemmaseparator@A{0.5em}
1911 \def\afterlemmaseparator@B{0.5em}
1912 \def\afterlemmaseparator@C{0.5em}
1913 \def\afterlemmaseparator@D{0.5em}
1914 \def\afterlemmaseparator@E{0.5em}
1915 \def\afterlemmaseparator@Z{0.5em}
1916 \newcommandx*{\afterlemmaseparator}[2][1,usedefault]{%
1917 \ifstrequal{#1}{}% If arg 1 is empty
1918 {%
1919 \csdef{afterlemmaseparator@A}{#2}%
1920 \csdef{afterlemmaseparator@B}{#2}%
1921 \csdef{afterlemmaseparator@C}{#2}%
1922 \csdef{afterlemmaseparator@D}{#2}%
1923 \csdef{afterlemmaseparator@E}{#2}%

```

```

1924 \csdef{afterlemmaseparator@Z}{#2}%
1925 }%
1926 {%
1927 \csdef{afterlemmaseparator@#1}{#2}%
1928 }%
1929 }

```

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

```

1930 \newcommand*{\nolemmaseparator}[1][1]{\lemmaseparator[#1]}
1931 % \end{macrocode}
1932 % \end{macro}
1933 % \begin{macro}{\inplaceofnumber}
1934 % \cs{inplaceoflemmaseparator}\oarg{series}\marg{value} is used to define glue which is p
1935 % \begin{macrocode}
1936 \def\inplaceoflemmaseparator@A{1em}
1937 \def\inplaceoflemmaseparator@B{1em}
1938 \def\inplaceoflemmaseparator@C{1em}
1939 \def\inplaceoflemmaseparator@D{1em}
1940 \def\inplaceoflemmaseparator@E{1em}
1941 \def\inplaceoflemmaseparator@Z{1em}
1942 \newcommand*{\inplaceoflemmaseparator}[2][1,usedefault]{%
1943 \ifstrequal{#1}{}% If arg 1 is empty
1944 {%
1945 \csdef{inplaceoflemmaseparator@A}{#2}%
1946 \csdef{inplaceoflemmaseparator@B}{#2}%
1947 \csdef{inplaceoflemmaseparator@C}{#2}%
1948 \csdef{inplaceoflemmaseparator@D}{#2}%
1949 \csdef{inplaceoflemmaseparator@E}{#2}%
1950 \csdef{inplaceoflemmaseparator@Z}{#2}%
1951 }%
1952 {%
1953 \csdef{inplaceoflemmaseparator@#1}{#2}%
1954 }%
1955 }

```

`\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,  $\text{\TeX}$  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.

```

1956 \def\afternote@A{1em plus.4em minus.4em}
1957 \def\afternote@B{1em plus.4em minus.4em}
1958 \def\afternote@C{1em plus.4em minus.4em}
1959 \def\afternote@D{1em plus.4em minus.4em}
1960 \def\afternote@E{1em plus.4em minus.4em}
1961 \def\afternote@Z{1em plus.4em minus.4em}
1962 \newcommand*{\afternote}[2][1,usedefault]{%
1963 \ifstrequal{#1}{}% If arg 1 is empty
1964 {%
1965 \csdef{afternote@A}{#2}%
1966 \csdef{afternote@B}{#2}%
1967 \csdef{afternote@C}{#2}%
1968 \csdef{afternote@D}{#2}%
1969 \csdef{afternote@E}{#2}%
1970 \csdef{afternote@Z}{#2}%
1971 }%
1972 {%
1973 \csdef{afternote@#1}{#2}%
1974 }%
1975 }
1976 % \end{macrocode}
1977 % \end{macro}
1978 % \begin{macro}{\interparanoteglue}
1979 % \begin{macro}{\ipn@skip}
1980 % The \cs{ipn@skip} skip and \cs{interparanoteglue} command are kept for ascendant compatibility, but
1981 % \begin{macrocode}
1982 \newskip\ipn@skip
1983 \newcommand*{\interparanoteglue}[1]{%
1984 {\notefontsetup\global\ipn@skip=#1 \relax}}
1985 \interparanoteglue{1em plus.4em minus.4em}

```

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

```
1986 \newcommand{\parafootftmsep}{}

```

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

```

1987 \def\parafootsep@A{\parafootftmsep}
1988 \def\parafootsep@B{\parafootftmsep}
1989 \def\parafootsep@C{\parafootftmsep}
1990 \def\parafootsep@D{\parafootftmsep}

```

```

1991 \def\parafootsep@E{\parafootftmsep}
1992 \def\parafootsep@Z{\parafootftmsep}
1993 \newcommandx*\parafootsep}[2][1,usedefault]{%
1994 \ifstrequal{#1}{}% If arg 1 is empty
1995 {%
1996 \csdef{parafootsep@A}{#2}%
1997 \csdef{parafootsep@B}{#2}%
1998 \csdef{parafootsep@C}{#2}%
1999 \csdef{parafootsep@D}{#2}%
2000 \csdef{parafootsep@E}{#2}%
2001 \csdef{parafootsep@Z}{#2}%
2002 }%
2003 {%
2004 \csdef{parafootsep@#1}{#2}%
2005 }%
2006 }
2007 % \end{macrocode}
2008 \end{macro}
2009 \subsubsection{Line number printing}
2010 \begin{macro}{\printlinefootnote}
2011 % The \cs{printlinefootnote} macro is called in each \cs{<type>footfmt} command. It prints
2012 % \begin{macrocode}
2013 \newcommand{\printlinefootnote}[2]{%
2014 \iftoggle{nonumberthisnote@}{%Try if the line number must printed for this specific not
2015 \hspace{\csuse{inplaceofnumber@#2}}%
2016 }%
2017 {%
2018 {%
2019 \iftoggle{nonumberinfootnote@#2}%Try if the line number must printed (by default
2020 {%
2021 \hspace{\csuse{inplaceofnumber@#2}}%
2022 }%
2023 {%
2024 {\iftoggle{numberonlyfirstinline@#2}% If for this series the line number must
2025 {%
2026 \ifcsdef{prevline#2}%
2027 {%Be sure the \prevline exists.
2028 \ifnumequal{\csuse{prevline#2}}{\line@num}%Try it
2029 {%
2030 \ifcsstring{symlinenum@#2}}% Try if a symbol is define
2031 {%
2032 \hspace{\csuse{inplaceofnumber@#2}}%
2033 }%
2034 {\hspace{\csuse{beforeasymlinenum@#2}}\notenumfont\csuse{symlinenum
2035 }%
2036 {%
2037 \hspace{\csuse{beforenumberinfootnote@#2}}\notenumfont\printlines
2038 }%
2039 }%
2040 }%

```

```

2041 \hspace{\csuse{beforenumberinfootnote@#2}}\notenumfont\printlines#1|\hspace{\csus
2042 }%
2043 }%
2044 {%
2045 \hspace{\csuse{beforenumberinfootnote@#2}}\notenumfont\printlines#1|\hspace{\csuse{afte
2046 }%
2047 \csnumgdef{prevline#2}{\line@num}%
2048 }%
2049 }%
2050 }%
2051 }%
2052 }

```

## 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.
2053 \countdef\pageno=0 \pageno=1
2054 \newcommand*{\advancepageno}{\ifnum\pageno<\z@ \global\advance\pageno\m@ne
2055 \else\global\advance\pageno\@ne\fi}
2056

```

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
 \vbox{\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 2057 \providecommand{\m@m@makecolfloats}{%
2058 \edef\@freelist{\@freelist\@midlist}%
2059 \global \let \@midlist \@empty
2060 \@combinefloats}
2061 \providecommand{\m@m@makecoltext}{%
2062 \ifvbox\@kludgeins
2063 \@makespecialcolbox
2064 \else
2065 \setbox\@outputbox \vbox to\@colht {%
2066 \@texttop
2067 \dimen@ \dp\@outputbox
2068 \unvbox\@outputbox
2069 \vskip -\dimen@
2070 \@textbottom}%
2071 \fi}
2072 \providecommand{\m@m@makecolintro}{%
2073

```

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

```

2074 \gdef\l@d@makecol{%
2075 \l@ddofootinsert
2076 \m@m@makecolfloats

```

```

2077 \m@m@makecoltext
2078 \global \maxdepth \@maxdepth}
2079

```

`\l@ddofootinsert` This macro essentially holds the initial portion of the kernel `\@makecol` code.

```

2080 \newcommand*{\l@ddofootinsert}{%
2081 %% \page@start
2082 \ifvoid\footins
2083 \setbox\@outputbox \box\@cclv
2084 \else
2085 \setbox\@outputbox \vbox {%
2086 \boxmaxdepth \@maxdepth
2087 \@tempdima\dp\@cclv
2088 \unvbox \@cclv
2089 \vskip \skip\footins
2090 \color@begingroup
2091 \normalcolor
2092 \footnoterule
2093 \unvbox \footins
2094 \color@endgroup
2095 }%
2096 \fi

```

That's the end of the copy of the kernel code. We finally call a macro to handle all the additional EDMAC feet.

```

2097 \l@ddoxtrafeet
2098 }
2099

```

`\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).

```

2100 \newcommand*{\l@ddoxtrafeet}{%
2101 \doxtrafeeti
2102 \doxtrafeetii}
2103

```

`\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'.

```

2104 \newcommand*{\doxtrafeetii}{%
2105 \setbox\@outputbox \vbox{%
2106 \unvbox\@outputbox
2107 \@opxtrafeetii}}

```

`\@opxtrafeetii` The extra critical feet to be added to the output.

```

2108 \newcommand*{\@opxtrafeetii}{%
2109 \ifvoid\Afootins\else\Afootstart{A}\Afootgroup{A}\fi
2110 \ifvoid\Bfootins\else\Bfootstart{B}\Bfootgroup{B}\fi

```



```

2111 \ifvoid\Cfootins\else\Cfootstart{C}\Cfootgroup{C}\fi
2112 \ifvoid\Dfootins\else\Dfootstart{D}\Dfootgroup{D}\fi
2113 \ifvoid\Efootins\else\Efootstart{E}\Efootgroup{E}\fi}
2114

```

`\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`.

```

2115 \newcommand*{\l@ddodoreintrafeet}{%
2116 \doreintrafeeti
2117 \doreintrafeetii}
2118

```

`\doreintrafeetii` `\doreintrafeetii` is the code for catering for the class 2 extra critical footnotes within `\@reinserts`. The implementation may well have to change.

```

2119 \newcommand*{\doreintrafeetii}{%
2120 \ifvoid\Afootins\else\insert\Afootins{\unvbox\Afootins}\fi
2121 \ifvoid\Bfootins\else\insert\Bfootins{\unvbox\Bfootins}\fi
2122 \ifvoid\Cfootins\else\insert\Cfootins{\unvbox\Cfootins}\fi
2123 \ifvoid\Dfootins\else\insert\Dfootins{\unvbox\Dfootins}\fi
2124 \ifvoid\Efootins\else\insert\Efootins{\unvbox\Efootins}\fi
2125 }
2126

```

`\l@d@reinserts` And here is the modified version of `\@reinserts`.

```

2127 \gdef \l@d@reinserts{%
2128 \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
2129 \l@ddodoreintrafeet
2130 \ifvbox\@kludgeins\insert\@kludgeins{\unvbox\@kludgeins}\fi
2131 }
2132

```

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.)

```

2133 \@ifclassloaded{memoir}{%
 memoir is loaded so we use memoir’s built in hooks.
2134 \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
2135 \g@addto@macro{\m@mdodoreintrafeet}{\l@ddodoreintrafeet}%
2136 }{%
 memoir has not been loaded, so redefine @makecol and @reinserts.
2137 \gdef\@makecol{\l@d@makecol}%
2138 \gdef\@reinserts{\l@d@reinserts}%
2139 }
2140

```

`\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.

```
2141 \newcommand*{\addfootins}[1]{%
```

```
2142 \footnormal{#1}
```

Add it to the output.

```
2143 \g@addto@macro{\@opxtrafeetii}{%
```

```
2144 \ifvoid\@nameuse{#1footins}\else
```

```
2145 \@nameuse{#1footstart{#1}}\@nameuse{#1footgroup}{#1}\fi}
```

Add it to the reinsertions.

```
2146 \g@addto@macro{\doreinxtrafeetii}{%
```

```
2147 \ifvoid\@nameuse{#1footins}\else
```

```
2148 \insert\@nameuse{#1footins}{\unvbox\@nameuse{#1footins}}\fi}
```

Add it to minipages.

```
2149 \g@addto@macro{\l@dedbeginmini}{%
```

```
2150 \expandafter\let\csname #1footnote\endcsname = \@nameuse{mp#1footnote}}}
```

And at the end of a minipage.

```
2151 \g@addto@macro{\l@dedendmini}{%
```

```
2152 \ifvoid\@nameuse{mp#1footins}\else\@nameuse{mpfootgroup#1{#1}}\fi}
```

```
2153 }
```

```
2154
```

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.

```
2155 \newif\if@led@nofoot
```

```
2156 \newcommand*{\@led@extranofeet}{}%
```

```
2157
```

```
2158 \ifclassloaded{memoir}{%
```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

`\@mem@extranofeet`

```
2159 \g@addto@macro{\@mem@extranofeet}{%
```

```
2160 \ifvoid\Afootins\else\@mem@nofootfalse\fi
```

```
2161 \ifvoid\Bfootins\else\@mem@nofootfalse\fi
```

```
2162 \ifvoid\Cfootins\else\@mem@nofootfalse\fi
```

```
2163 \ifvoid\Dfootins\else\@mem@nofootfalse\fi
```

```
2164 \ifvoid\Efootins\else\@mem@nofootfalse\fi
```

```
2165 \ifvoid\footinsA\else\@mem@nofootfalse\fi
```

```
2166 \ifvoid\footinsB\else\@mem@nofootfalse\fi
```

```
2167 \ifvoid\footinsC\else\@mem@nofootfalse\fi
```

```
2168 \@led@extranofeet}
```

```
2169 }{%
```

As memoir is not loaded we have to do it all here.

```

\@led@testifnofoot
\@doclearpage 2170 \newcommand*{\@led@testifnofoot}{%
2171 \@led@nofoottrue
2172 \ifvoid\footins\else\@led@nofootfalse\fi
2173 \ifvoid\Afootins\else\@led@nofootfalse\fi
2174 \ifvoid\Bfootins\else\@led@nofootfalse\fi
2175 \ifvoid\Cfootins\else\@led@nofootfalse\fi
2176 \ifvoid\Dfootins\else\@led@nofootfalse\fi
2177 \ifvoid\Efootins\else\@led@nofootfalse\fi
2178 \ifvoid\footinsA\else\@led@nofootfalse\fi
2179 \ifvoid\footinsB\else\@led@nofootfalse\fi
2180 \ifvoid\footinsC\else\@led@nofootfalse\fi
2181 \@led@extranofeet}
2182
2183 \renewcommand{\@doclearpage}{%
2184 \@led@testifnofoot
2185 \if@led@nofoot
2186 \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
2187 \setbox\@tempboxa\box\@cclv
2188 \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
2189 \global \let \@toplist \@empty
2190 \global \let \@botlist \@empty
2191 \global \@colroom \@colht
2192 \ifx \@currlist \@empty
2193 \else
2194 \@latexerr{Float(s) lost}\@ehb
2195 \global \let \@currlist \@empty
2196 \fi
2197 \@makefcolumn\@deferlist
2198 \@whilesw\if@fcolmade \fi{\@opcol\@makefcolumn\@deferlist}%
2199 \if@twocolumn
2200 \if@firstcolumn
2201 \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%
2202 \global \let \@dbltoplist \@empty
2203 \global \@colht \textheight
2204 \beginingroup
2205 \@dblfloatplacement
2206 \@makefcolumn\@dbldeferlist
2207 \@whilesw\if@fcolmade \fi{\@outputpage
2208 \@makefcolumn\@dbldeferlist}%
2209 \endgroup
2210 \else
2211 \vbox{}\clearpage
2212 \fi
2213 \fi
2214 \else
2215 \setbox\@cclv\vbox{\box\@cclv\vfil}%
2216 \l@makecol\@opcol
2217 \clearpage

```

```

2218 \fi}
2219 }
2220

```

## 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.

```

2221 \list@create{\labelref@list}

```

\zz@@@ A convenience macro to zero two labeling counters in one go.

```

2222 %% \newcommand*\zz@@@{000|000|000} % set three counters to zero in one go
2223 \newcommand*\zz@@@{000|000} % set two counters to zero in one go
2224

```

\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.

```

2225 \newcommand*\edlabel}[1]{\@bsphack
2226 \write\linenum@out{\string\@lab}%
2227 \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.

```

2228 \xdef\label@refs{\zz@@@}%
2229 \else
2230 \gl@p\labelref@list\to\label@refs
2231 \ifvmode
2232 \advancelabel@refs
2233 \fi
2234 \fi
2235 % \edef\next{\write\@aux{\string\l@dmake@labels\label@refs|{#1}}}%
2236 % \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).

```

2237 \protected@write\@auxout{}%
2238 {\string\l@dmake@labels\space\thepage|\label@refs|{#1}}%
2239 \@esphack}
2240

```

```

\advancelabel@refs
\labelrefsparseline 2241 %In cases where \cs{edlabel} is the first element in a paragraph, we have a problem with line counts,
\labelrefsparsesubline 2242 %Hence, we need to test \cs{edlabel} if it occurs at the start of a paragraph. To do so, we use \cs{
2243 %We do so using \cs[advancelabel@refs] command.
2244 \newcommand{\advancelabel@refs}{%
2245 \newcount\line%
2246 \ledmac@warning{\label@refs}
2247 \line=\expandafter\labelrefsparseline\label@refs%
2248 \advance\line by 1%
2249 \ifsublines@%
2250 \newcount\subline%
2251 \subline=\labelrefsparsesubline\label@refs%
2252 \advance\subline by 1%
2253 \def\label@refs{\the\line|\the\subline}%
2254 \else%
2255 \def\label@refs{\the\line|0}%
2256 \fi%
2257 }
2258 \def\labelrefsparseline#1|#2{#1}
2259 \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).

```

2260 \newcommand*{\l@dmake@labels}{%
2261 \def\l@dmake@labels#1|#2|#3|#4{%
2262 \expandafter\ifx\csname the@label#4\endcsname \relax\else
2263 \led@warn@DuplicateLabel{#4}%

```

```

2264 \fi
2265 \expandafter\gdef\csname the@label#4\endcsname{#1|#2|#3}%
2266 \ignorespaces}
2267

```

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.

```

2268 \AtBeginDocument{%
2269 \def\l@dmake@labels#1|#2|#3|#4{%
2270 }
2271

```

**\@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.

```

2272 \newcommand*{\@lab}{\xright@appenditem
2273 {\linenumrep{\line@num}}|}%
2274 \ifsublines@ \sublinenumrep{\subline@num}\else 0\fi}\to\labelref@list}
2275

```

**\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.

```

2276 \newcommand*{\edpageref}[1]{\l@dref@undefined{#1}\l@dgetref@num{1}{#1}}
2277 \newcommand*{\xpageref}[1]{\l@dgetref@num{1}{#1}}
2278

```

**\lineref** If the specified label exists, \lineref gives its line number.

```

\lineref 2279 \newcommand*{\lineref}[1]{\l@dref@undefined{#1}\l@dgetref@num{2}{#1}}
2280 \newcommand*{\xlineref}[1]{\l@dgetref@num{2}{#1}}
2281

```

**\sublineref** If the specified label exists, \sublineref gives its sub-line number.

```

\sublineref 2282 \newcommand*{\sublineref}[1]{\l@dref@undefined{#1}\l@dgetref@num{3}{#1}}
2283 \newcommand*{\xsublineref}[1]{\l@dgetref@num{3}{#1}}
2284

```

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.

```
2285 \newcommand*{\l@dref@undefined}[1]{%
2286 \expandafter\ifx\csname the@label#1\endcsname\relax
2287 \led@warn@RefUndefined{#1}%
2288 \fi}
2289
```

`\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.

```
2290 \newcommand*{\l@dgetref@num}[2]{%
2291 \expandafter
2292 \ifx\csname the@label#2\endcsname \relax
2293 000%
2294 \else
2295 \expandafter\expandafter\expandafter
2296 \l@dlabel@parse\csname the@label#2\endcsname| #1%
2297 \fi}
2298
```

`\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`.)

```
2299 \newcommand*{\l@dlabel@parse}{%
2300 \def\l@dlabel@parse#1|#2|#3|#4{%
2301 \ifcase #4\relax
2302 \or #1%
2303 \or #2%
2304 \or #3%
2305 \fi}
2306
```

`\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.

```
2307 \newcommand*{\xxref}[2]{%
2308 {\expandafter\ifx\csname the@label#1\endcsname
2309 \relax \expandafter\let\csname the@label#1\endcsname\zz@@@fi
2310 \expandafter\ifx\csname the@label#2\endcsname \relax
2311 \expandafter\let\csname the@label#2\endcsname\zz@@@fi
2312 \linenum{\csname the@label#1\endcsname|}%
2313 \csname the@label#2\endcsname}}
2314
```

`\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`.

```
2315 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\endcsname{#2}}
2316
```

(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 2317 \newwrite\l@d@end
 2318 \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.
```

```
2319 \newcommand{\l@dend@open}[1]{\global\l@dend@true\immediate\openout\l@d@end=#1\relax}
2320 \newcommand{\l@dend@close}{\global\l@dend@false\immediate\closeout\l@d@end}
2321
```



`\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.

```

2322 \newcommand{\l@dend@stuff}{%
2323 \ifl@dend@relax\else
2324 \l@dend@open{\jobname.end}%
2325 \fi
2326 \immediate\write\l@d@end{\string\l@d@section{\the\section@num}}
2327 }

```

`\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.

```

2328 \newcommand*{\Aendnote}[1]{\{\newlinechar='40
2329 \immediate\write\l@d@end{\string\Aend%
2330 {\ifnumberedpar@l@d@nums\fi}%
2331 {\ifnumberedpar@tag\fi}{#1}}\ignorespaces}
2332 \newcommand*{\Bendnote}[1]{\{\newlinechar='40
2333 \immediate\write\l@d@end{\string\Bend%
2334 {\ifnumberedpar@l@d@nums\fi}%
2335 {\ifnumberedpar@tag\fi}{#1}}\ignorespaces}
2336 \newcommand*{\Cendnote}[1]{\{\newlinechar='40
2337 \immediate\write\l@d@end{\string\Cend%
2338 {\ifnumberedpar@l@d@nums\fi}%
2339 {\ifnumberedpar@tag\fi}{#1}}\ignorespaces}
2340 \newcommand*{\Dendnote}[1]{\{\newlinechar='40
2341 \immediate\write\l@d@end{\string\Dend%
2342 {\ifnumberedpar@l@d@nums\fi}%
2343 {\ifnumberedpar@tag\fi}{#1}}\ignorespaces}
2344 \newcommand*{\Eendnote}[1]{\{\newlinechar='40
2345 \immediate\write\l@d@end{\string\Eend%
2346 {\ifnumberedpar@l@d@nums\fi}%
2347 {\ifnumberedpar@tag\fi}{#1}}\ignorespaces}
2348 }

```

`\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 `\ld@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.

```

2349 \def\endprint#1#2#3{{\notefontsetup{\notenumfont\printendlines#1}}%
2350 \enspace{\select@lemmafnt#1|#2}\enskip#3\par}}
2351 \providecommand*{\@gobblethree}[3]{}
2352 \let\Aend=\@gobblethree
2353 \let\Bend=\@gobblethree
2354 \let\Cend=\@gobblethree
2355 \let\Dend=\@gobblethree
2356 \let\Eend=\@gobblethree
2357 \let\ld@section=\@gobble
2358

```

`\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.

```

2359 \newcommand*{\setprintendlines}[6]{%
2360 \ld@pnumfalse \ld@dashfalse
2361 \ifnum#4=#1 \else
2362 \ld@pnumtrue
2363 \ld@dashtrue
2364 \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.

```

2365 \ifld@pnum \ld@elintrue \else \ld@elinfalse \fi
2366 \ifnum#2=#5 \else
2367 \ld@elintrue
2368 \ld@dashtrue
2369 \fi

```

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

```

2370 \ld@ssubfalse
2371 \ifnum#3=0 \else
2372 \ld@ssubtrue
2373 \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.

```

2374 \ld@eslfalse
2375 \ifnum#6=0 \else

```

```

2376 \ifnum#6=#3
2377 \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
2378 \else
2379 \l@d@esltrue
2380 \l@d@dashtrue
2381 \fi
2382 \fi}

```

`\printendlines` Now we're ready to print it all.

```

2383 \def\printendlines#1|#2|#3|#4|#5|#6|#7|{\begingroup
2384 \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).

```

2385 \printnpnum{#1} \linenumrep{#2}%
2386 \ifl@d@ssub \fullstop \sublinenumrep{#3}\fi
2387 \ifl@d@dash \endashchar\fi
2388 \ifl@d@pnum \printnpnum{#4}\fi
2389 \ifl@d@elin \linenumrep{#5}\fi
2390 \ifl@d@esl \ifl@d@elin \fullstop\fi \sublinenumrep{#6}\fi
2391 \endgroup}
2392

```

`\printnpnum` A macro to print a page number in an endnote.

```

2393 \newcommand*\printnpnum[1]{p.#1} }
2394

```

`\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.

```

2395 \newcommand*\doendnotes[1]{\l@dend@close
2396 \begingroup
2397 \makeatletter
2398 \expandafter\let\csname #1end\endcsname=\endprint
2399 \input\jobname.end
2400 \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.

```

2401 \newcommand*\noendnotes{\global\let\l@dend@stuff=\relax
2402 \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  
`\@xympar` not disturb the flow.

```
2403 \let\l@dold@xympar\@xympar
2404 \renewcommand{\@xympar}{%
2405 \ifnumberedpar@
2406 \led@warn@NoMarginpars
2407 \@esphack
2408 \else
2409 \l@dold@xympar
2410 \fi}
2411
```

We provide side notes as replacement for `\marginpar` in numbered text.

`\sidenote@margin` These are the sidenote equivalents to `\line@margin` and `\linenummargin` for spec-  
`\sidenotemargin` ifying which margin. The default is the right margin (opposite to the default for line  
`\l@dgetsidenote@margin` numbers).

```
2412 \newcount\sidenote@margin
2413 \newcommand*{\sidenotemargin}[1]{%
2414 \l@dgetsidenote@margin{#1}%
2415 \ifnum\@l@dttempcntb>\m@ne
2416 \global\sidenote@margin=\@l@dttempcntb
2417 \fi}}
2418 \newcommand*{\l@dgetsidenote@margin}[1]{%
2419 \def\@tempa{#1}\def\@tempb{left}%
2420 \ifx\@tempa\@tempb
2421 \@l@dttempcntb \z@
2422 \else
2423 \def\@tempb{right}%
2424 \ifx\@tempa\@tempb
2425 \@l@dttempcntb \@ne
2426 \else
2427 \def\@tempb{outer}%
2428 \ifx\@tempa\@tempb
2429 \@l@dttempcntb \tw@
2430 \else
2431 \def\@tempb{inner}%
2432 \ifx\@tempa\@tempb
2433 \@l@dttempcntb \thr@@
2434 \else
2435 \led@warn@BadSidenotemargin
2436 \@l@dttempcntb \m@ne
2437 \fi
2438 \fi
2439 \fi
2440 \fi}
2441 \sidenotemargin{right}
2442
```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

`\l@drp@rbox`

```

2443 \newbox\l@dlp@rbox
2444 \newbox\l@drp@rbox
2445

```

\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 2446 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep 2447 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup 2448 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup 2449 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
2450 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
2451 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
2452

```

\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 2453 \newcommand*\ledleftnote[1]{\edtext{}\l@dlsnote{#1}}
2454 \newcommand*\ledrightnote[1]{\edtext{}\l@drsnote{#1}}
2455 \newcommand*\ledsidenote[1]{\edtext{}\l@dcsnote{#1}}
2456
2457

```

\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 2458 \newif\ifrightrightnoteup
2459 \rightnoteuptrue
2460 \newcommand*\l@dlsnote[1]{%
2461 \ifnumberedpar@
2462 \xright@appenditem{\noexpand\l@dlsnote{#1}}%
2463 \to\inserts@list
2464 \global\advance\insert@count \@ne
2465 \fi\ignorespaces}
2466 \newcommand*\l@drsnote[1]{%
2467 \ifnumberedpar@
2468 \xright@appenditem{\noexpand\l@drsnote{#1}}%
2469 \to\inserts@list
2470 \global\advance\insert@count \@ne
2471 \fi\ignorespaces}
2472 \newcommand*\l@dcsnote[1]{%
2473 \ifnumberedpar@
2474 \xright@appenditem{\noexpand\l@dcsnote{#1}}%
2475 \to\inserts@list
2476 \global\advance\insert@count \@ne
2477 \fi\ignorespaces}
2478

```

\vl@dlsnote Put the left/right text into boxes, but just save the moveable text.

```

\vl@drsnote 2479 \newcommand*\vl@dlsnote[1]{\setl@dlp@rbox{#1}}
\vl@dcsnote 2480 \newcommand*\vl@drsnote[1]{\setl@drp@rbox{#1}}

```

```
2481 \newcommand*{\vl@dcnote}[1]{\gdef\l@dcnotetext{#1}}
2482
```

`\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.

```
2483 \newcommand*{\setl@dlp@rbox}[1]{%
2484 {\parindent\z@\hspace=\ledlsnotewidth\ledlsnotefontsetup
2485 \global\setbox\l@dlp@rbox
2486 \ifleftnoteup
2487 =\vbox to\z@{\vss #1}%
2488 \else
2489 =\vbox to 0.70\baselineskip{\strut#1\vss}%
2490 \fi}}
2491 %% \global\setbox\l@dlp@rbox=\vbox to\z@{\vss #1}% aligns on top line
2492 \newcommand*{\setl@drp@rbox}[1]{%
2493 {\parindent\z@\hspace=\ledrsnotewidth\ledrsnotefontsetup
2494 \global\setbox\l@drp@rbox
2495 \ifrightrightnoteup
2496 =\vbox to\z@{\vss #1}%
2497 \else
2498 =\vbox to 0.7\baselineskip{\strut#1\vss}%
2499 \fi}}
2500 \newif\ifleftnoteup
2501 \leftnoteuptrue
```

`\save\l@dcnote` Save the moveable note text in `\l@dcnotetext`.

```
\l@dcnotetext 2502 \newcommand*{\save\l@dcnote}[3]{%
2503 \gdef\l@dcnotetext{#3}}
2504
```

`\affixside@note` This macro puts any moveable sidenote text into the left or right sidenote box, depending on which margin it is meant to go in. It's a very much stripped down version of `\affixlin@num`.

```
2505 \newcommand*{\affixside@note}{%
2506 \gdef\@templ@d{}%
2507 \ifx\@templ@d\l@dcnotetext \else
2508 \if@twocolumn
2509 \if@firstcolumn
2510 \setl@dlp@rbox{\l@dcnotetext}%
2511 \else
2512 \setl@drp@rbox{\l@dcnotetext}%
2513 \fi
2514 \else
2515 \@l@dttempcntb=\sidenote@margin
2516 \ifnum\@l@dttempcntb>\@ne
2517 \advance\@l@dttempcntb by\page@num
2518 \fi
2519 \ifodd\@l@dttempcntb
```

```

2520 \setl@drp@rbox{\l@dcstotetext}%
2521 \else
2522 \setl@dlp@rbox{\l@dcstotetext}%
2523 \fi
2524 \fi
2525 \fi}
2526

```

## 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.

```

2527 \providecommand*\multiplefootnotemarker{3sp}
2528 \providecommand*\multfootsep{\normalfont,}
2529

```

\m@mmf@prepare A pair of self-cancelling kerns. This may have been defined in the memoir class.

```

2530 \providecommand*\m@mmf@prepare{%
2531 \kern-\multiplefootnotemarker
2532 \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.

```

2533 \providecommand*\m@mmf@check{%
2534 \ifdim\lastkern=\multiplefootnotemarker\relax
2535 \edef\x@sf{\the\spacefactor}%
2536 \unkern
2537 \multfootsep
2538 \spacefactor\x@sf\relax
2539 \fi}
2540

```

We have to modify \@footnotetext and \@footnotemark. However, if memoir is used the modifications have already been made.

```

2541 \ifclassloaded{memoir}{\%

```

\@footnotetext Add \m@mmf@prepare at the end of \@footnotetext.

```

2542 \let\l@dold@footnotetext\@footnotetext
2543 \renewcommand{\@footnotetext}[1]{%
2544 \l@dold@footnotetext{#1}%
2545 \m@mmf@prepare}

```

`\@footnotemark` Modify `\@footnotemark` to cater for adjacent `\footnotes`.

```

2546 \renewcommand*{\@footnotemark}{%
2547 \leavevmode
2548 \ifhmode
2549 \edef\x@sf{\the\spacefactor}%
2550 \m@mmf@check
2551 \nobreak
2552 \fi
2553 \@makefnmark
2554 \m@mmf@prepare
2555 \ifhmode\spacefactor\x@sf\fi
2556 \relax}

```

Finished the modifications for the non-memoir case.

```

2557 }
2558

```

`\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.

```

2559 \let\l@doldold@footnotetext\@footnotetext
2560 \renewcommand{\@footnotetext}[1]{%
2561 \ifnumberedpar@
2562 \edtext{-}{\l@dbfnote{#1}}%
2563 \else
2564 \l@doldold@footnotetext{#1}%
2565 \fi}

```

`\l@dbfnote` `\l@dbfnote` adds the footnote to the insert list, and `\v1@dbfnote` calls the original  
`\v1@dbfnote` `\@footnotetext`.

```

2566 \newcommand{\l@dbfnote}[1]{%
2567 \ifnumberedpar@
2568 \xright@appenditem{\noexpand\v1@dbfnote{#1}}{\@thefnmark}}%
2569 \to\inserts@list
2570 \global\advance\insert@count \@ne
2571 \fi\ignorespaces}
2572 \newcommand{\v1@dbfnote}[2]{%
2573 \def\@thefnmark{#2}%
2574 \l@doldold@footnotetext{#1}}
2575

```

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.

```
2576 \newcommand{\footnoteA}[1]{%
2577 \stepcounter{footnoteA}%
2578 \protected@xdef\@thefnmarkA{\thefootnoteA}%
2579 \@footnotemarkA
2580 \vfootnoteA{A}{#1}\m@mmf@prepare}
2581
```

`\footinsA` The insert for the A series.

```
2582 \newinsert\footinsA
```

`\c@footnoteA` The A series counter.

```
\thefootnoteA 2583 \newcounter{footnoteA}
2584 \renewcommand{\thefootnoteA}{\arabic{footnoteA}}
2585
```

`\footfootmarkA` This macro typesets the A series marker at the start of the footnote text (where it appears at the foot of the page).

```
2586 \newcommand*{\footfootmarkA}{\thefootnoteA}
2587
```

`\mpfootnoteA` The extras for minipages.

```
\mpfootinsA 2588 \newcommand{\mpfootnoteA}[1]{%
2589 \stepcounter{footnoteA}%
2590 \protected@xdef\@thefnmarkA{\thefootnoteA}%
2591 \@footnotemarkA
2592 \mpvfootnoteA{A}{#1}\m@mmf@prepare}
2593 \newinsert\mpfootinsA
2594
```

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 2595 \newcommand*{\prebodyfootmark}{%
2596 \leavevmode
2597 \ifhmode
2598 \edef\x@sf{\the\spacefactor}%
2599 \m@mmf@check
2600 \nobreak
2601 \fi}
```

```

2602 \newcommand{\postbodyfootmark}{%
2603 \m@mmf@prepare
2604 \ifhmode\spacefactor\@x@sf\fi\relax}
2605

```

`\normal@footnotemarkX` `\normal@footnotemarkX{<series>}` sets up the typesetting of the marker at the point where the footnote is called for.

```

2606 \newcommand*{\normal@footnotemarkX}[1]{%
2607 \prebodyfootmark
2608 \@nameuse{bodyfootmark#1}%
2609 \postbodyfootmark}
2610

```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{<series>}` *really* typesets the in-text marker. The style is the normal superscript.

```

2611 \newcommand*{\normalbodyfootmarkX}[1]{%
2612 \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>`.

```

2613 \newcommand*{\normalvfootnoteX}[2]{%
2614 \insert\@nameuse{footins#1}\bgroup
2615 \notefontsetup
2616 \footsplitskips
2617 \spaceskip=\z@skip \xspaceskip=\z@skip
2618 \@nameuse{footfmt#1}{#1}{#2}\egroup}
2619

```

`\mpnormalvfootnoteX` The minipage version.

```

2620 \newcommand*{\mpnormalvfootnoteX}[2]{%
2621 \global\setbox\@nameuse{mpfootins#1}\vbox{%
2622 \unvbox\@nameuse{mpfootins#1}
2623 \notefontsetup
2624 \hsize\columnwidth
2625 \@parboxrestore
2626 \color@begingroup
2627 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
2628

```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

2629 \newcommand*{\normalfootfmtX}[2]{%
2630 \ledsetnormalparstuff
2631 {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2632 #2\strut\par}}
2633

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

2634 \newcommand*{\normalfootfootmarkX}[1]{%
2635 \textsuperscript{\@nameuse{@thefnmark#1}}}%
2636

```

`\normalfootstartX` `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

2637 \newcommand*{\normalfootstartX}[1]{%
2638 \vskip\skip\@nameuse{footins#1}%
2639 \leftskip=\z@
2640 \rightskip=\z@
2641 \@nameuse{footnoterule#1}}%
2642

```

`\normalfootnoteruleX` The rule drawn before the footnote series group.

```

2643 \let\normalfootnoteruleX=\footnoterule
2644

```

`\normalfootgroupX` `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

2645 \newcommand*{\normalfootgroupX}[1]{%
2646 \unvbox\@nameuse{footins#1}}%
2647

```

`\mpnormalfootgroupX` The minipage version.

```

2648 \newcommand*{\mpnormalfootgroupX}[1]{%
2649 \vskip\skip\@nameuse{mpfootins#1}%
2650 \normalcolor
2651 \@nameuse{footnoterule#1}%
2652 \unvbox\@nameuse{mpfootins#1}}%
2653

```

`\normalbfnoteX`

```

2654 \newcommand{\normalbfnoteX}[2]{%
2655 \ifnumberedpar@
2656 \xright@appenditem{\noexpand\vbnoteX{#1}{#2}}{\@nameuse{thefootnote#1}}}%
2657 \to\inserts@list
2658 \global\advance\insert@count \@ne
2659 \fi\ignorespaces}%
2660

```

`\vbnoteX`

```

2661 \newcommand{\vbnoteX}[3]{%
2662 \@namedef{@thefnmark#1}{#3}%
2663 \@nameuse{regvfootnote#1}{#1}{#2}}%
2664

```

`\vnumfootnoteX`

```

2665 \newcommand{\vnumfootnoteX}[2]{%
2666 \ifnumberedpar@

```

```

2667 \edtext{}{\normalbfnoteX{#1}{#2}}%
2668 \else
2669 \@nameuse{regvfootnote#1}{#1}{#2}%
2670 \fi}
2671

```

`\footnormalX` `\footnormalX{<series>}` initialises the settings for the `<series>` footnotes. This should always be called for each series.

```

2672 \newcommand*{\footnormalX}[1]{%
2673 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
2674 \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
2675 \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
2676 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
2677 \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
2678 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
2679 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
2680 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
2681 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
2682 \count\csname footins#1\endcsname=1000
2683 \dimen\csname footins#1\endcsname=\ledfootinsdim
2684 \skip\csname footins#1\endcsname=1.2em \@plus .6em \@minus .6em

```

Additions for minipages.

```

2685 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
2686 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
2687 \count\csname mpfootins#1\endcsname=1000
2688 % \dimen\csname mpfootins#1\endcsname=0.8\vsize
2689 \dimen\csname mpfootins#1\endcsname=\ledfootinsdim
2690 \skip\csname mpfootins#1\endcsname=1.2em \@plus .6em \@minus .6em
2691 }
2692

```

### 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>}`

```

2693 \newcommand*{\foottwocolX}[1]{%
2694 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
2695 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
2696 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
2697 \twocolfootsetupX{#1}
2698 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
2699 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
2700 \mptwocolfootsetupX{#1}}
2701

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX

```

```

2702 \newcommand*{\twocolfootsetupX}[1]{%
2703 \count\csname footins#1\endcsname 500
2704 \multiply\dimen\csname footins#1\endcsname by \tw@}
2705 \newcommand*{\mptwocolfootsetupX}[1]{%
2706 \count\csname mpfootins#1\endcsname 500
2707 \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
2708

\twocolvfootnoteX \twocolvfootnoteX{<series>}

2709 \newcommand*{\twocolvfootnoteX}[2]{%
2710 \insert\csname footins#1\endcsname\bgroup
2711 \notefontsetup
2712 \footplitskips
2713 \spaceskip=\z@skip \xspaceskip=\z@skip
2714 \@nameuse{footfmt#1}{#1}{#2}\egroup}
2715

\twocolfootfmtX \twocolfootfmtX{<series>}

2716 \newcommand*{\twocolfootfmtX}[2]{%
2717 \normal@pars
2718 \hsize .45\hsize
2719 \parindent=\z@
2720 %% \parfillskip=0pt \@plus 1fil
2721 \tolerance=5000\relax
2722 \raggedright
2723 \leavevmode
2724 {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2725 #2\strut\par}\allowbreak}
2726

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX 2727 \newcommand*{\twocolfootgroupX}[1]{\notefontsetup
2728 \splittopskip=\ht\strutbox
2729 \expandafter
2730 \rigidbalance\csname footins#1\endcsname \tw@ \splittopskip}}
2731 \newcommand*{\mptwocolfootgroupX}[1]{%
2732 \vskip\skip\@nameuse{mpfootins#1}
2733 \normalcolor
2734 \@nameuse{footnoterule#1}
2735 \splittopskip=\ht\strutbox
2736 \expandafter
2737 \rigidbalance\csname mpfootins#1\endcsname \tw@ \splittopskip}}
2738

```

### 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>}
2739 \newcommand*{\footthreecolX}[1]{%
2740 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
2741 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
2742 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
2743 \threecolfootsetupX{#1}
2744 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
2745 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
2746 \mpthreecolfootsetupX{#1}}
2747

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX 2748 \newcommand*{\threecolfootsetupX}[1]{%
2749 \count\csname footins#1\endcsname 333
2750 \multiply\dimen\csname footins#1\endcsname by \thr@@
2751 \newcommand*{\mpthreecolfootsetupX}[1]{%
2752 \count\csname mpfootins#1\endcsname 333
2753 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
2754

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
2755 \newcommand*{\threecolvfootnoteX}[2]{%
2756 \insert\csname footins#1\endcsname\bgroup
2757 \notefontsetup
2758 \footssplitskips
2759 \@nameuse{footfmt#1}{#1}{#2}\egroup}
2760

\threecolfootfmtX \threecolfootfmtX{<series>}
2761 \newcommand*{\threecolfootfmtX}[2]{%
2762 \normal@pars
2763 \hsize .3\hsize
2764 \parindent=\z@
2765 %%% \parfillskip=0pt \@plus 1fil
2766 \tolerance=5000\relax
2767 \raggedright
2768 \leavevmode
2769 {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2770 #2\strut\par}\allowbreak}
2771

\threecolfootgroupX \threecolfootgroupX{<series>}
\mpthreecolfootgroupX 2772 \newcommand*{\threecolfootgroupX}[1]{\notefontsetup
2773 \splittopskip=\ht\strutbox
2774 \expandafter
2775 \rigidbalance\csname footins#1\endcsname \thr@@ \splittopskip}}
2776 \newcommand*{\mpthreecolfootgroupX}[1]{%
2777 \vskip\skip\@nameuse{mpfootins#1}
2778 \normalcolor

```

```

2779 \@nameuse{footnoterule#1}
2780 \splittopskip=\ht\strutbox
2781 \expandafter
2782 \rigidbalance\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
2783

```

### 27.2.3 Paragraphed footnotes

The following macros set footnotes as one paragraph.

```

\footparagraphX \footparagraphX{<series>}
2784 \newcommand*{\footparagraphX}[1]{%
2785 \expandafter\newcount\csname prevpage#1@num\endcsname
2786 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
2787 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
2788 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
2789 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
2790 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
2791 \count\csname footins#1\endcsname=1000
2792 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
2793 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
2794 \count\csname mpfootins#1\endcsname=1000
2795 \para@footsetupX{#1}}
2796

```

```

\para@footsetupX \para@footsetupX{<series>}
2797 \newcommand*{\para@footsetupX}[1]{\notefontsetup
2798 \dimen0=\baselineskip
2799 \multiply\dimen0 by 1024
2800 \divide\dimen0 by \hsize \multiply\dimen0 by \footfudgefiddle\relax
2801 \expandafter
2802 \xdef\csname footfudgefactor#1\endcsname{%
2803 \expandafter\strip@pt\dimen0 }}
2804

```

```

\parafootstartX \parafootstartX{<series>}
2805 \newcommand*{\parafootstartX}[1]{%
2806 \vskip\skip\@nameuse{footins#1}%
2807 \leftskip=\z@
2808 \rightskip=\z@
2809 \parindent=\z@
2810 \vskip\skip\@nameuse{footins#1}%
2811 \@nameuse{footnoterule#1}}
2812

```

```

\para@vfootnoteX \para@vfootnoteX{<series>}{<text>}
\mppara@vfootnoteX 2813 \newcommand*{\para@vfootnoteX}[2]{%
2814 \insert\csname footins#1\endcsname
2815 \bgroup

```

```

2816 \notefontsetup
2817 \footsplitskips
2818 \setbox0=\vbox{\hsize=\maxdimen
2819 \noindent\@nameuse{footfmt#1}{#1}{#2}}%
2820 \setbox0=\hbox{\unvbox0[#1]}%
2821 \dp0=\z@
2822 \ht0=\csname footfudgefactor#1\endcsname\wd0
2823 \box0
2824 \penalty0
2825 \egroup}
2826 \newcommand*{\mppara@vfootnoteX}[2]{%
2827 \global\setbox\@nameuse{mpfootins#1}\vbox{%
2828 \unvbox\@nameuse{mpfootins#1}
2829 \notefontsetup
2830 \footsplitskips
2831 \setbox0=\vbox{\hsize=\maxdimen
2832 \noindent\color@begingroup\@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
2833 \setbox0=\hbox{\unvbox0[#1]}%
2834 \dp0=\z@
2835 \ht0=\csname footfudgefactor#1\endcsname\wd0
2836 \box0
2837 \penalty0}}
2838
\parafootfmtX \parafootfmtX{<series>}
2839 \newcommand*{\parafootfmtX}[2]{%
2840 \insertparafootsep{#1}%
2841 \ledsetnormalparstuff
2842 {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2843 #2\penalty-10}}
2844
\para@footgroupX \para@footgroupX{<series>}
\mppara@footgroupX 2845 \newcommand*{\para@footgroupX}[1]{%
2846 \unvbox\csname footins#1\endcsname
2847 \makehboxofhboxes
2848 \setbox0=\hbox{\unhbox0 \removehboxes}%
2849 \notefontsetup
2850 \noindent\unhbox0\par}
2851 \newcommand*{\mppara@footgroupX}[1]{%
2852 \vskip\skip\@nameuse{mpfootins#1}
2853 \normalcolor
2854 \@nameuse{footnoterule#1}
2855 \unvbox\csname mpfootins#1\endcsname
2856 \makehboxofhboxes
2857 \setbox0=\hbox{\unhbox0 \removehboxes}%
2858 \notefontsetup
2859 \noindent\unhbox0\par}}
2860

```



### 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.

```
2861 \newcommand{\footnoteB}[1]{%
2862 \stepcounter{footnoteB}%
2863 \protected@xdef\@thefnmarkB{\thefootnoteB}%
2864 \@footnotemarkB
2865 \vfootnoteB{B}{#1}\m@mmf@prepare}
2866
```

`\c@footnoteB`

```
\thefootnoteB 2867 \newcounter{footnoteB}
2868 \renewcommand{\thefootnoteB}{\arabic{footnoteB}}
2869
```

`\footinsB`

```
2870 \newinsert\footinsB
2871
```

`\mpfootnoteB` The extras for minipages.

```
\mpfootinsB 2872 \newcommand{\mpfootnoteB}[1]{%
2873 \stepcounter{footnoteB}%
2874 \protected@xdef\@thefnmarkB{\thefootnoteB}%
2875 \@footnotemarkB
2876 \mpvfootnoteB{B}{#1}\m@mmf@prepare}
2877 \newinsert\mpfootinsB
2878
```

`\footnoteC` `\footnoteC{<text>}` is the user command for a series C footnote.

```
2879 \newcommand{\footnoteC}[1]{%
2880 \stepcounter{footnoteC}%
2881 \protected@xdef\@thefnmarkC{\thefootnoteC}%
2882 \@footnotemarkC
2883 \vfootnoteC{C}{#1}\m@mmf@prepare}
```

`\c@footnoteC`

```
\thefootnoteC 2884 \newcounter{footnoteC}
\footinsC 2885 \renewcommand{\thefootnoteC}{\arabic{footnoteC}}
2886 \newinsert\footinsC
2887
```

`\mpfootnoteC` The extras for minipages.

```
\mpfootinsC 2888 \newcommand{\mpfootnoteC}[1]{%
2889 \stepcounter{footnoteC}%
2890 \protected@xdef\@thefnmarkC{\thefootnoteC}%
2891 \@footnotemarkC
2892 \mpvfootnoteC{C}{#1}\m@mmf@prepare}
2893 \newinsert\mpfootinsC
2894
```

Don't forget to initialise the series.

```
2895 \footnormalX{A}
2896 \footnormalX{B}
2897 \footnormalX{C}
2898
```

`\doxtrafeeti` We have to add all the new kinds of familiar footnotes to the output routine. These  
`\doreinxtrafeeti` are the class 1 feet.

```
2899 \newcommand*{\doxtrafeeti}{%
2900 \setbox\@outputbox \vbox{%
2901 \unvbox\@outputbox
2902 \ifvoid\footinsA\else\footstartA{A}\footgroupA{A}\fi
2903 \ifvoid\footinsB\else\footstartB{B}\footgroupB{B}\fi
2904 \ifvoid\footinsC\else\footstartC{C}\footgroupC{C}\fi
2905 }}
2906
2907 \newcommand{\doreinxtrafeeti}{%
2908 \ifvoid\footinsA\else\insert\footinsA{\unvbox\footinsA}\fi
2909 \ifvoid\footinsB\else\insert\footinsB{\unvbox\footinsB}\fi
2910 \ifvoid\footinsC\else\insert\footinsC{\unvbox\footinsC}\fi
2911 }
2912
```

`\addfootinsX` Make life just a little easier for those who want additional series of class 1 footnotes.

```
2913 \newcommand*{\addfootinsX}[1]{%
2914 \footnormalX{#1}%
2915 \g@addto@macro{\doxtrafeeti}{%
2916 \setbox\@outputbox \vbox{%
2917 \unvbox\@outputbox
2918 \ifvoid\@nameuse{footins#1}\else
2919 \@nameuse{footstart#1}{#1}\@nameuse{footgroup#1}{#1}\fi}%
2920 \g@addto@macro{\doreinxtrafeeti}{%
2921 \ifvoid\@nameuse{footins#1}\else
2922 \insert\@nameuse{footins#1}{\unvbox\@nameuse{footins#1}\fi}%
2923 \g@addto@macro{\l@dfambeginmini}{%
2924 \expandafter\expandafter\expandafter\let\expandafter\expandafter
2925 \csname footnote#1\endcsname \csname mpfootnote#1\endcsname}%
2926 \g@addto@macro{\l@dfamendmini}{%
2927 \ifvoid\@nameuse{mpfootins#1}\else\@nameuse{mpfootgroup#1}{#1}\fi}%
2928 }
```

## 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 `\@iiiminipage` and `\endminipage` macros. We'll arrange this so that additional series can be easily added.

`\l@dfetbeginmini` These will be the hooks in `\@iiiminipage` and `\endminipage` They can be extended  
`\l@dfetendmini` to handle other things if necessary.

```
2929 \newcommand*{\l@dfetbeginmini}{\l@dedbeginmini\l@dfambeginmini}
2930 \newcommand*{\l@dfetendmini}{\l@dedendmini\l@dfamendmini}
2931
```

`\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.

```
2932 \newcommand*{\l@dedbeginmini}{%
2933 \let\Afootnote=\mpAfootnote%
2934 \let\Bfootnote=\mpBfootnote%
2935 \let\Cfootnote=\mpCfootnote%
2936 \let\Dfootnote=\mpDfootnote%
2937 \let\Efootnote=\mpEfootnote}
2938 \newcommand*{\l@dedendmini}{%
2939 \ifvoid\mpAfootins\else\mpAfootgroup{A}\fi%
2940 \ifvoid\mpBfootins\else\mpBfootgroup{B}\fi%
2941 \ifvoid\mpCfootins\else\mpCfootgroup{C}\fi%
2942 \ifvoid\mpDfootins\else\mpDfootgroup{D}\fi%
2943 \ifvoid\mpEfootins\else\mpEfootgroup{E}\fi}
2944
```

`\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.

```
2945 \newcommand*{\l@dfambeginmini}{%
2946 \let\footnoteA=\mpfootnoteA%
2947 \let\footnoteB=\mpfootnoteB%
2948 \let\footnoteC=\mpfootnoteC}
2949 \newcommand*{\l@dfamendmini}{%
2950 \ifvoid\mpfootinsA\else\mpfootgroupA{A}\fi%
2951 \ifvoid\mpfootinsB\else\mpfootgroupB{B}\fi%
2952 \ifvoid\mpfootinsC\else\mpfootgroupC{C}\fi}
2953
```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```
2954 \def\@iiiminipage#1#2[#3]#4{%
2955 \leavevmode
2956 \@pboxswfalse
2957 \setlength\@tempdima{#4}%
2958 \def\@mpargs{#1}#2[#3]#4}%
2959 \setbox\@tempboxa\vbox\bgroup
2960 \color@begingroup
2961 \hsize\@tempdima
2962 \textwidth\hsize \columnwidth\hsize
2963 \@parboxrestore
2964 \def\@mpfn{\mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
2965 \let\@footnotetext\@mpfootnotetext
```

The next line is our addition to the original.

```

2966 \l@dfeetbeginmini% added
2967 \let\@listdepth\@mplistdepth \@mplistdepth\z@
2968 \@minipagerestore
2969 \setminipage}
2970

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

2971 \def\endminipage{%
2972 \par
2973 \unskip
2974 \ifvoid\@mpfootins\else
2975 \l@dunboxmpfoot
2976 \fi

```

The next line is our addition to the original.

```

2977 \l@dfeetendmini% added
2978 \@minipagefalse
2979 \color@endgroup
2980 \egroup
2981 \expandafter\@iiiparbox\@mpargs{\unvbox\@tempboxa}}
2982

```

`\l@dunboxmpfoot`

```

2983 \newcommand*{\l@dunboxmpfoot}{%
2984 \vskip\skip\@mpfootins
2985 \normalcolor
2986 \footnoterule
2987 \unvbox\@mpfootins}
2988

```

`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.

```

2989 \newenvironment{ledgroup}{%
2990 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
2991 \let\@footnotetext\@mpfootnotetext
2992 \l@dfeetbeginmini%
2993 }{%
2994 \par
2995 \unskip
2996 \ifvoid\@mpfootins\else
2997 \l@dunboxmpfoot
2998 \fi
2999 \l@dfeetendmini%
3000 }
3001

```

`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 *width* minipage. The optional *pos* controls the sideways position of numbered text.

```
3002 \newenvironment{ledgroupsize}[2][1]{%
```

Set the various text measures.

```
3003 \hsize #2\relax
```

```
3004 %% \textwidth #2\relax
```

```
3005 %% \columnwidth #2\relax
```

Initialize fills for centering.

```
3006 \let\ledllfill\hfil
```

```
3007 \let\ledrlfill\hfil
```

```
3008 \def\@tempa{#1}\def\@tempb{1}%
```

Left adjusted numbered lines

```
3009 \ifx\@tempa\@tempb
```

```
3010 \let\ledllfill\relax
```

```
3011 \else
```

```
3012 \def\@tempb{r}%
```

```
3013 \ifx\@tempa\@tempb
```

Right adjusted numbered lines

```
3014 \let\ledrlfill\relax
```

```
3015 \fi
```

```
3016 \fi
```

Set up the footnoting.

```
3017 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
```

```
3018 \let\@footnotetext\@mpfootnotetext
```

```
3019 \l@dfetbeginmini%
```

```
3020 }{%
```

```
3021 \par
```

```
3022 \unskip
```

```
3023 \ifvoid\@mpfootins\else
```

```
3024 \l@dunboxmpfoot
```

```
3025 \fi
```

```
3026 \l@dfeetendmini%
```

```
3027 }
```

```
3028
```

## 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 3029 \newcommand{\pagelinesep}{-}
```

```
3030 \newcommand{\edindexlab}{$&}
```

```
3031 \newcounter{labidx}
```

```
3032 \setcounter{labidx}{0}
```

```
3033
```

`\doedindexlabel` This macro sets an `\edlabel`.

```
3034 \newcommand{\doedindexlabel}{\stepcounter{labidx}%
3035 \edlabel{\edindexlab\thelabidx}}
3036
```

`\thepageline` This macro makes up the page/line number combo from the label/ref.

```
3037 \newcommand{\thepageline}{%
3038 \thepage\pagelinesep\lineref{\edindexlab\thelabidx}}
3039
```

The memoir class provides more flexible indexing than the standard classes. We need different code if the memoir class is being used.

```
3040 \ifclassloaded{memoir}{%
 memoir is being used.
```

`\makeindex` Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing. In this case `\edindex` has an optional argument. We use the hook provided in memoir v1.61.

```
3041 \g@addto@macro{\makememindexhook}{%
3042 \def\edindex{\@bsphack%
3043 \ifnextchar [{\l@d@index}{\l@d@index[\jobname]}}
3044 \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 a virtually a verbatim copy of memoir's `\@index`, the change being calling `\l@dwrindexm@m` instead of `\@wrindexm@m`.

```
3045 \def\l@d@index[#1]{%
3046 \@ifundefined{#1idxfile}%
3047 {\ifreportnoidxfile
3048 \led@warn@NoIndexFile{#1}%
3049 \fi
3050 \begingroup
3051 \@sanitize
3052 \nowrindex}%
3053 {\def\@idxfile{#1}%
3054 \doedindexlabel
3055 \begingroup
3056 \@sanitize
3057 \l@d@wrindexm@m}}
```

`\l@d@wrindexm@m` `\l@d@wrindexm@m{item}` writes the idx file name and the indexed item to the aux file. These are almost verbatim copies of memoir's `\@wrindexm@m` and `\@wrindexhyp`.

```
3058 \newcommand{\l@d@wrindexm@m}[1]{\l@d@wrindexhyp#1||\%}
3059 \def\l@d@wrindexhyp#1|#2|#3\%{
3060 \ifshowindexmark\@showidx{#1}\fi
3061 \ifx\#2\%
```

```

3062 \protected@write\@auxout{}%
3063 {\string\@wrindexm@m{\@idxfile}{#1|hyperpage}{\thepage}}}%
3064 \else
3065 \def\Hy@temp@A{#2}%
3066 \ifx\Hy@temp@A\HyInd@ParenLeft
3067 \protected@write\@auxout{}%
3068 {\string\@wrindexm@m{\@idxfile}{#1|#2hyperpage}{\thepage}}}%
3069 \else
3070 \protected@write\@auxout{}%
3071 {\string\@wrindexm@m{\@idxfile}{#1|#2}{\thepage}}}%
3072 \fi
3073 \fi
3074 \endgroup
3075 \@esphack}

```

That finishes the memoir-specific code.

```
3076 }{%
```

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.

```

3077 \g@addto@macro{\makeindex}{%
3078 \def\edindex{\@bsphack
3079 \doedindexlabel
3080 \begingroup
3081 \@sanitize
3082 \@wredindex}}
3083 \newcommand{\edindex}[1]{\@bsphack\@esphack}

```

`\@wredindex` Write the index information to the idx file.

```

3084 \newcommand{\@wredindex}[1]{%
3085 \protected@write\@indexfile{}%
3086 {\string\indexentry{#1}{\thepage}}}%
3087 \endgroup
3088 \@esphack}

```

That finishes the non-memoir index code.

```
3089 }
```

```
3090
```

`\l@d@wrindexhyp` If the hyperref package is not loaded, it doesn't make sense to clutter up the index with hyperreffing things.

```

3091 \AtBeginDocument{\@ifpackageloaded{hyperref}{}{%
3092 \def\l@d@wrindexhyp#1||\{\%
3093 \ifshowindexmark\@showidx{#1}\fi
3094 \protected@write\@auxout{}%
3095 {\string\@wrindexm@m{\@idxfile}{#1}{\thepage}}}%
3096 \endgroup
3097 \@esphack}}}
3098

```

## 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.

```
3099 \newtoks\@emptytoks
3100
```

The rest is from `amsmath`.

`\l@denvbody` A token register to contain the body.

```
3101 \newtoks\l@denvbody
3102
```

`\addtol@denvbody` `\addtol@denvbody{arg}` adds `arg` to the token register `\l@denvbody`.

```
3103 \newcommand{\addtol@denvbody}[1]{%
3104 \global\l@denvbody\expandafter{\the\l@denvbody#1}}
3105
```

`\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`.

```
3106 \newcommand{\l@dcollect@body}[1]{%
3107 \l@denvbody{\expandafter#1\expandafter{\the\l@denvbody}}%
3108 \edef\processl@denvbody{\the\l@denvbody\noexpand\end{\@currenvir}}%
3109 \l@denvbody\@emptytoks \def\l@dbegin@stack{b}%
3110 \begingroup
3111 \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
3112 \edef\processl@denvbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
3113 \processl@denvbody}
3114
```

`\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.

```
3115 \def\l@dpush@begins#1\begin#2{%
3116 \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
3117
```

`\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.

```

3118 \def\l@collect@@body#1\end#2{%
3119 \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
3120 \expandafter\@gobble\l@dbegin@stack}%
3121 \ifx\@empty\l@dbegin@stack
3122 \endgroup
3123 \@checkend{#2}%
3124 \addtol@denvbody{#1}%
3125 \else
3126 \addtol@denvbody{#1\end{#2}}%
3127 \fi
3128 \processl@denvbody % A little tricky! Note the grouping
3129 }
3130
```

There was a question on CTT about how to use `\collect@body` for a macro taking an argument. The following is part of that thread.

From: Heiko Oberdiek <oberdiek@uni-freiburg.de>  
 Newsgroups: comp.text.tex  
 Subject: Re: Using `\collect@body` with commands that take >1 argument  
 Date: Fri, 08 Aug 2003 09:03:20 +0200

eed132@psu.edu (Evan) wrote:  
 > I'm trying to make a new Latex environment that acts like the  
 > `\colorbox` command that is part of the `color` package. I looked through  
 > the FAQ and ran across this bit about using the `\collect@body` command  
 > that is part of AMSLaTeX:  
 > <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=cmdasenv>  
 >  
 > It almost works. If I do something like the following:  
 > `\newcommand{\redbox}[1]{\colorbox{red}{#1}}`  
 >  
 > `\makeatletter`  
 > `\newenvironment{redbox}{\collect@body \redbox}{}`

You will get an error message: Command `\redbox` already defined.  
 Thus you must rename either the command `\redbox` or the environment name.

```

> \begin{coloredbox}{blue}
> Yadda yadda yadda... this is on a blue background...
> \end{coloredbox}
> and can't figure out how to make the \collect@body take this.

> \collect@body \colorbox{red}
> \collect@body {\colorbox{red}}
```

The argument of `\collect@body` has to be one token exactly.

```

\documentclass{article}
\usepackage{color}
\usepackage{amsmath}

\newcommand{\redbox}[1]{\colorbox{red}{#1}}
\makeatletter
\newenvironment{coloredbox}[1]{%
 \def\next@{\colorbox{#1}}%
 \collect@body\next@
}{%

% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}[1]{%
 \def\next@{\mycoloredbox{#1}}%
 \collect@body\next@
}{%
\newcommand{\mycoloredbox}[2]{%
 \colorbox{#1}{\ignorespaces#2\unskip}%
}

% support of optional color model argument
\newcommand\coloredboxIII\endcsname{}
\def\coloredboxIII#1#{%
 \@coloredboxIII{#1}%
}
\def\@coloredboxIII#1#2{%
 \def\next@{\mycoloredboxIII{#1}{#2}}%
 \collect@body\next@
}
\newcommand{\mycoloredboxIII}[3]{%
 \colorbox{#1}{#2}{\ignorespaces#3\unskip}%
}

\makeatother

\begin{document}
 Black text before
 \begin{coloredbox}{blue}
 Hello World
 \end{coloredbox}
 Black text after

 Black text before
 \begin{coloredboxII}{blue}
 Hello World
 \end{coloredboxII}
 Black text after

```

```

Black text before
\begin{coloredboxIII}[rgb]{0,0,1}
 Hello World
\end{coloredboxIII}
Black text after

\end{document}

Yours sincerely
Heiko <oberdiek@uni-freiburg.de>

```

## 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 3131 \newcommand*{\hangingsymbol}{ }
 3132 \newif\ifinstanza
 3133 \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.

3134 \newif\ifinserthangingsymbol
3135 \newcommand{\inserthangingsymbol}{%
3136 \ifinserthangingsymbol%
3137 \ifinstanza%
3138 \hfill\hangingsymbol%
3139 \fi%
3140 \fi%
3141 }

```

`\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.

```

3142 \newcommand*{\ampersand}{\char'\&}
3143

```

`\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.

```
3144 \chardef\body=\catcode'\@
3145 \catcode'\@=11
3146 \chardef\next=\catcode'\&
3147 \catcode'\&=\active
3148
```

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.

```
3149 \newcount\stanza@count
3150 \newlength{\stanzaindentbase}
3151 \setlength{\stanzaindentbase}{20pt}
3152
```

`\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.

```
3153 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
3154 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,,|}%
3155 \stanza@count\z@
3156 \def\next{\expandafter\strip@szacnt\@tempa
3157 \ifx\@tempb\empty\let\next\relax\else
3158 \expandafter\mathchardef\csname #1@\number\stanza@count
3159 \@endcsname\@tempb\relax
3160 \advance\stanza@count\@ne\fi\next}%
3161 \next}
3162
```

`\setstanzaindents` 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 `stanzaindentsrepetition` counter can be used when the indentation is repeated every *n* verses. The `\managestanza@modulo` is a command which modifies the counter `stanza@modulo`. The command adds 1 to `stanza@modulo`, but if `stanza@modulo` is equal to the `stanzaindentsrepetition` counter, the command restarts it.

```
3163 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
3164 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
3165
3166 \newcounter{stanzaindentsrepetition}
3167 \newcount\stanza@modulo
3168
```

```

3169 \newcommand*{\managestanza@modulo}[0]{
3170 \advance\stanza@modulo\@ne
3171 \ifnum\stanza@modulo>\value{stanzaindentrepetition}
3172 \stanza@modulo\@ne
3173 \fi
3174 }

```

`\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` `\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.

```

3175 \def\stanza@line{
3176 \ifnum\value{stanzaindentrepetition}=0
3177 \parindent=\csname sza@\number\stanza@count
3178 @\endcsname\stanzaindentbase
3179 \else
3180 \managestanza@modulo
3181 \parindent=\csname sza@\number\stanza@modulo
3182 @\endcsname\stanzaindentbase
3183 \fi
3184 \pstart\stanza@hang\ignorespaces}
3185 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
3186 \hangindent\expandafter
3187 \noexpand\csname sza@0@\endcsname\stanzaindentbase
3188 \hangafter\@ne}
3189 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
3190 \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
3191 \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 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.

```

3192 \let\startstanzahook\relax
3193 \let\endstanzaextra\relax
3194 \xdef\stanza{\noexpand\instanzatrue\expandafter

```

```

3195 \begingroup\startstanzahook%
3196 \catcode'\&\active\global\stanza@count\@ne\stanza@modulo\@ne
3197 \noexpand\ifnum\expandafter\noexpand
3198 \csname sza@0@\endcsname=\z@\let\noexpand\stanza@hang\relax
3199 \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
3200 \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
3201 \expandafter\noexpand\csname szp@0@\endcsname=\z@
3202 \let\noexpand\sza@penalty\relax\noexpand\fi \def\noexpand&{%
3203 \noexpand\endlock\noexpand\pend\noexpand\sza@penalty\global
3204 \advance\stanza@count\@ne\noexpand\stanza@line}\def\noexpand
3205 &{\noexpand\endlock\noexpand\pend\endgroup\noexpand\instanzafalse\expandafter\er
3206 \noexpand\stanza@line}
3207

```

`\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`.

```

3208 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
3209 \hskip -#1\llap{#2}\hskip #1\ignorespaces}
3210

```

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.

```

3211 \catcode'\&=\next
3212 \catcode'\@=\body
3213 %% \let\ampersand=\&
3214 \setstanzavalues{szp}{0}
3215

```

## 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, \footnote 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..

```
3216 \newcommand*{\l@dtabnoexpands}{%
3217 \def\ss{\noexpand\ss}%
3218 \def\"##1{\noexpand\"##1}%
3219 \def\'##1{\noexpand\'##1}%
3220 \def\`##1{\noexpand\`##1}%
3221 \def\^##1{\noexpand\^##1}%
3222 \def\phantom##1{\noexpand}%
3223 \def\hphantom##1{\noexpand\hphantom{##1}}%
3224 \def\underbrace##1{\noexpand\underbrace{##1}}%
3225 \def\dots{\noexpand\dots}%
3226 \let\rtab=0%
3227 \let\ctab=0%
3228 \let\ltab=0%
3229 \let\rtabtext=0%
3230 \let\ltabtext=0%
3231 \let\ctabtext=0%
3232 \let\edbeforetab=0%
3233 \let\edaftertab=0%
3234 \let\edatab=0%
3235 \let\edatabell=0%
3236 \let\edatleft=0%
3237 \let\edatright=0%
3238 \let\edvertline=0%
3239 \let\edvertdots=0%
3240 \let\edrowfill=0%
3241 }
3242
```

`\l@dampcount` `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter for the columns. These were `\Undcount` and `\stellencount` respectively.

```
3243 \newcount\l@dampcount
3244 \l@dampcount=1\relax
3245 \newcount\l@dcolcount
3246 \l@dcolcount=0\relax
3247
```

```

\hilfsbox Some (temporary) helper items.
\hilfsskip 3248 \newbox\hilfsbox
\Hilfsbox 3249 \newskip\hilfsskip
\hilfscount 3250 \newbox\Hilfsbox
3251 \newcount\hilfscount
3252

```

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).

```

3253 \newdimen\dcoli
3254 \newdimen\dcolii
3255 \newdimen\dcoliii
3256 \newdimen\dcoliv
3257 \newdimen\dcolv
3258 \newdimen\dcolvi
3259 \newdimen\dcolvii
3260 \newdimen\dcolviii
3261 \newdimen\dcolix
3262 \newdimen\dcolx
3263 \newdimen\dcolxi
3264 \newdimen\dcolxii
3265 \newdimen\dcolxiii
3266 \newdimen\dcolxiv
3267 \newdimen\dcolxv
3268 \newdimen\dcolxvi
3269 \newdimen\dcolxvii
3270 \newdimen\dcolxviii
3271 \newdimen\dcolxix
3272 \newdimen\dcolxx
3273 \newdimen\dcolxxi
3274 \newdimen\dcolxxii
3275 \newdimen\dcolxxiii
3276 \newdimen\dcolxxiv
3277 \newdimen\dcolxxv
3278 \newdimen\dcolxxvi
3279 \newdimen\dcolxxvii
3280 \newdimen\dcolxxviii
3281 \newdimen\dcolxxix
3282 \newdimen\dcolxxx
3283 \newdimen\dcolerr % added for error handling
3284

```

\l@dcolwidth This is a cunning way of storing the columnwidths indexed by the column number \l@dcolcount, like an array. (was \Dimenzuordnung)

```

3285 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???
3286 \or \dcoli \or \dcolii \or \dcoliii
3287 \or \dcoliv \or \dcolv \or \dcolvi
3288 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx

```



```

3289 \or \dcolxi \or \dcolxii \or \dcolxiii
3290 \or \dcolxiv \or \dcolxv \or \dcolxvi
3291 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
3292 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
3293 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
3294 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
3295 \else \dcolerr \fi}
3296

```

`\step1@dcolcount` This increments the column counter, and issues an error message if it is too large.

```

3297 \newcommand*{\step1@dcolcount}{\advance\l@dcolcount\@ne
3298 \ifnum\l@dcolcount>30\relax
3299 \led@err@TooManyColumns
3300 \fi}
3301

```

`\l@dsetmaxcolwidth` Sets the column width to the maximum value seen so far. (was `\dimenzuordnung`)

```

3302 \newcommand{\l@dsetmaxcolwidth}{%
3303 \ifdim\l@dcolwidth < \wd\hilfsbox
3304 \l@dcolwidth = \wd\hilfsbox
3305 \else \relax \fi}
3306

```

`\EDTEXT` We need to be able to modify the `\edtext` and `\critext` macros and also restore `\xedtext` their original definitions.

```

\CRITEXT 3307 \let\EDTEXT=\edtext
\xcritext 3308 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
3309 \let\CRITEXT=\critext
3310 \long\def\xcritext #1#2/{\CRITEXT{#1}{#2}/}

```

`\EDLABEL` We need to be able to modify and restore the `\edlabel` macro.

```

\xedlabel 3311 \let\EDLABEL=\edlabel
3312 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}

```

`\EDINDEX` Macros supporting modification and restoration of `\edindex`.

```

\xedindex 3313 \let\EDINDEX=\edindex
\nulledindex 3314 \ifl@dmemoir
3315 \newcommand{\xedindex}{\@bsphack%
3316 \ifnextchar [{\l@d@index}{\l@d@index[\jobname]}}
3317 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
3318 \else
3319 \newcommand{\xedindex}{\@bsphack%
3320 \doedindexlabel
3321 \begingroup
3322 \@sanitize
3323 \wredindex}
3324 \newcommand{\nulledindex}[1]{\@bsphack\@esphack}
3325 \fi
3326

```

```

\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 3327 \let\A@@footnote=\Afootnote
\D@@footnote 3328 \let\B@@footnote=\Bfootnote
\E@@footnote 3329 \let\C@@footnote=\Cfootnote
 3330 \let\D@@footnote=\Dfootnote
 3331 \let\E@@footnote=\Efootnote

\@line@count Macro supporting restoration of \linenum.
 3332 \let\@line@count=\linenum

\l@dgobbledarg \l@dgobbledarg replaces its delineated argument by \relax (was \verschwinden).
\l@dgobblearg \l@dgobblearg{\arg} replaces its argument by \relax.
 3333 \def\l@dgobbledarg #1/{\relax}
 3334 \newcommand*\l@dgobblearg}[1]{\relax}
 3335

\Relax
\NEXT 3336 \let\Relax=\relax
\@hilfs@count 3337 \let\NEXT=\next
 3338 \newcount\@hilfs@count
 3339

\measuremcell Measure (recursively) the width required for a math cell. (was \messen)
 3340 \def\measuremcell #1{%
 3341 \ifx #1\ \ifnum\l@dc@count=0\let\NEXT\relax%
 3342 \else\l@dcheckcols%
 3343 \l@dc@count=0%
 3344 \let\NEXT\measuremcell%
 3345 \fi%
 3346 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
 3347 \step\l@dc@count%
 3348 \l@dsetmaxcolwidth%
 3349 \let\NEXT\measuremcell%
 3350 \fi\NEXT}
 3351

\measuretcell Measure (recursively) the width required for a text cell. (was \messentext)
 3352 \def\measuretcell #1{%
 3353 \ifx #1\ \ifnum\l@dc@count=0\let\NEXT\relax%
 3354 \else\l@dcheckcols%
 3355 \l@dc@count=0%
 3356 \let\NEXT\measuretcell%
 3357 \fi%
 3358 \else\setbox\hilfsbox=\hbox{#1}%
 3359 \step\l@dc@count%
 3360 \l@dsetmaxcolwidth%
 3361 \let\NEXT\measuretcell%
 3362 \fi\NEXT}
 3363

```

`\measuremrow` Measure (recursively) the width required for a math row. (was `\Messen`)

```
3364 \def\measuremrow #1\{\%
3365 \ifx #1&\let\NEXT\relax%
3366 \else\measuremcell #1&\&\&\&%
3367 \let\NEXT\measuremrow%
3368 \fi\NEXT}
```

`\measuretrow` Measure (recursively) the width required for a text row. (was `\Messentext`)

```
3369 \def\measuretrow #1\{\%
3370 \ifx #1&\let\NEXT\relax%
3371 \else\measuretcell #1&\&\&\&%
3372 \let\NEXT\measuretrow%
3373 \fi\NEXT}
3374
```

`\edtabcolsep` The length `\edtabcolsep` controls the distance between columns. (was `\abstand`)

```
3375 \newskip\edtabcolsep
3376 \global\edtabcolsep=10pt
3377
```

`\NEXT`

```
\Next 3378 \let\NEXT\relax
3379 \let\Next=\next
```

`\variab`

```
3380 \newcommand{\variab}{\relax}
3381
```

`\l@dcheckcols` Check that the number of columns is consistent. (was `\tabfehlermeldung`)

```
3382 \newcommand*{\l@dcheckcols}{\%
3383 \ifnum\l@dcolcount=1\relax
3384 \else
3385 \ifnum\l@dampcount=1\relax
3386 \else
3387 \ifnum\l@dcolcount=\l@dampcount\relax
3388 \else
3389 \l@d@err@UnequalColumns
3390 \fi
3391 \fi
3392 \l@dampcount=\l@dcolcount
3393 \fi}
3394
```

`\l@dmodforcritext` Modify and restore various macros for when `\critext` is used.

```
\l@drestoreforcritext 3395 \newcommand{\l@dmodforcritext}{\%
3396 \let\critext\relax%
3397 \let\Afootnote\l@dgoobledarg%
3398 \let\Bfootnote\l@dgoobledarg%
3399 \let\Cfootnote\l@dgoobledarg%
```

```

3400 \let\Dfootnote\l@dgoobledarg%
3401 \let\Efootnote\l@dgoobledarg%
3402 \let\edindex\nulledindex%
3403 \let\linenum@gobble}
3404 \newcommand{\l@drestoreforcritext}{%
3405 \def\Afootnote##1##2/{\A@@footnote{##1}{##2}}%
3406 \def\Bfootnote##1##2/{\B@@footnote{##1}{##2}}%
3407 \def\Cfootnote##1##2/{\C@@footnote{##1}{##2}}%
3408 \def\Dfootnote##1##2/{\D@@footnote{##1}{##2}}%
3409 \def\Efootnote##1##2/{\E@@footnote{##1}{##2}}%
3410 \let\edindex\xedndex}
3411

```

\l@dmodforedtext Modify and restore various macros for when \edtext is used.

```

\l@drestoreforedtext 3412 \newcommand{\l@dmodforedtext}{%
3413 \let\edtext\relax
3414 \let\Afootnote\l@dgooblearg
3415 \let\Bfootnote\l@dgooblearg
3416 \let\Cfootnote\l@dgooblearg
3417 \let\Dfootnote\l@dgooblearg
3418 \let\Efootnote\l@dgooblearg
3419 \let\edindex\nulledindex
3420 \let\linenum@gobble}
3421 \newcommand{\l@drestoreforedtext}{%
3422 \def\Afootnote##1{\A@@footnote{##1}}%
3423 \def\Bfootnote##1{\B@@footnote{##1}}%
3424 \def\Cfootnote##1{\C@@footnote{##1}}%
3425 \def\Dfootnote##1{\D@@footnote{##1}}%
3426 \def\Efootnote##1{\E@@footnote{##1}}%
3427 \let\edindex\xedndex}
3428

```

\l@dnullfills Nullify and restore some column fillers, etc.

```

\l@drestorefills 3429 \newcommand{\l@dnullfills}{%
3430 \def\edlabel##1{}%
3431 \def\edrowfill##1##2##3{}%
3432 }
3433 \newcommand{\l@drestorefills}{%
3434 \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
3435 }
3436

```

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.

```

3437 \newcommand{\letsforverteilen}{%
3438 \let\critext\xcritext
3439 \let\edtext\xedtext
3440 \let\edindex\xedindex
3441 \let\Afootnote\A@@footnote
3442 \let\Bfootnote\B@@footnote
3443 \let\Cfootnote\C@@footnote
3444 \let\Dfootnote\D@@footnote
3445 \let\Efootnote\E@@footnote
3446 \let\linenum\@line@num
3447 \hilfsskip=\l@dcolwidth%
3448 \advance\hilfsskip by -\wd\hilfsbox

```

```

3449 \def\edlabel##1{\xedlabel{##1}}
3450

```

`\setmcellright` Typeset (recursively) cells of display math right justified. (was `\rverteilen`)

```

3451 \def\setmcellright #1&{\def\edlabel##1}%
3452 \let\edindex\nulledindex
3453 \ifx #1\\ \ifnum\l@dc@lcount=0\removelastskip
3454 \let\Next\relax%
3455 \else\l@dc@lcount=0%
3456 \let\Next=\setmcellright%
3457 \fi%
3458 \else%
3459 \disablel@dtabfeet%
3460 \step1@dc@lcount%
3461 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
3462 \letsforverteilen%
3463 \hskip\hilfsskip$\displaystyle{#1}$%
3464 \hskip\edtabcolsep%
3465 \let\Next=\setmcellright%
3466 \fi\Next}
3467

```

`\settc@lright` Typeset (recursively) cells of text right justified. (was `\rverteilentext`)

```

3468 \def\settc@lright #1&{\def\edlabel##1}%
3469 \let\edindex\nulledindex
3470 \ifx #1\\ \ifnum\l@dc@lcount=0\removelastskip
3471 \let\Next\relax%
3472 \else\l@dc@lcount=0%
3473 \let\Next=\settc@lright%
3474 \fi%
3475 \else%
3476 \disablel@dtabfeet%
3477 \step1@dc@lcount%
3478 \setbox\hilfsbox=\hbox{#1}%
3479 \letsforverteilen%
3480 \hskip\hilfsskip#1%
3481 \hskip\edtabcolsep%
3482 \let\Next=\settc@lright%
3483 \fi\Next}

```

`\setmcellleft` Typeset (recursively) cells of display math left justified. (was `\lverteilen`)

```

3484 \def\setmcellleft #1&{\def\edlabel##1}%
3485 \let\edindex\nulledindex
3486 \ifx #1\\ \ifnum\l@dc@lcount=0 \let\Next\relax%
3487 \else\l@dc@lcount=0%
3488 \let\Next=\setmcellleft%
3489 \fi%
3490 \else \disablel@dtabfeet%
3491 \step1@dc@lcount%

```

```

3492 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
3493 \letsforverteilen
3494 $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
3495 \let\Next=\setmcellleft%
3496 \fi\Next}
3497

```

`\settcellleft` Typeset (recursively) cells of text left justified. (was `\lverteilentext`)

```

3498 \def\settcellleft #1{\def\edlabel##1{%
3499 \let\edindex\nulledindex
3500 \ifx #1\ \ifnum\l@dc@count=0 \let\Next\relax%
3501 \else\l@dc@count=0%
3502 \let\Next=\settcellleft%
3503 \fi%
3504 \else \disablel@dtabfeet%
3505 \step1@dc@count%
3506 \setbox\hilfsbox=\hbox{#1}%
3507 \letsforverteilen
3508 #1\hskip\hilfsskip\hskip\edtabcolsep%
3509 \let\Next=\settcellleft%
3510 \fi\Next}

```

`\setmcellcenter` Typeset (recursively) cells of display math centered. (was `\zverteilen`)

```

3511 \def\setmcellcenter #1{\def\edlabel##1{%
3512 \let\edindex\nulledindex
3513 \ifx #1\ \ifnum\l@dc@count=0\let\Next\relax%
3514 \else\l@dc@count=0%
3515 \let\Next=\setmcellcenter%
3516 \fi%
3517 \else \disablel@dtabfeet%
3518 \step1@dc@count%
3519 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
3520 \letsforverteilen%
3521 \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
3522 \hskip\edtabcolsep%
3523 \let\Next=\setmcellcenter%
3524 \fi\Next}
3525

```

`\settcellcenter` Typeset (recursively) cells of text centered. (new)

```

3526 \def\settcellcenter #1{\def\edlabel##1{%
3527 \let\edindex\nulledindex
3528 \ifx #1\ \ifnum\l@dc@count=0 \let\Next\relax%
3529 \else\l@dc@count=0%
3530 \let\Next=\settcellcenter%
3531 \fi%
3532 \else \disablel@dtabfeet%
3533 \step1@dc@count%
3534 \setbox\hilfsbox=\hbox{#1}%

```

```

3535 \letsforverteilen%
3536 \hskip 0.5\hlfsskip #1\hskip 0.5\hlfsskip%
3537 \hskip\edtabcolsep%
3538 \let\Next=\settcellcenter%
3539 \fi\Next}
3540

```

\NEXT

```

3541 \let\NEXT=\relax
3542

```

\setmrowright Typeset (recursively) rows of right justified math. (was \rsetzen)

```

3543 \def\setmrowright #1\{\%
3544 \ifx #1& \let\NEXT\relax
3545 \else \centerline{\setmcellright #1&\&\&\&}
3546 \let\NEXT=\setmrowright
3547 \fi\NEXT}

```

\settroright Typeset (recursively) rows of right justified text. (was \rsetzentext)

```

3548 \def\settroright #1\{\%
3549 \ifx #1& \let\NEXT\relax
3550 \else \centerline{\settcclright #1&\&\&\&}
3551 \let\NEXT=\settroright
3552 \fi\NEXT}
3553

```

\setmrowleft Typeset (recursively) rows of left justified math. (was \lsetzen)

```

3554 \def\setmrowleft #1\{\%
3555 \ifx #1& \let\NEXT\relax
3556 \else \centerline{\setmcellleft #1&\&\&\&}
3557 \let\NEXT=\setmrowleft
3558 \fi\NEXT}

```

\settrorleft Typeset (recursively) rows of left justified text. (was \lsetzentext)

```

3559 \def\settrorleft #1\{\%
3560 \ifx #1& \let\NEXT\relax
3561 \else \centerline{\settcclleft #1&\&\&\&}
3562 \let\NEXT=\settrorleft
3563 \fi\NEXT}
3564

```

\setmrowcenter Typeset (recursively) rows of centered math. (was \zsetzen)

```

3565 \def\setmrowcenter #1\{\%
3566 \ifx #1& \let\NEXT\relax%
3567 \else \centerline{\setmcellcenter #1&\&\&\&}
3568 \let\NEXT=\setmrowcenter
3569 \fi\NEXT}

```



`\settrrowcenter` Typeset (recursively) rows of centered text. (new)

```

3570 \def\settrrowcenter #1\{%
3571 \ifx #1& \let\NEXT\relax
3572 \else \centerline{\settrcellcenter #1&\&\&}
3573 \let\NEXT=\settrrowcenter
3574 \fi\NEXT}
3575

```

`\nullsetzen` (was `\nullsetzen`)

```

3576 \newcommand{\nullsetzen}{%
3577 \stepl@dc@colcount%
3578 \l@dc@colwidth=0pt%
3579 \ifnum\l@dc@colcount=30\let\NEXT\relax%
3580 \l@dc@colcount=0\relax
3581 \else\let\NEXT\nullsetzen%
3582 \fi\NEXT}
3583

```

`\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.

```

3584 \newcommand{\edatleft}[3][\@empty]{%
3585 \ifx#1\@empty
3586 \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
3587 depth 0pt \right. $\hss}\vfil}
3588 \else
3589 \vbox to 4pt{\vss\hbox{#1\left#2\vrule width0pt height #3
3590 depth 0pt \right. $\vfil}
3591 \fi}

```

`\edatright` `\edatright[ $\langle math \rangle$ ]{ $\langle symbol \rangle$ }{ $\langle len \rangle$ }` (combination and generalisation of original `\sekla` and `\sekla`). Right  $\langle symbol \rangle$ ,  $2\langle len \rangle$  high with appended  $\langle math \rangle$  vertically centered.

```

3592 \newcommand{\edatright}[3][\@empty]{%
3593 \ifx#1\@empty
3594 \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
3595 depth 0pt \right#2 $\hss}\vfil}
3596 \else
3597 \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
3598 depth 0pt \right#2 #1 $\vfil}
3599 \fi}
3600

```

`\edvertline` `\edvertline{ $\langle len \rangle$ }` vertical line  $\langle len \rangle$  high. (was `\sestrich`)

```

3601 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
3602

```

`\edvertdots` `\edvertdots{ $\langle len \rangle$ }` vertical dotted line  $\langle len \rangle$  high. (was `\sepunkte`)

```

3603 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
3604 {\cleaders\hbox{$\m@th\hbox{.}\vbox to 0.5em{ }$}\vfil}}}
3605

```

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`)

```

3606 \newdimen\edfilldimen
3607 \edfilldimen=0pt
3608

```

`\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...`

```

3609 \newcounter{addcolcount}
3610 \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 reimplement of the  
 original `\@add@`.

```

3611 \newcommand{\l@dtabaddcols}[2]{%

```

```

3612 \l@dccheckstartend{#1}{#2}%
3613 \ifl@dstartendok
3614 \setcounter{addcolcount}{#1}%
3615 \@whilenum \value{addcolcount}<#2\relax \do
3616 {\advance\edfilldimen by \the\csname dcol\theadcolcount\endcsname
3617 \advance\edfilldimen by \edtabcolsep
3618 \stepcounter{addcolcount}}%
3619 \advance\edfilldimen by \the\csname dcol\theadcolcount\endcsname
3620 \fi
3621 }
3622

```

`\ifl@dstartendok` `\l@dccheckstartend{<startcol>}{<endcol>}` checks that the values of `<startcol>` and `<endcol>` are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```

3623 \newif\ifl@dstartendok
3624 \newcommand{\l@dccheckstartend}[2]{%
3625 \l@dstartendoktrue
3626 \ifnum #1<\@ne
3627 \l@dstartendokfalse
3628 \led@err@LowStartColumn
3629 \fi
3630 \ifnum #2>30\relax
3631 \l@dstartendokfalse
3632 \led@err@HighEndColumn
3633 \fi
3634 \ifnum #1>#2\relax
3635 \l@dstartendokfalse
3636 \led@err@ReverseColumns
3637 %%% \ledmac@error{Start column is greater than end column}{\@ehc}%
3638 \fi
3639 }
3640

```

`\edrowfill` `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a LaTeX style reimplementation and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktel` macros.

```

3641 \newcommand*{\edrowfill}[3]{%
3642 \l@dtabaddcols{#1}{#2}%
3643 \hb@xt@ \the\l@dcwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
3644 \let\@edrowfill=\edrowfill
3645 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
3646

```

`\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`)

```
3647 \newcommand{\leftltab}[1]{%
3648 \hb@xt@\z@\vbox{\edtabindent%
3649 \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
3650
```

`\leftrtab` `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`. (was `\linksrtab`)

```
3651 \newcommand{\leftrtab}[2]{%
3652 #2\hb@xt@\z@\vbox{\edtabindent%
3653 \advance\Hilfsskip by\dcoli%
3654 \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
3655
```

`\leftctab` `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`. (was `\linksztab`)

```
3656 \newcommand{\leftctab}[2]{%
3657 \hb@xt@\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
3658 \advance\Hilfsskip by 0.5\dcoli%
3659 \setbox\hilfsbox=\hbox{\def\edlabel##1}%
3660 \disablel@dtabfeet$\displaystyle{#2}$}%
3661 \advance\Hilfsskip by -0.5\wd\hilfsbox%
3662 \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
3663 #2}
3664
```

`\rightctab` `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`. (was `\rechtsztab`)

```
3665 \newcommand{\rightctab}[2]{%
3666 \setbox\hilfsbox=\hbox{\def\edlabel##1}%
3667 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
3668 #1\hb@xt@\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
3669 \advance\Hilfsskip by 0.5\l@dcolwidth%
3670 \advance\Hilfsskip by -\wd\hilfsbox%
3671 \setbox\hilfsbox=\hbox{\def\edlabel##1}%
3672 \disablel@dtabfeet$\displaystyle{#1}$}%
3673 \advance\Hilfsskip by -0.5\wd\hilfsbox%
3674 \advance\Hilfsskip by \edtabcolsep%
3675 \moveright\Hilfsskip\hbox{ #2}}\hss}%
3676 }
3677
```

`\rightltab` `\rightltab{<math>}{<text>}` for `\edaftertab` in `\ltab`. (was `\rechtsltab`)

```
3678 \newcommand{\rightltab}[2]{%
3679 \setbox\hilfsbox=\hbox{\def\edlabel##1}%
3680 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
3681 #1\hb@xt@\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
3682 \advance\Hilfsskip by\l@dcolwidth%
3683 \advance\Hilfsskip by-\wd\hilfsbox%
3684 \setbox\hilfsbox=\hbox{\def\edlabel##1}%
3685 \disablel@dtabfeet$\displaystyle{#1}$}%
3686 \advance\Hilfsskip by-\wd\hilfsbox%
```

```

3687 \advance\Hilfsskip by\edtabcolsep%
3688 \moveright\Hilfsskip\hbox{ #2}\hss}%
3689 }
3690

```

`\righttab` `\righttab{ $\langle math \rangle\langle text \rangle}$`  for `\edaftertab` in `\rtab`. (was `\rechtsrtab`)

```

3691 \newcommand{\righttab}[2]{%
3692 \setbox\hilfsbox=\hbox{\def\edlabel##1{%
3693 \disablel@dtabfeet#2}%
3694 #1\hb@xt@#2{\vbox{\edtabindent%
3695 \advance\Hilfsskip by-\wd\hilfsbox%
3696 \advance\Hilfsskip by\edtabcolsep%
3697 \moveright\Hilfsskip\hbox{ #2}\hss}%
3698 }
3699

```

`\rtab` `\rtab{ $\langle body \rangle}$`  typesets  $\langle body \rangle$  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  $\langle body \rangle$  to get the column widths, and then in a second pass to typeset the body.

```

3700 \newcommand{\rtab}[1]{%
3701 \l@dnnullfills
3702 \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
3703 \def\edaftertab##1##2{\righttab{##1}{##2}}%
3704 \measurembody{#1}%
3705 \l@drestorefills
3706 \variab
3707 \setmrowright #1\&\&\&%
3708 \enablel@dtabfeet}
3709

```

`\measurembody` `\measurembody{ $\langle body \rangle}$`  measures the array  $\langle body \rangle$ .

```

3710 \newcommand{\measurembody}[1]{%
3711 \disablel@dtabfeet%
3712 \l@dcolcount=0%
3713 \nullsetzen%
3714 \l@dcolcount=0
3715 \measuremrow #1\&\&\&%
3716 \global\l@dampcount=1}
3717

```

`\rtabtext` `\rtabtext{ $\langle body \rangle}$`  typesets  $\langle body \rangle$  as a tabular with the entries right justified. (was `\rtabtext`)

```

3718 \newcommand{\rtabtext}[1]{%
3719 \l@dnnullfills
3720 \measuretbody{#1}%
3721 \l@drestorefills

```

```

3722 \variab
3723 \setthrowright #1\\&\\%
3724 \enablel@dtabfeet}
3725

```

`\measuretbody` `\measuretbody{<body>}` measures the tabular `<body>`.

```

3726 \newcommand{\measuretbody}[1]{%
3727 \disablel@dtabfeet%
3728 \l@dcolcount=0%
3729 \nullsetzen%
3730 \l@dcolcount=0
3731 \measuretbody #1\\&\\%
3732 \global\l@dampcount=1}
3733

```

`\ltab` Array with entries left justified. (was `\ltab`)

```

\edbeforetab 3734 \newcommand{\ltab}[1]{%
\edaftertab 3735 \l@dnnullfills
3736 \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
3737 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
3738 \measuretbody{#1}%
3739 \l@drestorefills
3740 \variab
3741 \setmrowleft #1\\&\\%
3742 \enablel@dtabfeet}
3743

```

`\ltabtext` Tabular with entries left justified. (was `\ltabtext`)

```

3744 \newcommand{\ltabtext}[1]{%
3745 \l@dnnullfills
3746 \measuretbody{#1}%
3747 \l@drestorefills
3748 \variab
3749 \setmrowleft #1\\&\\%
3750 \enablel@dtabfeet}
3751

```

`\ctab` Array with centered entries. (was `\ztab`)

```

\edbeforetab 3752 \newcommand{\ctab}[1]{%
\edaftertab 3753 \l@dnnullfills
3754 \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
3755 \def\edaftertab##1##2{\rightctab{##1}{##2}}%
3756 \measuretbody{#1}%
3757 \l@drestorefills
3758 \variab
3759 \setmrowcenter #1\\&\\%
3760 \enablel@dtabfeet}
3761

```

`\ctabtext` Tabular with entries centered. (new)

```
3762 \newcommand{\ctabtext}[1]{%
3763 \l@dnulldfills
3764 \measuretbody{#1}%
3765 \l@drestorefills
3766 \variab
3767 \settrrowcenter #1\\&\\%
3768 \enablel@dtabfeet}
3769
```

`\spreadtext` (was `\breitertext`)

```
3770 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
3771 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
```

`\spreadmath` (was `\breiter`, 'breiter' = 'broadly')

```
3772 \newcommand{\spreadmath}[1]{%
3773 \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
3774
```

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`)

```
3775 \def\tabellzwischen #1{%
3776 \ifx #1\ \let\NEXT\relax \l@dcolcount=0
3777 \else \stepl@dcolcount%
3778 \l@dcolwidth = #1 mm
3779 \let\NEXT=\tabellzwischen
3780 \fi \NEXT }
3781
```

`\edatabell` For example `\edatabell 4 & 19 & 8 \\` specifies 3 columns with widths of 4, 19, and 8mm. (was `\atabell`)

```
3782 \def\edatabell #1\\{%
3783 \tabellzwischen #1&\\&}
```

`\Setzen` (was `\Setzen`, 'setzen' = 'set')

```
3784 \def\Setzen #1{%
3785 \ifx #1\relax \let\NEXT=\relax
3786 \else \stepl@dcolcount%
3787 \let\tabelskip=\l@dcolwidth
3788 \EDTAB #1|
3789 \let\NEXT=\Setzen
3790 \fi \NEXT}
3791
```

\EDATAB (was \ATAB)

```
3792 \def\EDATAB #1\{%
3793 \ifx #1\Relax \centerline{\Setzen #1\relax&}
3794 \let\Next\relax
3795 \else \centerline{\Setzen #1&\relax&}
3796 \let\Next=\EDATAB
3797 \fi\Next}
```

\edatab (was \atab)

```
3798 \newcommand{\edatab}[1]{%
3799 \variab%
3800 \EDATAB #1\\Relax\\}
3801
```

\HILFSskip More helpers.

```
\Hilfsskip 3802 \newskip\HILFSskip
3803 \newskip\Hilfsskip
3804
```

\EDTABINDENT (was \TABINDENT)

```
3805 \newcommand{\EDTABINDENT}{%
3806 \ifnum\l@dc@lcount=30\let\NEXT\relax\l@dc@lcount=0%
3807 \else\step\l@dc@lcount%
3808 \advance\Hilfsskip by\l@dc@lwidth%
3809 \ifdim\l@dc@lwidth=0pt\advance\hilfscout\@ne
3810 \else\advance\Hilfsskip by \the\hilfscout\edtabcolsep%
3811 \hilfscout=1\fi%
3812 \let\NEXT=\EDTABINDENT%
3813 \fi\NEXT}%

```

\edtabindent (was \tabindent)

```
3814 \newcommand{\edtabindent}{%
3815 \l@dc@lcount=0\relax
3816 \Hilfsskip=0pt%
3817 \hilfscout=1\relax
3818 \EDTABINDENT%
3819 \hilfsskip=\hsize%
3820 \advance\hilfsskip -\Hilfsskip%
3821 \Hilfsskip=0.5\hilfsskip%
3822 }%
3823
```

\EDTAB (was \TAB)

```
3824 \def\EDTAB #1|#2|{%
3825 \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
3826 \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
3827 \advance\tabelskip -\wd\tabhilfbox%
3828 \advance\tabelskip -\wd\tabHilfbox%
3829 \unhbox\tabhilfbox\hskip\tabelskip%
```



```
3830 \unhbox\tabHilfbox}%
3831
```

\EDTABtext (was \TABtext)

```
3832 \def\EDTABtext #1|#2|{%
3833 \setbox\tabhilfbox=\hbox{#1}%
3834 \setbox\tabHilfbox=\hbox{#2}%
3835 \advance\tabelskip -\wd\tabhilfbox%
3836 \advance\tabelskip -\wd\tabHilfbox%
3837 \unhbox\tabhilfbox\hskip\tabelskip%
3838 \unhbox\tabHilfbox}%
```

\tabhilfbox Further helpers.

```
\tabHilfbox 3839 \newbox\tabhilfbox
3840 \newbox\tabHilfbox
3841
```

```
%%
% That finishes tabmac
%%
```

edarrayl The 'environment' forms for \ltab, \ctab and \rtab.

```
edarrayc 3842 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{}
edarrayr 3843 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{}
3844 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{}
3845
```

edtabularl The 'environment' forms for \ltabtext, \ctabtext and \rtabtext.

```
edtabularc 3846 \newenvironment{edtabularl}{\l@dcollect@body\ltabtext}{}
edtabularr 3847 \newenvironment{edtabularc}{\l@dcollect@body\ctabtext}{}
3848 \newenvironment{edtabularr}{\l@dcollect@body\rtabtext}{}
3849
```

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 3850 \newcommand{\usingcritext}{%
\usingedtext 3851 \def\disablel@dtabfeet{\l@dmodforcritext}%
3852 \def\enablel@dtabfeet{\l@drestoreforcritext}}
3853 \newcommand{\usingedtext}{%
3854 \def\disablel@dtabfeet{\l@dmodforedtext}%
3855 \def\enablel@dtabfeet{\l@drestoreforedtext}}
3856
3857 \usingedtext
3858
```

## 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 :

```
\newcommand*{\Afootmt}[4][4]{<code>}
```

## B Examples

### B.1 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.

This section presents some sample documents.

The examples in sections B.3 through B.6, plus B.8, 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:: eximples 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 possest 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!] *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`.



|                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
|                            | Incipit Quartus ΠΕΡΙΦΥΣΕΩΝ                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 741C          |
| ΑΝΑΚΕΦΑΛΙΟΣΙΣ<br>NATVRARVM | NVTRITOR. Prima nostrae Physiologiae intentio praecipuaque materia erat quod ΥΠΕΡΟΥΣΙΑΔΕΣ (hoc est superessentialis) natura sit causa creatrix existentium et non existentium omnium, a nullo creata, unum principium, una origo, unus et uniuersalis uniuersorum fons, a nullo manans, dum ab eo manant omnia, trinitas coessentialis in tribus substantiis, ΑΝΑΡΧΟΣ (hoc est sine principio), principium et finis, una bonitas, deus unus, ΟΜΟΥΣΙΟΣ et ΥΠΕΡΟΥΣΙΟΣ (id est coessentialis et superessentialis). Et, ut ait sanctus Epifanius, episcopus Constantiae Cypri, in ΑΓΚΥΡΑΤΩ sermone de fide: <i>Tria sancta, tria consancta, tria agentia, tria coagentia, tria formantia, tria conformantia, tria operantia, tria cooperantia, tria subsistentia, tria consubsistentia sibi inuicem coexistentia. Trinitas haec sancta uocatur: tria existentia, una consonantia, una deitas eiusdem essentiae, eiusdem uirtutis, eiusdem subsistentiae, similia similiter aequalitatem gratiae operantur patris et filii et sancti spiritus. Quomodo autem sunt, ipsis relinquitur docere: ‘Nemo enim nouit patrem nisi filius, neque filium nisi pater, et cuicumque filius reuelauerit’; reuelatur autem per spiritum sanctum. Non ergo haec tria existentia aut ex ipso aut per ipsum aut ad ipsum in unoquoque digne intelliguntur,  R, 264<sup>f</sup>  sicut ipsa reuelant: ΦΩΣ, ΠΥΡ, ΠΝΕΥΜΑ (hoc est lux, ignis, spiritus).</i> | 5<br>10<br>15 |
|                            | Haec, ut dixi, ab Epifanio tradita, ut quisquis interrogatus quae tria et quid unum in sancta trinitate debeat credere, sana fide  J, 1 <sup>v</sup>   respondere ualeat, aut ad fidem accedens sic erudiatur. Et mihi uidetur spiritum pro calore posuisse, quasi dixisset in similitudine: lux, ignis, calor. Haec enim tria unius essentiae sunt. Sed cur lucem primo dixit, non est mirum. Nam et pater lux est et ignis et calor; et filius est lux, ignis, calor; et spiritus sanctus lux, ignis, calor. Illuminat enim pater, illuminat filius, illuminat spiritus sanctus: ex ipsis enim omnis scientia et sapientia donatur.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 20<br>25      |

15–16 Matth. 11, 27 19 EPIPHANIVS, *Ancoratus* 67; PG 43, 137C–140A; GCS 25, p. 82, 2–12

1 incipit . . . ΠΕΡΙΦΥΣΕΩΝ] *om. R, incipit quartus M* 2 ΑΝΑΚΕΦΑΛΙΟΣΙΣ] *FJP, lege ἀνακεφαλαιώσεις* 2 physiologiae] *phisiologiae P, physeologiae R* 3 quod] *p. natura transp. MR* 3 ΥΠΕΡΟΥΣΙΑΔΕΣ] *codd. Vtrum ὑπερουσιώδης (hoc est superessentialis) natura cum Gale (p.160) an ὑπερουσιότης (hoc est superessentialis natura) cum Floss (PL 122,741C) intelligendum sit, ambigitur* 7 ΟΜΟΥΣΙΟΣ] *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] *substantiae 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  
Josephá?  
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 phleidl 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 straothaig 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.2 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.

---

```

3859 (*easy)
3860 % ledeasy.tex simple example of the ledmac package
3861 \documentclass{article}
3862 \usepackage{ledmac}
3863 %% number every line
3864 \setcounter{firstlinenum}{1}
3865 \setcounter{linenumincrement}{1}
3866 %% Show some B series familiar footnotes, lettered and paragraphed
3867 \renewcommand*{\thefootnoteB}{\alph{footnoteB}}
3868 \footparagraphX{B}
3869 %% no endnotes
3870 \noendnotes
3871 %% narrow sidenotes
3872 \setlength{\ledrsnotewidth}{4em}
3873 \title{Simple Example}
3874 \author{Peter Wilson\thanks{Standing on the shoulders of giants.}}
3875 \date{}
3876 \begin{document}
3877 \maketitle
3878 \tableofcontents
3879 \section{First}
3880 This is a simple example of using the \textsf{ledmac}
3881 package with ordinary LaTeX constructs.
3882
3883 \subsection{Example text}\label{subsec}
3884
3885 \beginnumbering
3886 \pstart
3887 The \textsf{ledmac} package lets you do some unusual things in
3888 a LaTeX document. For example you can have lines numbered and
3889 there are
3890 \edtext{several}{\Afootnote{This is an ‘A’ footnote.}}
3891 \edtext{levels}{\Bfootnote{This is a ‘B’ level footnote.}}
3892 of footnotes.
3893 You can label lines within the numbered text and refer to them
3894 outside. Do not try and use any normal LaTeX
3895 \marginpars\footnote{You will get a warning but no text.}%
3896 \ledrightrightnote{Sidenotes are OK}
3897 or \edtext{exotica}{\Afootnote{Like floats.}}
3898 within the numbered portions of the text\edlabel{line}.
3899 \pend
3900 \endnumbering
3901
3902 \section{Last}

```

```

3903
3904 I forgot to mention that you can use ordinary
3905 footnotes\footnote{An ordinary footnote}\footnote{And another}
3906 outside the numbered text. You can also\footnoteB{Additionally}
3907 have\footnoteB{Specify} formatted footnotes\footnoteB{Like this}
3908 in normal\footnoteB{Text that does not have line numbers} text.
3909
3910 There are \lineref{line} numbered lines in the example shown
3911 in section~\ref{subsec}.
3912
3913 \end{document}
3914 </easy>

```

### B.3 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.

---

```

3915 (*features)
3916 % ledfeat.tex Small test file for ledmac package
3917 \documentclass{article}
3918 \usepackage{ledmac}
3919
3920 \noendnotes % we aren't having any endnotes
3921
3922 \makeatletter
3923 % I'd like a spaced out colon after the lemma:
3924 \newcommand{\spacedcolon}{\rmfamily\thinspace:\thinspace}
3925 \renewcommand*{\normalfootfmt}[3]{%
3926 \ledsetnormalparstuff
3927 {\notenumfont\printlines#1|}\strut\enspace
3928 {\select@lemmafont#1/#2}\spacedcolon\enskip#3\strut\par}
3929
3930 % And I'd like the 3-col notes printed with a hanging indent:
3931 \renewcommand*{\threecolfootfmt}[3]{%
3932 \normal@pars
3933 \hsize .3\hsize
3934 \setlength{\parindent}{0pt}
3935 \tolerance=5000 % high, but not infinite
3936 \raggedright
3937 \hangindent1.5em \hangafter1
3938 \leavevmode
3939 \strut\hbox to 1.5em{\notenumfont\printlines#1|\hfil}\ignorespaces
3940 {\select@lemmafont#1/#2}\rbracket\enskip
3941 #3\strut\par\allowbreak}
3942

```

```

3943 % And I'd like the 2-col notes printed with a double colon:
3944 \newcommand*{\doublecolon}{\rmfamily\thinspace:\thinspace}}
3945 \renewcommand*{\twocolfootfmt}[3]{%
3946 \normal@pars
3947 \hsize .45\hsize
3948 \setlength{\parindent}{0pt}
3949 \tolerance=5000
3950 \raggedright
3951 \leavevmode
3952 \strut{\notenumfont\printlines#1/}\enspace
3953 {\select@lemmafont#1/#2}\doublecolon\enskip
3954 #3\strut\par\allowbreak}
3955
3956 % And in the paragraphed footnotes, I'd like a colon too:
3957 \renewcommand*{\parafootfmt}[3]{%
3958 \ledsetnormalparstuff
3959 {\notenumfont\printlines#1/}\enspace
3960 {\select@lemmafont#1/#2}\spacedcolon\enskip
3961 #3\penalty-10 }
3962 \makeatother
3963
3964 % I'd like the line numbers picked out in bold.
3965 \renewcommand{\notenumfont}{\bfseries}
3966 \lineation{page}
3967 \linenummargin{inner}
3968 \setcounter{firstlinenum}{3} % just because I can
3969 \setcounter{linenumincrement}{1}
3970 \foottwocol{A}
3971 \footthreecol{B}
3972 \footparagraph{E}
3973 % I've changed \normalfootfmt, so invoke it again for C and D notes.
3974 \footnormal{C}
3975 \footnormal{D}
3976
3977 \begin{document}
3978
3979 \begin{numbering}
3980
3981 \pstart
3982 This is an \edtext{example}{
3983 \Afootnote{eximemple C, D.}}
3984 of some %\footnote{A normal footnote}
3985 text with \edtext{variant}{
3986 \Afootnote{alternative, A, B.}}
3987 readings recorded as 'A' footnotes. From here on, \edtext{though}{
3988 \Afootnote{however α, β}},
3989 we shall have \edtext{'C'}{
3990 \Bfootnote{B, \textit{pace} the text}}.
3991 \edtext{For spice, let us mark a longer passage, but give a different
3992 lemma for it, so that we don't get a \edtext{huge}{

```

```

3993 \Dfootnote{vast E, F; note that this is
3994 a 'D' note to section of text within a longer lemma}}
3995 amount of text in a note}{\lemma{For spice \dots\ note}
3996 \Cfootnote{The note here is type 'C'}}.
3997 \edtext{Finally}{
3998 \Efootnote{in the end X, Y}},
3999 \edtext{we}{
4000 \Efootnote{us K}}
4001 \edtext{shouldn't}{
4002 \Efootnote{ought not to L, M}}
4003 \edtext{forget the}{
4004 \Efootnote{omit to mention the \S, \P}}
4005 \edtext{paragraphed}{
4006 \Efootnote{blocked M, N}}
4007 \edtext{notes}{
4008 \Efootnote{variants HH, KK}},
4009 which are so \edtext{useful}{
4010 \Efootnote{truly useful L, P}}
4011 when there are \edtext{a great number of}{
4012 \Efootnote{many, many (preferably)}}
4013 short notes to be \edtext{recorded}{
4014 \Efootnote{noted: repetition}}.
4015 \pend
4016
4017 \pstart
4018 This is a second paragraph, giving more \textit{\edtext{examples}}{
4019 \Afootnote{eximples L, M.}}}
4020 of text with \edtext{variant}{
4021 \Afootnote{alternative, A, B.}}
4022 readings recorded as 'A' footnotes. From here on, \edtext{though}{
4023 \Bfootnote{however α, β}},
4024 we shall have \edtext{'B'}{
4025 \Bfootnote{B, as correctly stated in the text}} notes in the text.
4026 \edtext{For spice, let us mark a longer passage, but give a different
4027 lemma for it, so that we don't get a \textit{\edtext{huge}}{
4028 \Dfootnote{vast E, F; note that this is
4029 a 'D' note to text within a longer lemma.}}}
4030 amount of text in a note}{\lemma{For spice, \dots\ note}
4031 \Cfootnote{This is a rogue note of type 'C'.}}.
4032 \edtext{Finally}{
4033 \Bfootnote{In the end X, Y}},
4034 \edtext{we}{
4035 \Bfootnote{we here K}}
4036 \edtext{shouldn't}{
4037 \Bfootnote{ought not to L, M}}
4038 \edtext{forget the}{
4039 \Bfootnote{omit to mention the \S, \P}}
4040 \edtext{column}{
4041 \Bfootnote{blocked M, N}}
4042 \edtext{notes}{

```

```

4043 \Bfootnote{variants H}},
4044 which are so \edtext{useful}{
4045 \Bfootnote{very, very useful L, P}}
4046 when there are \edtext{many}{
4047 \Bfootnote{lots of Z}}
4048 short notes to be \edtext{recorded}{
4049 \Bfootnote{recorded and put down: M (repetition)}}.
4050 \pend
4051
4052 \endnumbering
4053 \end{document}
4054 </features>

```

---

## B.4 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..

---

```

4055 (*ioc)
4056 %% ledioc.tex
4057 \documentclass{article}
4058 \usepackage{ledmac}
4059
4060 \noendnotes
4061 \makeatletter
4062
4063 \newcommand{\os}{\scriptsize}
4064 \setcounter{firstsublinenum}{1000}
4065 \frenchspacing \setlength{\parskip}{0pt} \hyphenpenalty=1000
4066
4067 % Say \nolinenums if you want no line numbers in the notes.
4068 \newif\ifnolinenums
4069 \newcommand{\nolinenums}{\global\nolinenumstrue}
4070 \newcommand{\linenums}{\global\nolinenumsfalse}
4071
4072 \renewcommand{\rightlinenum}{\ifbypage@\ifnum\line@num<10\kern.5em\fi\else
4073 \ifnum\line@num<10\kern1em\else\ifnum\line@num<100
4074 \kern.5em\fi\fi\fi\kern.5em\numlabfont\the\line@num
4075 \ifnum\subline@num>0:\the\subline@num\fi}
4076
4077 \renewcommand{\leftlinenum}{\numlabfont\the\line@num
4078 \ifnum\subline@num>0:\the\subline@num\fi \kern.5em}
4079 \linenummargin{outer}
4080 \lineation{page}

```



```

4081
4082 \newcommand{\ggfootfmt}[3]{%
4083 \notefontsetup
4084 \let\par=\endgraf
4085 \rightskip=0pt \leftskip=0pt
4086 \setlength{\parindent}{0pt} \parfillskip=0pt plus 1fil
4087 \ifnolinenums\relax\else
4088 \begingroup \os \printlines#1/\endgroup
4089 \enskip
4090 \fi
4091 {\rmfamily #2\def\@tempa{#2}\ifx\@tempa\empty
4092 \else\enskip\fi#3\penalty-10 }}
4093
4094 % Now reset the \Afootnote parameters and macros:
4095 \footparagraph{A}
4096 \let\Afootfmt=\ggfootfmt
4097 \dimen\Afootins=\vsize
4098 \skip\Afootins=3pt plus9pt
4099 \newcommand*{\ggfootstart}[1]{\vskip\skip\Afootins}
4100 \let\Afootstart=\ggfootstart
4101
4102 \newcommand*{\stage}[1]{\pstart\startsub\parindent=0pt
4103 \hangindent=3em\hangafter=0
4104 {\itshape #1}\let\par=\finishstage}
4105 \newcommand{\finishstage}{\pend\endsub}
4106 \newcommand{\sen}{\leavevmode\lower1ex\hbox{\textrm{'}}}%
4107 \newcommand{\senspeak}[1]{\pstart\obeylines\setbox0=\hbox{\textrm{'}}}%
4108 \leavevmode
4109 \lower1ex\copy0\kern-\wd0\hskip1em{\textit{#1}}}%
4110 \hbox tolex{\ignorespaces}
4111 \newcommand*{\speak}[1]{\pstart\obeylines\hskip1em{\textit{#1}}}%
4112 \hbox tolex{\ignorespaces}
4113 \def\nospeaker{\parindent=0em\pstart\let\par=\pend}
4114 \newcommand*{\nospeak}{\pstart\obeylines}
4115 \makeatother
4116
4117 \begin{document}
4118
4119 \setlength{\parindent}{0pt}
4120
4121 \beginnumbering
4122
4123 \stage{Oedipus \edtext{entreth}{\Afootnote{\textit{intrat} MS}}.}
4124
4125 \nospeak
4126 Or that with wrong the right and doubtlesse heire,
4127 Shoulde banisht be out of his princely seate.
4128 Yet thou O queene, so fyle thy sugred tounge,
4129 And with suche counsell decke thy mothers tale,
4130 That peace may bothe the brothers heartes inflame,

```

```

4131 And rancour yelde, that erst possesse the same.
4132 \pend
4133
4134 \speak{Eteocl.} Mother, beholde, youre hestes for to obey,
4135 In person nowe am I resorted hither:
4136 In haste therefore, fayne woulde I knowe what cause
4137 With hastie speede, so moued hath your mynde
4138 To call me nowe so causelesse out of tyme,
4139 When common wealth moste craues my onely ayde:
4140 Fayne woulde I knowe, what queynt commoditie
4141 Persuades you thus to take a truce for tyme,
4142 And yelde the gates wide open to my foe,
4143 The gates that myght our stately state defende,
4144 And nowe are made the path of our decay.
4145 \pend
4146
4147 \sensspeak{Ioca.} Represse deare son, those raging stormes of wrath,
4148 \sen That so bedimme the eyes of thine intende,
4149 \edtext{\sen As when \edtext{the}{\Afootnote{thie MS}} tongue %
4150 (a redy Instrument)
4151 \sen Would \edtext{fayne pronounce}{\Afootnote{faynest tell MS}} %
4152 the meaning of \edtext{the minde}{\Afootnote{thy minde MS}},
4153 \sen \edtext{It}{\lemma{It \dots\ worde.}}\Afootnote{Thie %
4154 swelling hart puft vp with wicked ire / Can scarce pronounce %
4155 one inward louing thought. MS}} cannot speake one honest %
4156 seemely worde.}{\lemma{As \dots\ worde.}}\Afootnote{\textit{not %
4157 in} \os73}}
4158 \sen But when disdayne is shrunke, or sette asyde,
4159 \sen And mynde of man with leysure can discourse
4160 \sen What seemely wordes his tale may best beseeme,
4161 \sen And that the tounge vnfoldes without affectes
4162 \sen Then may proceede an answere sage and graue,
4163 \sen And euery sentence sawst with sobernesse:
4164 Wherefore vnbende thyne angrie browes deare chylde,
4165 And caste thy rolling eyes none other waye,
4166 That here doost not \edtext{\textit{Medusaes}}{%
4167 \Afootnote{One of the furies. {\os75}m}} face beholde,
4168 But him, euen him, thy blood and brother deare.
4169 And thou beholde, my \textit{Polinices} eke,
4170 Thy brothers face, wherein when thou mayst see
4171 Thine owne image, remember therewithall,
4172 That what offence thou woldst to him were done,
4173 \pend
4174 \endnumbering
4175
4176 \end{document}
4177
4178 \end{document}

```

---

## B.5 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.

---

```

4179 (*arden)
4180 %% ledarden.tex
4181 \documentclass{article}
4182 \usepackage{ledmac}
4183
4184 \makeatletter
4185 \newcommand{\stage}[1]{\rlap{\hbox to \the\linenumsep{%
4186 \hfil\llap{[\textit{#1}]}}}
4187
4188 \newcommand{\speaker}[1]{\pstart\hangindent2em\hangafter1
4189 \leavevmode\textit{#1}\enspace\ignorespaces}
4190
4191 \newcommand{\exit}[1]{\hfill\stage{#1}}
4192
4193 % LEDMAC customizations:
4194 \noendnotes
4195 \setlength{\parindent}{0pt}
4196 \setlength{\linenumsep}{.4in}
4197 \rightskip\linenumsep
4198
4199 \renewcommand{\interparanoteglue}{1em plus.5em minus.1em}
4200
4201 \newcommand{\scf}{\tiny}
4202 \let\Afootnoterule=\relax \let\Bfootnoterule=\relax
4203
4204 \renewcommand{\rightlinenum}{\numlabfont\llap{\the\line@num}}
4205 \frenchspacing
4206
4207 % Footnote formats:
4208 % \nonumparafootfmt is a footnote format without line numbers.
4209 \newcommand{\nonumparafootfmt}[3]{%
4210 \ledsetnormalparstuff
4211 \rightskip=0pt
4212 \select@lemmafnt#1/#2\rbracket\enskip
4213 \itshape #3\penalty-10 }
4214
4215 \newcommand{\newparafootfmt}[3]{%
4216 \ledsetnormalparstuff
4217 {\notenumfont\printlines#1/}\fullstop\enspace
4218 {\select@lemmafnt#1/#2\rbracket\enskip
4219 \itshape #3\penalty-10 }
4220

```

```

4221 \newcommand{\newtwocolfootfmt}[3]{%
4222 \normal@pars
4223 \hsize .48\hsize
4224 \tolerance=5000
4225 \rightskip=0pt \leftskip=0pt \parindent=5pt
4226 \strut\notenumfont\printlines#1/\fullstop\enspace
4227 \itshape #2/\rbracket\penalty100\hskip .5em plus .5em
4228 \normalfont #3\strut\goodbreak}
4229
4230 % Footnote style selections etc. (done last):
4231 \footparagraph{A}
4232 \foottwocol{B}
4233 \let\Afootfmt=\newparafootfmt
4234 \let\Bfootfmt=\newtwocolfootfmt
4235 \let\collation=\Afootnote
4236 \let\note=\Bfootnote
4237 \lineation{section}
4238 \linenummargin{right}
4239 \makeatother
4240
4241 %%%
4242
4243 \begin{document}
4244 \pagestyle{empty}
4245
4246 % Initially, we don't want line numbers.
4247 \let\Afootfmt=\nonumparafootfmt
4248
4249 \beginnumbering
4250 \pstart
4251 \centerline{[\edtext{SCENE III}{
4252 \lemma{Scene III}
4253 \collation{Capell; om. Q, F; \textnormal{Scene IV} Pope.}}.---%
4254 \edtext{\textit{Venice}}}{
4255 \collation{om. Q, F; Shylock's house Theobald; The same.
4256 A Room in Shylock's House Capell.}}.]}
4257 \pend
4258 \bigskip
4259
4260 \pstart
4261 \centerline{\textit{Enter} JESSICA \textit{and}
4262 [\edtext{LAUNCELOT}{
4263 \lemma{Launcelot}
4264 \collation{Rowe; om. Q, F.}}] \textit{the clown.}} \pend \bigskip
4265
4266 \let\Afootfmt=\newparafootfmt % we do want line numbers from now
4267
4268 \setline{0}%
4269
4270 \speaker{Jes.}\edtext{I am}{

```

```

4271 \collation{Q, F; \textnormal{I'm} Pope.}}
4272 sorry thou wilt leave my father so,\
4273 Our house is hell, and thou (a merry devil)\
4274 Didst rob it of some taste of tediousness,---\
4275 But fare thee well, there is a ducat for thee,\
4276 And Launcelot, \edtext{soon}{
4277 \note{early.}}
4278 at supper shalt thou see\
4279 Lorenzo, who is thy new master's guest,\
4280 Give him this letter,---do it secretly,---\
4281 And so farewell: I would not have my father\
4282 See me \edtext{in}{
4283 \collation{Q; om. F.}}
4284 talk with thee.
4285 \pend
4286
4287 \speaker{Laun.}
4288 \edtext{}{\lemma{\textit{Laun.}}\collation{Q2; Clowne. Q, F.}}%
4289 \edtext{Adieu!}{
4290 \collation{\textnormal{Adiew}, Q, F.}}
4291 tears \edtext{exhibit}{
4292 \note{Eccles paraphrased "My tears serve to express what my
4293 tongue should, if sorrow would permit it," but probably it is
4294 Launce-\lot's blunder for prohibit (Halliwell) or inhibit
4295 (Clarendon).}}
4296 my tongue, most beautiful \edtext{pagan}{
4297 \note{This may have a scurrilous undertone: cf. \textit{2 H 4,}
4298 {\scf II.} \textrm{iii. 168.}}}%
4299 , most sweet \edtext{Jew!}{
4300 \collation{\textnormal{Iewe}, Q, F. \quad \textnormal{do}} Q, F;
4301 \textnormal{did} F2.}}%
4302 ---if a Christian \edtext{do}{
4303 \note{Malone upheld the reading of Qq and F by comparing {\scf II.}
4304 vi. 23: "When you shall please to play the thieves for
4305 wives"; Launcelot seems fond of hinting at what is going to
4306 happen (cf. {\scf II.} v. 22--3). If F2's "did" is accepted,
4307 \textit{get} is used for beget, as in {\scf III.} v. 9.}}
4308 not play the knave and get thee, I am much deceived; but \edtext{adieu!}{
4309 \collation{\textnormal{adiew}, Q, F.}}
4310 these \edtext{foolish drops do \edtext{something}{
4311 \collation{Q; \textnormal{somewhat} F.}}
4312 drown my manly spirit}{
4313 \lemma{foolish\textnormal{\dots}spirit}
4314 \note{"tears do not become a man" (\textit{AYL.}, {\scf III.}
4315 iv. 3); cf. also \textit{H 5,} {\scf IV.} vi. 28--32.}}%
4316 : \edtext{adieu!}{
4317 \collation{\textnormal{adiew}. Q, F. \quad \textnormal{S. D.}} Q2, F; om. Q;
4318 after l. 15 Capell.}}
4319 \exit{Exit.}
4320 \pend

```

```

4321
4322 \speaker{Jes.}
4323 Farewell good Launcelot.\
4324 Alack, what heinous sin is it in me\
4325 To be ashamed to be my father's \edtext{child!}{
4326 \collation{\textnormal{child}, Q, F; \textnormal{Child?} Rowe.}}
4327 \pend
4328 \endnumbering
4329
4330 \end{document}
4331
4332 \</arden>

```

---

## B.6 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.

---

```

4333 \<periph>
4334 % ledmixed.tex
4335 \documentclass{article}
4336 \usepackage{ledmac}
4337
4338 \noendnotes
4339 %% \overfullrule0 pt
4340 \lefthyphenmin=3
4341

```

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.

```

4342 \let\Ma=$
4343 \let\aM=$
4344 \usepackage[delims]{lgreek}
4345
4346 % 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.

```

4347 % causes problems:
4348 \newcommand{\morenoexpands}{\let$=0}
4349
4350 \makeatletter
4351
4352 \newbox\lp@rbox
4353
4354 \newcommand{\ffootnote}[1]{%
4355 \ifnumberedpar@
4356 \xright@appenditem{\noexpand\vffootnote{f}}{\l@d@nums}{\csexpandonce{@tag}}{\csexpandonce{content}}
4357 \to\inserts@list
4358 \global\advance\insert@count by 1
4359 % \else %% may be used only in numbered text
4360 % \vffootnote{f}{\{0|0|0|0|0|0|0\}{\#1}}%
4361 \fi\ignorespaces}
4362
4363 \newcommand{\gfootnote}[1]{%
4364 \ifnumberedpar@
4365 \xright@appenditem{\noexpand\vgfootnote{g}{\#1}}%
4366 \to\inserts@list
4367 \global\advance\insert@count by 1
4368 % \else %% may be used only in numbered text
4369 % \vgfootnote{g}{\#1}%
4370 \fi\ignorespaces}
4371
4372 \newcommand{\setlp@rbox}[3]{%
4373 {\parindent\z@\hspace=2.5cm\raggedleft\scriptsize
4374 \baselineskip 9pt%
4375 \global\setbox\lp@rbox=\vbox to\z@{\vss#3}}}
4376
4377 \newcommand{\vffootnote}[2]{\setlp@rbox#2}
4378
4379 \newcommand{\vgfootnote}[2]{\def\rd@ta{#2}}
4380
4381
4382
4383 \renewcommand{\affixline@num}{%
4384 \ifsublines@
4385 \@l@tempcntb=\subline@num
4386 \ifnum\subline@num>\c@firstsublinenum
4387 \@l@tempcnta=\subline@num
4388 \advance\@l@tempcnta by-\c@firstsublinenum
4389 \divide\@l@tempcnta by\c@sublinenumincrement
4390 \multiply\@l@tempcnta by\c@sublinenumincrement
4391 \advance\@l@tempcnta by\c@firstsublinenum
4392 \else
4393 \@l@tempcnta=\c@firstsublinenum
4394 \fi
4395 %
4396 \ifcase\sub@lock

```

```

4397 \or
4398 \ifnum\sublock@disp=1
4399 \@l@dttempcntb=0 \@l@dttempcnta=1
4400 \fi
4401 \or
4402 \ifnum\sublock@disp=2 \else
4403 \@l@dttempcntb=0 \@l@dttempcnta=1
4404 \fi
4405 \or
4406 \ifnum\sublock@disp=0
4407 \@l@dttempcntb=0 \@l@dttempcnta=1
4408 \fi
4409 \fi
4410 \else
4411 \@l@dttempcntb=\line@num
4412 \ifnum\line@num>\c@firstlinenum
4413 \@l@dttempcnta=\line@num
4414 \advance\@l@dttempcnta by-\c@firstlinenum
4415 \divide\@l@dttempcnta by\c@linenumincrement
4416 \multiply\@l@dttempcnta by\c@linenumincrement
4417 \advance\@l@dttempcnta by\c@firstlinenum
4418 \else
4419 \@l@dttempcnta=\c@firstlinenum
4420 \fi
4421 \ifcase\@lock
4422 \or
4423 \ifnum\lock@disp=1
4424 \@l@dttempcntb=0 \@l@dttempcnta=1
4425 \fi
4426 \or
4427 \ifnum\lock@disp=2 \else
4428 \@l@dttempcntb=0 \@l@dttempcnta=1
4429 \fi
4430 \or
4431 \ifnum\lock@disp=0
4432 \@l@dttempcntb=0 \@l@dttempcnta=1
4433 \fi
4434 \fi
4435 \fi
4436 %
4437 \ifnum\@l@dttempcnta=\@l@dttempcntb
4438 \@l@dttempcntb=\line@margin
4439 \ifnum\@l@dttempcntb>1
4440 \advance\@l@dttempcntb by\page@num
4441 \fi
4442 \ifodd\@l@dttempcntb
4443 % #1\rlap{{\rightlinenum}}}%
4444 \xdef\rd@ta{\the\line@num}%
4445 \else
4446 \llap{{\leftlinenum}}}%#1%

```



```

4447 \fi
4448 \else
4449 %#1%
4450 \fi
4451 \ifcase\@lock
4452 \or
4453 \global\@lock=2
4454 \or \or
4455 \global\@lock=0
4456 \fi
4457 \ifcase\sub@lock
4458 \or
4459 \global\sub@lock=2
4460 \or \or
4461 \global\sub@lock=0
4462 \fi}
4463
4464 \lineation{page}
4465 \linenummargin{right}
4466 \footparagraph{A}
4467 \footparagraph{B}
4468
4469 \renewcommand{\notenumfont}{\footnotesize}
4470 \newcommand{\notetextfont}{\footnotesize}
4471
4472 \let\Afootnoterule=\relax
4473 \count\Afootins=825
4474 \count\Bfootins=825
4475
4476 \newcommand{\Aparafootfmt}[3]{%
4477 \ledsetnormalparstuff
4478 \scriptsize
4479 \notenumfont\printlines#1\enspace
4480 % \lemmafont#1/#2\enskip
4481 \notetextfont
4482 #3\penalty-10\hskip 1em plus 4em minus.4em\relax}
4483
4484 \newcommand{\Bparafootfmt}[3]{%
4485 \ledsetnormalparstuff
4486 \scriptsize
4487 \notenumfont\printlines#1\enspace
4488 \select@lemmafont#1/#2\rbracket\enskip
4489 \notetextfont
4490 #3\penalty-10\hskip 1em plus 4em minus.4em\relax }
4491 \makeatother
4492
4493 \let\Afootfmt=\Aparafootfmt
4494 \let\Bfootfmt=\Bparafootfmt
4495 \def\lemmafont#1/#2/#3/#4/#5/#6/#7/{\scriptsize}
4496 \parindent=1em

```

```

4497
4498 \newcommand{\lmarpar}[1]{\edtext{}{\ffootnote{#1}}}
4499 \newcommand{\rmarpar}[1]{\edtext{}{\gfootnote{#1}}}
4500 \emergencystretch40pt
4501
4502 %%%
4503
4504 \begin{document}
4505
4506 \beginnumbering
4507 \pstart
4508 \rmarpar{741C}
4509 \noindent \edtext{Incipit Quartus $PERIFUSEWN$}{%
4510 \lemma{incipit\ .~.~.\ $PERIFUSEWN$}\Bfootnote{\textit{om.\ R},
4511 incipit quartus \textit{M}}}
4512 \pend
4513 \medskip
4514
4515 \pstart
4516 \noindent \edtext{NVTRITOR}{\lemma{$ANAKEFALIOSIS$}\Bfootnote{\textit{
4517 FJP, lege} \langleanakefala'iwsis\rangle}.\lmarpar{$ANAKEFALIOSIS$
4518 NATVRARVM} Prima nostrae
4519 \edtext{Physiologiae}{\lemma{physiologiae}\Bfootnote{phisiologiae
4520 \textit{P}, physeologiae \textit{R}}}
4521 intentio praecipuaque mat\ -e\ -ria erat
4522 \edtext{quod}{\Bfootnote{\textit{p}.\ natura \textit{transp}.\ MR}}}
4523 \edtext{$UPEROUSIADES$}{\Bfootnote{\textit{codd}.\ Vtrum}
4524 \langleuperousi'wdhs$ (hoc est superessentialis) natura \textit{cum Gale
4525 (p.160) an} \langleuperousi'oths$ (hoc est superessentialis natura)
4526 \textit{cum Floss (PL 122,741C) intelligendum sit, ambigitur}}}
4527 (hoc est superessentialis) natura sit causa creatrix existentium et
4528 non existentium omnium, a nullo creata, unum principium, una
4529 origo, unus et uniuersalis uniuersorum fons, a nullo manans, dum
4530 ab eo manant omnia, trinitas coessentialis in tribus substantiis,
4531 $ANARQOS$ (hoc est sine principio), principium et finis, una
4532 bonitas, deus unus,
4533 \edtext{$OMOUSIOS$}{\Bfootnote{\textit{codd., lege} \langleomoo'usios\rangle}
4534 \edtext{et}{\lemma{\textbf{et}}\Bfootnote{\textit{
4535 R}1, \textit{om.\ R}0}}
4536 $UPEROUSIOS$ (id est coessentialis et superessentialis). Et, ut
4537 ait sanctus Epifanius, episcopus Constantiae Cypri, in
4538 \edtext{$AGKURATW$}{\Bfootnote{anchurato \textit{MR}}}
4539 sermone
4540 \edtext{de fide}{\Bfootnote{Glo\Ma\langle\am ssa\Ma\rangle\am: Ita
4541 enim uocatur sermo eius de fide $AGKURATOS$, id est procuratus
4542 \textit{mg}.\ add.\ FJP}}:
4543 \begin{itshape}Tria sancta, tria consancta, tria
4544 \edtext{agentia}{\Bfootnote{actiua \textit{MR}}},
4545 tria coagentia, tria
4546 \edtext{formantia}{\Bfootnote{formatiua \textit{MR}}},

```

```

4547 tria conformantia, tria
4548 \edtext{operantia}{\Bfootnote{operatiua \textit{MR}}},
4549 tria cooperantia, tria subsistentia, tria\rm arpar{742C}
4550 consubsistentia sibi inuicem coexistentia. Trinitas haec
4551 sancta uocatur: tria existentia, una consonantia, una deitas
4552 \edtext{eiusdem}{\Bfootnote{eiusdemque \textit{M}}}}
4553 essentiae,
4554 \edtext{eiusdem uirtutis, eiusdem
4555 \edtext{subsistentiae}{\Bfootnote{substantiae \textit{R}}}}{\%
4556 \Bfootnote{\textit{om.\ M}}},
4557 similia
4558 \edtext{similiter}{\Bfootnote{ex simili \textit{MR}}}}
4559 aequalitatem gratiae operantur patris et filii et sancti spiritus.
4560 Quomodo autem
4561 \edtext{sunt}{\Bfootnote{\textit{om.\ M}}},
4562 ipsis relinquitur docere:
4563 \edtext{'Nemo enim nouit patrem nisi filius, neque filium nisi pater,
4564 et cuicumque filius reuelauerit'}{\Afootnote{Matth.\ 11, 27}};
4565 reuelatur autem per spiritum sanctum. Non ergo haec tria existentia
4566 aut ex ipso aut per ipsum aut ad ipsum in unoquoque digne intelliguntur,
4567 \Ma\mid\! R, 264^{\rm r}\!\mid\!aM\ sicut ipsa reuelant:\end{itshape}
4568 $FWS, PUR, PNEUMA$
4569 \edtext{(hoc est lux, ignis, spiritus)}{\Afootnote{EPIPHANIVS,
4570 \textit{Ancoratus} 67; PG~43, 137C--140A; GCS 25, p.~82, 2--12}}.
4571 \pend
4572
4573 \pstart
4574 Haec, ut dixi, ab Epifanio tradita, ut quisquis interrogatus quae
4575 tria et quid unum in sancta trinitate debeat credere, sana fide
4576 \Ma\!\mid J, 1^{\rm v}\!\mid\!aM\ respondere ualeat, aut ad
4577 fidem accedens\rm arpar{743A} sic erudiatur. Et mihi uidetur
4578 spiritum pro calore posuisse, quasi dixisset in similitudine:
4579 lux, ignis, calor. Haec enim tria unius essentiae sunt. Sed cur
4580 lucem primo dixit, non est mirum. Nam et pater lux est et ignis
4581 et calor; et filius est lux, ignis, calor; et
4582 \edtext{spiritus sanctus}{\Bfootnote{sanctus spiritus \textit{R}}}}
4583 lux, ignis, calor. Illuminat enim pater, illuminat filius, illuminat
4584 spiritus sanctus: ex ipsis enim omnis scientia et sapientia donatur.
4585 \pend
4586 \endnumbering
4587
4588 \end{document}
4589
4590 </periph>

```

---

## B.7 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.

---

```

4591 (*dekker)
4592 %% This is ledekker.tex, a sample critical text edition
4593 %% written in LaTeX2e with the ledmac package.
4594 %% (c) 2003--2004 by Dr. Dirk-Jan Dekker,
4595 %% University of Nijmegen (The Netherlands)
4596 %% (PRW) Modified slightly by PRW to fit the ledmac manual
4597
4598 \documentclass[10pt, letterpaper, onside]{article}
4599 \usepackage[latin]{babel}
4600 \usepackage{ledmac}
4601
4602 \lineation{section}
4603 \linenummargin{left}
4604 \sidenotemargin{outer}
4605
4606 \renewcommand{\notenumfont}{\footnotesize}
4607 \newcommand{\notetextfont}{\footnotesize}
4608
4609 %\let\Afootnoterule=\relax
4610 %\let\Bfootnoterule=\relax
4611 %\let\Cfootnoterule=\relax
4612
4613 \addtolength{\skip\Afootins}{1.5mm}
4614 %\addtolength{\skip\Bfootins}{1.5mm}
4615 %\addtolength{\skip\Cfootins}{1.5mm}
4616
4617 \makeatletter
4618
4619 \renewcommand*{\para@vfootnote}[2]{%
4620 \insert\cename #1footins\endcename
4621 \bgroup
4622 \notefontsetup
4623 \footsplitskips
4624 \l@dparsefootspec #2\ledplinenumtrue % new from here
4625 \ifnum\@nameuse{previous@#1@number}=\l@dparsedstartline\relax
4626 \ledplinenumfalse
4627 \fi
4628 \ifnum\previous@page=\l@dparsedstartpage\relax
4629 \else \ledplinenumtrue \fi
4630 \ifnum\l@dparsedstartline=\l@dparsedendline\relax

```

---

<sup>32</sup>On 1st September 2004 the University changed its name to Radboud University.

```

4631 \else \ledplinenumtrue \fi
4632 \expandafter\xdef\csname previous@#1@number\endcsname{\l@dparsedstartline}
4633 \xdef\previous@page{\l@dparsedstartpage} % to here
4634 \setbox0=\vbox{\hsize=\maxdimen
4635 \noindent\csname #1footfmt\endcsname#2}%
4636 \setbox0=\hbox{\unvvh0}%
4637 \dp0=0pt
4638 \ht0=\csname #1footfudgefactor\endcsname\wd0
4639 \box0
4640 \penalty0
4641 \egroup
4642 }
4643
4644 \newcommand*\previous@A@number{-1}
4645 \newcommand*\previous@B@number{-1}
4646 \newcommand*\previous@C@number{-1}
4647 \newcommand*\previous@page{-1}
4648
4649 \newcommand{\abb}[1]{#1%
4650 \let\rbracket\nobrak\relax}
4651 \newcommand{\nobrak}{\textnormal{}}
4652 \newcommand{\morenoexpands}{%
4653 \let\abb=0%
4654 }
4655
4656 \newcommand{\Aparafootfmt}[3]{%
4657 \ledsetnormalparstuff
4658 \scriptsize
4659 \notenumfont\printlines#1\enspace
4660 % \lemmafont#1/#2\enskip
4661 \notetextfont
4662 #3\penalty-10\hskip 1em plus 4em minus.4em\relax}
4663
4664 \newcommand{\Bparafootfmt}[3]{%
4665 \ledsetnormalparstuff
4666 \scriptsize
4667 \notenumfont\printlines#1%
4668 \ifledplinenum
4669 \enspace
4670 \else
4671 {\hskip 0em plus 0em minus .3em}
4672 \fi
4673 \select@lemmafont#1/#2\rbracket\enskip
4674 \notetextfont
4675 #3\penalty-10\hskip 1em plus 4em minus.4em\relax }
4676
4677 \newcommand{\Cparafootfmt}[3]{%
4678 \ledsetnormalparstuff
4679 \notenumfont\printlines#1\enspace
4680 % \lemmafont#1/#2\enskip

```

```

4681 \notetextfont
4682 #3\penalty-10\hskip 1em plus 4em minus.4em\relax}
4683
4684 \makeatother
4685
4686 \footparagraph{A}
4687 \footparagraph{B}
4688 \footparagraph{C}
4689
4690 \let\Afootfmt=\Aparafootfmt
4691 \let\Bfootfmt=\Bparafootfmt
4692 \let\Cfootfmt=\Cparafootfmt
4693
4694 \emergencystretch40pt
4695
4696 \author{Guillelmus de Berchen}
4697 \title{Chronicle of Guelders}
4698 \date{}
4699 \hyphenation{archi-epi-sco-po Huns-dis-brug li-be-ra No-vi-ma-gen-si}
4700 \begin{document}
4701 \maketitle
4702 \thispagestyle{empty}
4703
4704 \section*{St.\ Stephen's Church in Nijmegen}
4705 \begin{numbering}
4706 \autopar
4707
4708 \noindent
4709 Nobilis itaque comes Otto imperio et dominio Novimagensi sibi,
4710 ut praeferetur, impignoratis et commissis
4711 \edtext{proinde}{\Bfootnote{primum D}} praeesse cupiens, anno
4712 \textsc{liiii}\ledsidenote{1254} superius descripto, mense
4713 Iu\edtext{}{\Afootnote{p.\ 227~R}}nio, una cum iudice, scabinis
4714 ceterisque civibus civitatis Novimagensis, pro ipsius et
4715 inhabitantium in ea necessitate,\edtext{}{\Afootnote{p.\ 97~N}}
4716 commodo et utilitate, ut
4717 \edtext{ecclesia eius}{\Bfootnote{ecclesia D: eius eius H}}
4718 parochialis
4719 \edtext{\abb{extra civitatem}}{\Bfootnote{\textit{om.}~H}} sita
4720 destrueretur et \edtext{infra}{\Bfootnote{intra D}} muros
4721 \edtext{transfer\edtext{}{\Afootnote{p.\ 129~D}}retur}%
4722 {\Bfootnote{transferreretur NH}}
4723 ac de novo construeretur, \edtext{a reverendo patre domino
4724 \edtext{Conrado de \edtext{Hofsteden}}%
4725 {\Bfootnote{Hoffstede D: Hoffsteden H}},
4726 archiepiscopo
4727 \edtext{Coloniensi}{\Bfootnote{Colononiensi H}}}%
4728 {\Cfootnote{Conrad of Hochstaden was archbishop of Cologne in
4729 1238--1261}}, licentiam){\Cfootnote{William is confusing two
4730 charters that are five years apart. Permission from St.\ Apostles'

```

4731 Church in Cologne had been obtained as early as 1249. Cf.\ Sloet,  
 4732 \textit{Oorkondenboek} nr.\ 707 (14 November 1249):  
 4733 ‘‘\ldots{}nos devotionis tue precibus annuentes, ut ipsam  
 4734 ecclesiam faciens demoliri transferas in locum alium competentem,  
 4735 tibi auctoritate presentium indulgemus\ldots{}’’}, et a  
 4736 venerabilibus \edtext{dominis}{\Bfootnote{viris H}} decano et  
 4737 capitulo sanctorum Apostolorum  
 4738 \edtext{Colonien}{\Bfootnote{Coloniae H}}, ipsius ecclesiae ab  
 4739 antiquo veris et pacificis patronis, consensum, citra tamen  
 4740 praeiudicium, damnum aut gravamen  
 4741 \edtext{iurium}{\Bfootnote{virium D}} et bonorum eorundem,  
 4742 impetravit.  
 4743  
 4744 \edtext{Et exinde \edtext{liberum}{\Bfootnote{librum H}} locum  
 4745 eiusdem civitatis \edtext{qui}{\Bfootnote{quae D}} dicitur  
 4746 \edtext{Hundisbrug}{\Bfootnote{Hundisburch D: Hunsdisbrug R}},  
 4747 de praelibati Wilhelmi Romanorum  
 4748 \edtext{regis}{\Bfootnote{imperatoris D}}, ipsius fundi  
 4749 do\edtext{}{\Afootnote{f.\ 72v~M}}mini, consensu, ad aedificandum  
 4750 \edtext{\abb{et consecrandum}}{\Bfootnote{\textit{om.}\ H}}  
 4751 ecclesi\edtext{}{\Afootnote{p.\ 228~R}}am et coemeterium,  
 4752 \edtext{eisdem}{\Bfootnote{eiusdem D}} decano et capitulo de  
 4753 expresso eiusdem civitatis assensu libera contradiderunt voluntate,  
 4754 obligantes se ipsi \edtext{comes}{\Bfootnote{comites D}} et civitas  
 4755 \edtext{\abb{dictis}}{\Bfootnote{\textit{om.}\ H}} decano et  
 4756 capitulo, quod in recompensationem illius areae infra castrum et  
 4757 portam, quae fuit dos ecclesiae, in qua plebanus habitare  
 4758 solebat---quae \edtext{tunc}{\Bfootnote{nunc H}} per novum fossatum  
 4759 civitatis est destructa---aliam aream competentem et ecclesiae  
 4760 novae,  
 4761 \edtext{ut praefertur, aedificandae}{\lemma{\abb{ut\ldots aedificandae}}}%  
 4762 \Bfootnote{\textit{om.}\ H}} satis  
 4763 \edtext{contiguam}{\Bfootnote{contiguum M}}, ipsi plebano darent  
 4764 et assignarent. Et desuper  
 4765 \edtext{\abb{apud}}{\Bfootnote{\textit{om.}\ H}} dictam ecclesiam  
 4766 sanctorum Apostolorum \edtext{est}{\Bfootnote{et H}}  
 4767 \edtext{littera}{\Bfootnote{litteram H}} sigillis ipsorum  
 4768 Ottonis\edtext{}{\Afootnote{p.\ 130~D}} comitis et civitatis  
 4769 \edtext{Novimagensis}{\Bfootnote{Novimagii D}}  
 4770 \edtext{sigillata}{\Bfootnote{sigillis communita H}}.}%  
 4771 {\Cfootnote{Cf.\ Sloet, \textit{Oorkondenboek} nr.\ 762 (June 1254)}}  
 4772  
 4773 % (PRW) the full document continues on after this point  
 4774 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
 4775 \endnumbering  
 4776 \end{document}  
 4777 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
 4778  
 4779 </dekker>

---

## B.8 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.

---

```

4780 (*braonain)
4781 %% This is ledbraonain.tex, a sample critical verse edition.
4782 %% Originally written for TeX processing with edmac and edstanza
4783 %% by Wayne Sullivan.
4784 %% Extensively modified by Peter Wilson for LaTeX and the ledmac package.
4785
4786 \documentclass{article}
4787 \usepackage{ledmac}
4788
4789 \setlength{\textheight}{40pc}
4790 \setlength{\textwidth}{24pc}
4791 \bigskipamount=12pt plus 6pt minus 6pt
4792 \newcommand*{\notetextfont}{\footnotesize}
4793
4794 %% Just one footnote series
4795 \footparagraph{C}
4796 \count\Cfootins=800
4797 \makeatletter
4798 %% but using two different formats
4799 \def\xparafootfmt#1#2#3{%
4800 \ledsetnormalparstuff
4801 {\notenumfont\printlines#1}\enspace
4802 %% {\select@lemmafnt#1/#2}\rbracket\enskip
4803 \notetextfont #3\penalty-10 }
4804 \def\yparafootfmt#1#2#3{%
4805 \ledsetnormalparstuff
4806 %% {\notenumfont\printlines#1}\enspace
4807 %% {\select@lemmafnt#1/#2}\rbracket\enskip
4808 \notetextfont #3\penalty-10 }
4809
4810 \let\Cfootfmt=\xparafootfmt
4811 \skip\Cfootins=\bigskipamount
4812 \makeatother
4813
4814 %% This is the default, but just to demonstrate...
4815 \setlength{\stanzaindentbase}{20pt}
4816

```



```

4817 %% MUST SET THE INDENTS
4818 %% indent multiples; first=hangindent.
4819 %% Must all be non-negative whole numbers
4820 \setstanzaindents{4,1,2,1,2,3,3,1,2,1}
4821
4822 %% Set stanza line penalties
4823 %% Must be nonnegative whole numbers.
4824 %% An initial zero indicates no penalties.
4825 \setstanzapenalties{1,5000,10500,5000,10500,5000,5000,5000,0}
4826 %\setstanzapenalties{0}% the default
4827
4828 %% Put some space between stanzas
4829 \let\endstanzaextra=\bigbreak % ==> \bigskip \penalty -200
4830
4831 %% (almost) force line break in foot paragraph
4832 \mathchardef\IMM=9999
4833 \def\lbreak{\hfil\penalty-\IMM}
4834
4835 %% Number each stanza in bold
4836 \newcounter{stanzanum}
4837 \setcounter{stanzanum}{0}
4838 \newcommand*{\numberit}{%
4839 \flagstanza[0.5\stanzaindentbase]{\textbf{\thestanzanum}}}
4840 %% Use the hook to insert the number (and counteract a new line)
4841 %% and reset the line number to zero
4842 \newcommand*{\startstanzahook}{\refstepcounter{stanzanum}%
4843 \numberit\vskip-\baselineskip%
4844 \setlinenum{0}}
4845
4846 %% Want to label the footnotes with the stanza and line number
4847 %% We'll use \linenum to replace the sub-line number
4848 %% with the stanza number, redefining \edtext to do this
4849 %% automatically for us.
4850 %%%
4851 \makeatletter
4852
4853 \renewcommand{\edtext}[2]{\leavevmode
4854 \begingroup
4855 \xdef\@tag{#1}%
4856 \set@line
4857 \global\insert@count=0
4858 \ignorespaces \linenum{/\the\c@stanzanum}#2\relax
4859 \flag@start
4860 \endgroup
4861 #1%
4862 \ifx\end@lemmas\empty \else
4863 \gl@p\end@lemmas\to\x@lemma
4864 \x@lemma
4865 \global\let\x@lemma=\relax
4866 \fi

```

```

4867 \flag@end}
4868
4869 %% We need only a very simple macro for footnote numbers,
4870 %% to produce the stanza number (sub-line) then the line number.
4871 \def\printstanzalines#1/#2/#3/#4/#5/#6/#7/{\begingroup
4872 #3\fullstop \linenumrep{#2}
4873 \endgroup}
4874 \let\oldprintlines\printlines
4875
4876 \makeatother
4877 %%%
4878
4879 \pagestyle{empty}
4880
4881 \begin{document}
4882
4883 \beginnumbering
4884
4885 \pstart \centering \textbf{22} \pend
4886
4887 \bigskip
4888 %% do not print line number beside heading
4889 \setcounter{firstlinenum}{1000}
4890 %% and heading footnotes use a different format
4891 \let\Cfootfmt=\yparafootfmt
4892
4893 \pstart
4894 \centerline{[Se\'an \'O Braon\'ain cct] chuim Tom\'ais U\'{i}
4895 \edtext{Dh\'unlaing}{\Cfootnote{\textbf{22} \textit{Teideal}: Dhuinnluing T,
4896 Se\'aghan Mac Domhnaill cct B\lbreak}}}
4897 \pend
4898
4899 \pstart
4900 \centerline{[Fonn: M\'airse\'ail U\'{i} Sh\'uilleabh\'ain (P\'ainseach
4901 na nUbh]}
4902 \pend
4903
4904 \bigskip
4905
4906 %% revert to the regular footnote format
4907 \let\Cfootfmt=\xparafootfmt
4908 %% but use our special number printing routine
4909 \let\printlines\printstanzalines
4910 %% Use letters for line numbering
4911 \linenumberstyle{alph}
4912 %% number lines from the second onwards
4913 \setcounter{firstlinenum}{2}
4914 \setcounter{linenumincrement}{1}
4915
4916 %% Each verse starts with \stanza.

```

```

4917 %% Lines end with &; the last line with \&.
4918
4919 \stanza
4920 A \edtext{dhuine}{\Cfootnote{dhuinne T}} gan ch\’eill do
4921 \edtext{mhaisligh}{\Cfootnote{mhaslaidh T, mhaslaig B}} an chl\’eir&
4922 is tharcaisnigh naomhscruipt na bhf\’aige,&
4923 na haitheanta \edtext{r\’eab}{\Cfootnote{raob T}} ’s an
4924 t-aifreann thr\’eig&
4925 \edtext{re}{\Cfootnote{le B}} taithneamh do chlaonchreideamh
4926 Mh\’artain,&
4927 c\’a rachair \edtext{’od}{\Cfootnote{dod B}} dh\’{\i}on ar
4928 \’Iosa Nasardha&
4929 nuair \edtext{chaithfimid}{\Cfootnote{chaithfam\’{\i}d T}} cruinn
4930 bheith ar \edtext{mhaoileann}{\Cfootnote{maoilinn B}} Josepha?&
4931 N\’{\i} caraoid Mac Crae chuim t’anama ’
4932 \edtext{phl\’e}{\Cfootnote{phleidh T}}&
4933 n\’a Calvin \edtext{bhiais}{\Cfootnote{bh\’{\i}os B}} taobh
4934 \edtext{ris}{\Cfootnote{leis B}} an l\’a sin.\&
4935
4936 \stanza
4937 N\’ach damanta an sc\’eal don chreachaire
4938 \edtext{chlaon}{\Cfootnote{claon B}}&
4939 ghlac baiste na cl\’eire ’na ph\’aiste&
4940 ’s do \edtext{glanadh}{\Cfootnote{glannuig T}} mar ghr\’ein
4941 \’on bpeaca r\’o-dhaor&
4942 tr\’{\i} \edtext{ainibhfios}{\Cfootnote{ainnibhfios T, ainnbhfios B}}
4943 \edtext{\’Eva}{\Cfootnote{\’Eabha B}} rinn \’Adam,&
4944 tuitim ar\’{\i}s f\’e chuing na haicme sin&
4945 tug atharrach br\’{\i} don scr\’{\i}bhinn bheannaithe,&
4946 d’aistrigh b\’easa \edtext{agus}{\Cfootnote{is B}} reachta na cl\’eire&
4947 ’s n\’ach \edtext{tugann}{\Cfootnote{tuigionn T}} aon
4948 gh\’eilleadh don Ph\’apa?\&
4949
4950 \stanza
4951 Gach \edtext{scolaire}{\Cfootnote{sgollaire T}} baoth, n\’{\i}
4952 \edtext{mholaim}{\Cfootnote{mholluim T}} a cheird&
4953 \edtext{’t\’a ag obair}{\Cfootnote{’t\’ag ccoabar T}}
4954 \edtext{le}{\Cfootnote{re B}} g\’eilleadh d\’a th\’aille&
4955 don \edtext{doirbhchoin chlaon}{\Cfootnote{dorbhchon daor B}}
4956 d\’a ngorthar Mac Crae,&
4957 deisceabal \edtext{straeigh}{\Cfootnote{straodhaig T}} as an
4958 gcoll\’aiste.&
4959 T\’a \edtext{\edtext{adaithe}}{\Cfootnote{fadaighthe B}}
4960 th\’{\i}os{\Cfootnote{fhadoghthe ts\’{\i}os T}} in
4961 \’{\i}ochtar ifrinn,&
4962 gan \edtext{solas}{\Cfootnote{sollus T}} gan soilse i
4963 dt\’{\i}orthaibh dorchu,&
4964 tuigsint an l\’einn, gach
4965 \edtext{cuirpeacht}{\Cfootnote{cuirripeacht T}} d\’ein&
4966 is \edtext{Lucifer}{\Cfootnote{Luicifer T, L\’ucifer B}} aosta

```

```
4967 'na \edtext{mh\'aistir}{\Cfootnote{mhaighistir T}}.\&
4968
4969 \endnumbering
4970
4971 \end{document}
4972
4973 </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>              | <i>21</i> , 3142, 3146, 3147, 3196, 3205,<br>3211, 3213, 4917, 4934, 4948, 4967                                                                                                                                                                                                                                                            |
| <code>\-</code>                  | ..... 4294, 4521                                                                                                                                                                                                                                                                                                                           |
| <code>\@oline</code>             | ..... 1596                                                                                                                                                                                                                                                                                                                                 |
| <code>\@wrindexm@m</code>        | .. 3063, 3068, 3071, 3095                                                                                                                                                                                                                                                                                                                  |
| <code>\@EDROWFILL@</code>        | ..... 3434, <u>3641</u>                                                                                                                                                                                                                                                                                                                    |
| <code>\@M</code>                 | ..... 1596, 3190, 3200                                                                                                                                                                                                                                                                                                                     |
| <code>\@MM</code>                | ..... 1313                                                                                                                                                                                                                                                                                                                                 |
| <code>\@adv</code>               | ..... <u>451</u> , 612                                                                                                                                                                                                                                                                                                                     |
| <code>\@arabic</code>            | ..... 728                                                                                                                                                                                                                                                                                                                                  |
| <code>\@aux</code>               | ..... 2235                                                                                                                                                                                                                                                                                                                                 |
| <code>\@auxout</code>            | .. 2237, 3062, 3067, 3070, 3094                                                                                                                                                                                                                                                                                                            |
| <code>\@botlist</code>           | ..... 2188, 2190                                                                                                                                                                                                                                                                                                                           |
| <code>\@ccclv</code>             | 2083, 2087, 2088, 2186, 2187, 2215                                                                                                                                                                                                                                                                                                         |
| <code>\@checkend</code>          | ..... 3123                                                                                                                                                                                                                                                                                                                                 |
| <code>\@colht</code>             | ..... 2065, 2191, 2203                                                                                                                                                                                                                                                                                                                     |
| <code>\@colroom</code>           | ..... 2191                                                                                                                                                                                                                                                                                                                                 |
| <code>\@combinefloats</code>     | ..... 2060                                                                                                                                                                                                                                                                                                                                 |
| <code>\@currenvir</code>         | ..... 3108, 3111, 3112                                                                                                                                                                                                                                                                                                                     |
| <code>\@currlist</code>          | ..... 2192, 2195                                                                                                                                                                                                                                                                                                                           |
| <code>\@dbldeferlist</code>      | ..... 2201, 2206, 2208                                                                                                                                                                                                                                                                                                                     |
| <code>\@dblfloatplacement</code> | ..... 2205                                                                                                                                                                                                                                                                                                                                 |
| <code>\@dbltoplist</code>        | ..... 2201, 2202                                                                                                                                                                                                                                                                                                                           |
| <code>\@deferlist</code>         | ..... 2188, 2197, 2198                                                                                                                                                                                                                                                                                                                     |
| <code>\@docclearpage</code>      | ..... <u>2170</u>                                                                                                                                                                                                                                                                                                                          |
| <code>\@edrowfill@</code>        | ..... <u>3641</u>                                                                                                                                                                                                                                                                                                                          |
| <code>\@ehb</code>               | ..... 2194                                                                                                                                                                                                                                                                                                                                 |
| <code>\@emptytoks</code>         | ..... <u>3099</u> , 3109                                                                                                                                                                                                                                                                                                                   |
| <code>\@footnotemark</code>      | ..... <u>2546</u>                                                                                                                                                                                                                                                                                                                          |
| <code>\@footnotemarkA</code>     | ..... 2579, 2591                                                                                                                                                                                                                                                                                                                           |
| <code>\@footnotemarkB</code>     | ..... 2864, 2875                                                                                                                                                                                                                                                                                                                           |
| <code>\@footnotemarkC</code>     | ..... 2882, 2891                                                                                                                                                                                                                                                                                                                           |
| <code>\@footnotetext</code>      | .....<br>..... <u>2542</u> , <u>2559</u> , 2965, 2991, 3018                                                                                                                                                                                                                                                                                |
| <code>\@freelist</code>          | ..... 2058                                                                                                                                                                                                                                                                                                                                 |
| <code>\@gobble</code>            | 646–650, 2357, 3120, 3403, 3420                                                                                                                                                                                                                                                                                                            |
| <code>\@gobblethree</code>       | ..... <u>2349</u>                                                                                                                                                                                                                                                                                                                          |
| <code>\@h</code>                 | ..... <u>1594</u>                                                                                                                                                                                                                                                                                                                          |
| <code>\@hilfs@count</code>       | ..... <u>3336</u>                                                                                                                                                                                                                                                                                                                          |
| <code>\@idxfile</code>           | 3053, 3063, 3068, 3071, 3095                                                                                                                                                                                                                                                                                                               |
| <code>\@ifclassloaded</code>     | .....<br>..... 23, 2133, 2158, 2541, 3040                                                                                                                                                                                                                                                                                                  |
| <code>\@ifnextchar</code>        | ..... 3043, 3316                                                                                                                                                                                                                                                                                                                           |
| <code>\@ifpackageloaded</code>   | ..... 3091                                                                                                                                                                                                                                                                                                                                 |
| <code>\@iiiminipage</code>       | ..... 2954                                                                                                                                                                                                                                                                                                                                 |
| <code>\@iiiparbox</code>         | ..... 2981                                                                                                                                                                                                                                                                                                                                 |
| <code>\@indexfile</code>         | ..... 3085                                                                                                                                                                                                                                                                                                                                 |
| <code>\@inputcheck</code>        | ..... 338                                                                                                                                                                                                                                                                                                                                  |
| <code>\@insert</code>            | ..... 1096–1098, 1132–1134                                                                                                                                                                                                                                                                                                                 |
| <code>\@k</code>                 | ..... <u>1594</u>                                                                                                                                                                                                                                                                                                                          |
| <code>\@kludgeins</code>         | ..... 2062, 2130                                                                                                                                                                                                                                                                                                                           |
| <code>\@l</code>                 | ..... <u>368</u> , 595                                                                                                                                                                                                                                                                                                                     |
| <code>\@l@dttempcnta</code>      | ... <u>21</u> , 485, 487, 489,<br>490, 880, 881, 883, 885, 888,<br>889, 904, 940–944, 946, 953–<br>957, 959, 962, 965, 967, 971,<br>1001, 1005, 1009, 1016, 1020,<br>1024, 1105, 1109, 1113, 1116,<br>1119, 1122, 1123, 4387–4391,<br>4393, 4399, 4403, 4407, 4413–<br>4417, 4419, 4424, 4428, 4432, 4437                                  |
| <code>\@l@dttempcntb</code>      | <u>21</u> , 205, 206, 211, 215,<br>219, 223, 226, 249, 250, 257,<br>261, 265, 267, 275, 276, 938,<br>950, 971, 979–981, 983, 1001,<br>1005, 1009, 1016, 1020, 1024,<br>1054–1056, 1058, 1111, 1112,<br>2415, 2416, 2421, 2425, 2429,<br>2433, 2436, 2515–2517, 2519,<br>4385, 4399, 4403, 4407, 4411,<br>4424, 4428, 4432, 4437–4440, 4442 |
| <code>\@l@reg</code>             | ..... 368                                                                                                                                                                                                                                                                                                                                  |
| <code>\@lab</code>               | ..... 568, 2226, <u>2272</u>                                                                                                                                                                                                                                                                                                               |
| <code>\@latexerr</code>          | ..... 2194                                                                                                                                                                                                                                                                                                                                 |
| <code>\@led@extranofeet</code>   | .. <u>2155</u> , 2168, 2181                                                                                                                                                                                                                                                                                                                |
| <code>\@led@nofootfalse</code>   | ..... 2172–2180                                                                                                                                                                                                                                                                                                                            |
| <code>\@led@nofoottrue</code>    | ..... 2171                                                                                                                                                                                                                                                                                                                                 |
| <code>\@led@testifnofoot</code>  | ..... <u>2170</u>                                                                                                                                                                                                                                                                                                                          |
| <code>\@line@@num</code>         | ..... <u>3332</u> , 3446                                                                                                                                                                                                                                                                                                                   |
| <code>\@listdepth</code>         | ..... 2967                                                                                                                                                                                                                                                                                                                                 |
| <code>\@lock</code>              | ..... 134, <u>326</u> , 393,<br>395, 397, 410, 518, 519, 521,<br>522, 538, 539, 541, 813, 851,<br>910, 912, 913, 915, 1013, 1028,<br>1030, 1032, 4421, 4451, 4453, 4455                                                                                                                                                                    |

- \@lopL ..... 435
  - \@lopR ..... 435
  - \@makecol ..... 2137
  - \@makefcolumn .. 2197, 2198, 2206, 2208
  - \@makespecialcolbox ..... 2063
  - \@maxdepth ..... 2078, 2086
  - \@mem@extranofeet ..... 2159
  - \@mem@nofootfalse ..... 2160–2167
  - \@midlist ..... 2058, 2059
  - \@minipagefalse ..... 2978
  - \@minipagerestore ..... 2968
  - \@minus ..... 1468, 1473, 2684, 2690
  - \@mpargs ..... 2958, 2981
  - \@mpfn ..... 2964, 2990, 3017
  - \@mpfootins 2974, 2984, 2987, 2996, 3023
  - \@mpfootnotetext ... 2965, 2991, 3018
  - \@mplistdepth ..... 2967
  - \@nameuse ..... 282,  
284, 1318, 1319, 1449, 1451,  
1522, 1523, 1562, 1564, 1648,  
1650, 1695, 1697, 2144, 2145,  
2147, 2148, 2150, 2152, 2608,  
2612, 2614, 2618, 2621, 2622,  
2627, 2631, 2635, 2638, 2641,  
2646, 2649, 2651, 2652, 2656,  
2663, 2669, 2714, 2724, 2732,  
2734, 2759, 2769, 2777, 2779,  
2806, 2810, 2811, 2819, 2827,  
2828, 2832, 2842, 2852, 2854,  
2918, 2919, 2921, 2922, 2927, 4625
  - \@nobreakfalse ..... 735
  - \@nobreaktrue ..... 733, 737
  - \@nowrindex ..... 3052
  - \@oldnobreak ..... 733, 735, 774
  - \@opcol ..... 2198, 2216
  - \@opxtrafeetii ..... 2107, 2108, 2143
  - \@outputbox .....  
. 2065, 2067, 2068, 2083, 2085,  
2105, 2106, 2900, 2901, 2916, 2917
  - \@outputpage ..... 2207
  - \@page ..... 415
  - \@parboxrestore .... 1322, 2625, 2963
  - \@pboxswfalse ..... 2956
  - \@pend ..... 435
  - \@pendR ..... 435
  - \@plus ..... 1328,  
1468, 1473, 2684, 2690, 2720, 2765
  - \@ref ..... 553, 598
  - \@ref@reg ..... 555
  - \@reinsets ..... 2138
  - \@set ..... 466, 617
  - \@setminipage ..... 2969
  - \@showidx ..... 3060, 3093
  - \@tag ..... 660, 675, 703,  
2331, 2335, 2339, 2343, 2347, 4855
  - \@tempboxa .... 2186, 2187, 2959, 2981
  - \@tempdima ..... 2087, 2957, 2961
  - \@templ@d ..... 2506, 2507
  - \@textbottom ..... 2070
  - \@texttop ..... 2066
  - \@thefnmarkA ..... 2578, 2590
  - \@thefnmarkB ..... 2863, 2874
  - \@thefnmarkC ..... 2881, 2890
  - \@toksa ..... 309, 317
  - \@toksb ..... 309, 316–318
  - \@toplist ..... 2188, 2189
  - \@whilenum ..... 3615
  - \@whilesw ..... 2198, 2207
  - \@wredindex ..... 3082, 3084, 3323
  - \@x@sf 2535, 2538, 2549, 2555, 2598, 2604
  - \@xloop ..... 1130, 1137
  - \@xympar ..... 2403
  - \^ ..... 362, 3221
  - \\_ ... 1424, 3649, 3654, 3662, 3995,  
4030, 4153, 4156, 4510, 4522,  
4523, 4535, 4542, 4556, 4561,  
4564, 4567, 4576, 4704, 4713,  
4715, 4721, 4730–4732, 4749–  
4751, 4755, 4762, 4765, 4768, 4771
- A**
- \A@@footnote ... 3327, 3405, 3422, 3441
  - \abb ..... 4649,  
4653, 4719, 4750, 4755, 4761, 4765
  - \absline@num ... 131, 325, 373, 376,  
379, 480, 483, 492, 506, 528,  
550, 560, 841, 863, 864, 872, 1095
  - Abu Kamil Shuja' b. Aslam ..... 8
  - \actionlines@list .....  
..... 328, 345, 348, 355, 480,  
483, 492, 506, 528, 550, 893, 896
  - \actions@list .....  
. 328, 349, 356, 481, 490, 494,  
496, 508, 517, 530, 537, 551, 897
  - \add@inserts ..... 820, 1084
  - \add@inserts@next ..... 1084
  - \add@penalties ..... 1105
  - \addfootins ..... 2141

- \addfootinsX ..... 2913
  - \addtocounter ..... 775
  - \addtol@denvbody ... 3103, 3124, 3126
  - \addtolength ..... 4613–4615
  - Adelard II ..... 7
  - \advancelabel@refs ..... 2232, 2241
  - \advanceline . 13, 57, 60, 612, 635, 650
  - \advancepageno ..... 2053
  - \Aend ..... 2329, 2349
  - \Aendnote ..... 15, 2328
  - \affixline@num ..... 818, 932, 4383
  - \affixpstart@num ..... 819, 1043
  - \affixside@note ..... 820, 2505
  - \Afootfmt .....
    - 4096, 4233, 4247, 4266, 4493, 4690
  - \Afootgroup ..... 2109
  - \Afootins ..... 1456, 2109, 2120,
    - 2160, 2173, 4097–4099, 4473, 4613
  - \Afootnote ..... 14,
    - 1148, 2933, 3327, 3397, 3405,
    - 3414, 3422, 3441, 3890, 3897,
    - 3983, 3986, 3988, 4019, 4021,
    - 4094, 4123, 4149, 4151–4153,
    - 4156, 4167, 4235, 4564, 4569,
    - 4713, 4715, 4721, 4749, 4751, 4768
  - \Afootnoterule ..... 4202, 4472, 4609
  - \Afootstart ..... 2109, 4100
  - \afterlemmaseparator ..... 18, 1910
  - \afterlemmaseparator@A ..... 1910
  - \afterlemmaseparator@B ..... 1911
  - \afterlemmaseparator@C ..... 1912
  - \afterlemmaseparator@D ..... 1913
  - \afterlemmaseparator@E ..... 1914
  - \afterlemmaseparator@Z ..... 1915
  - \afternote ..... 18, 1956
  - \afternote@A ..... 1956
  - \afternote@B ..... 1957
  - \afternote@C ..... 1958
  - \afternote@D ..... 1959
  - \afternote@E ..... 1960
  - \afternote@Z ..... 1961
  - \afternumberinfootnote .... 17, 1789
  - \afternumberinfootnote@A . 1789, 1829
  - \afternumberinfootnote@B . 1790, 1830
  - \afternumberinfootnote@C . 1791, 1831
  - \afternumberinfootnote@D . 1792, 1832
  - \afternumberinfootnote@E . 1793, 1833
  - \afternumberinfootnote@Z . 1794, 1834
  - \aftersymmlinenumber ..... 17, 1829
  - \aftersymmlinenumber@A ..... 1829
  - \aftersymmlinenumber@B ..... 1830
  - \aftersymmlinenumber@C ..... 1831
  - \aftersymmlinenumber@D ..... 1832
  - \aftersymmlinenumber@E ..... 1833
  - \aftersymmlinenumber@Z ..... 1834
  - \allowbreak .....
    - 1642, 1685, 2725, 2770, 3941, 3954
  - \alph ..... 3867
  - \alpha ..... 3988, 4023
  - \aM ..... 4343, 4540, 4567, 4576
  - \ampersand ..... 22, 3142, 3213
  - \Aparafootfmt . 4476, 4493, 4656, 4690
  - \AtBeginDocument ..... 2268, 3091
  - \author ..... 3874, 4696
  - \autopar ..... 10, 78, 782, 4706
  - \autoparfalse ..... 160, 783
  - \autopartrue ..... 796
- B**
- \B@@footnote ... 3327, 3406, 3423, 3442
  - \ballast ..... 33
  - \ballast@count ... 858, 861, 866, 1105
  - Beeton, Barbara Ann Neuhaus Friend 10
  - \beforelemmaseparator ..... 17, 1890
  - \beforelemmaseparator@A ..... 1890
  - \beforelemmaseparator@B ..... 1891
  - \beforelemmaseparator@C ..... 1892
  - \beforelemmaseparator@D ..... 1893
  - \beforelemmaseparator@E ..... 1894
  - \beforelemmaseparator@Z ..... 1895
  - \beforenumberinfootnote ... 17, 1769
  - \beforenumberinfootnote@A 1769, 1809
  - \beforenumberinfootnote@B 1770, 1810
  - \beforenumberinfootnote@C 1771, 1811
  - \beforenumberinfootnote@D 1772, 1812
  - \beforenumberinfootnote@E 1773, 1813
  - \beforenumberinfootnote@Z 1774, 1814
  - \beforemsymmlinenumber ..... 17, 1809
  - \beforemsymmlinenumber@A ..... 1809
  - \beforemsymmlinenumber@B ..... 1810
  - \beforemsymmlinenumber@C ..... 1811
  - \beforemsymmlinenumber@D ..... 1812
  - \beforemsymmlinenumber@E ..... 1813
  - \beforemsymmlinenumber@Z ..... 1814
  - \beginnumbering .....
    - .. 9, 115, 173, 740, 793, 3885,
    - 3979, 4121, 4249, 4506, 4705, 4883
  - \beginnumberingR ..... 788
  - \Bend ..... 2333, 2349
  - \Bendnote ..... 15, 2328



- \beta ..... 3988, 4023
  - \Bfootfmt ..... 4234, 4494, 4691
  - \Bfootgroup ..... 2110
  - \Bfootins ..... 1456,  
2110, 2121, 2161, 2174, 4474, 4614
  - \Bfootnote .. 14, 1163, 2934, 3328,  
3398, 3406, 3415, 3423, 3442,  
3891, 3990, 4023, 4025, 4033,  
4035, 4037, 4039, 4041, 4043,  
4045, 4047, 4049, 4236, 4510,  
4516, 4519, 4522, 4523, 4533,  
4534, 4538, 4540, 4544, 4546,  
4548, 4552, 4555, 4556, 4558,  
4561, 4582, 4711, 4717, 4719,  
4720, 4722, 4725, 4727, 4736,  
4738, 4741, 4744–4746, 4748,  
4750, 4752, 4754, 4755, 4758,  
4762, 4763, 4765–4767, 4769, 4770
  - \Bfootnoterule ..... 4202, 4610
  - \Bfootstart ..... 2110
  - \bfseries ..... 728, 3965
  - \bigbreak ..... 4829
  - \bigskip . 4258, 4264, 4829, 4887, 4904
  - \bigskipamount ..... 4791, 4811
  - \body ..... 1138, 1139, 3144, 3212
  - \bodyfootmarkA ..... 27
  - \box 832, 834, 1517, 1531, 1576, 1595,  
2083, 2187, 2215, 2823, 2836, 4639
  - \boxmaxdepth ..... 2086
  - \Bparafootfmt .. 4484, 4494, 4664, 4691
  - Bredon, Simon ..... 7
  - Breger, Herbert ..... 5, 8, 158
  - Brey, Gerhard ..... 7, 8
  - \brokenpenalty ..... 778
  - Burt, John ..... 6
  - Busard, Hubert L. L. .... 7
  - \bypage@false ..... 176, 189, 194
  - \bypage@true ..... 176, 184
  - \bypstart@false ..... 176, 185, 195
  - \bypstart@true ..... 176, 190
- C**
- \C@footnote ... 3327, 3407, 3424, 3443
  - \c@addcolcount ..... 3609
  - \c@ballast ..... 858, 866
  - \c@firstlinenum ... 232, 952, 954,  
957, 959, 4412, 4414, 4417, 4419
  - \c@firstsublinenum . 236, 939, 941,  
944, 946, 4386, 4388, 4391, 4393
  - \c@footnoteA ..... 2583
  - \c@footnoteB ..... 2867
  - \c@footnoteC ..... 2884
  - \c@labidx ..... 3029
  - \c@linenumincrement .....  
..... 232, 955, 956, 4415, 4416
  - \c@mpfootnote ..... 2964, 2990, 3017
  - \c@page ..... 595
  - \c@pstart ..... 728
  - \c@stanzanum ..... 4858
  - \c@sublinenumincrement .....  
..... 236, 942, 943, 4389, 4390
  - \Cend ..... 2337, 2349
  - \Cendnote ..... 15, 2328
  - \centering ..... 4885
  - \centerline ..... 3545,  
3550, 3556, 3561, 3567, 3572,  
3793, 3795, 4251, 4261, 4894, 4900
  - \Cfootfmt ..... 4692, 4810, 4891, 4907
  - \Cfootgroup ..... 2111
  - \Cfootins ..... 1457, 2111,  
2122, 2162, 2175, 4615, 4796, 4811
  - \Cfootnote ..... 14, 1163,  
2935, 3329, 3399, 3407, 3416,  
3424, 3443, 3996, 4031, 4728,  
4729, 4771, 4895, 4920, 4921,  
4923, 4925, 4927, 4929, 4930,  
4932–4934, 4938, 4940, 4942,  
4943, 4946, 4947, 4951–4955,  
4957, 4959, 4960, 4962, 4965–4967
  - \Cfootnoterule ..... 4611
  - \Cfootstart ..... 2111
  - \ch@ck@l@ck ..... 969, 997
  - \ch@cksub@l@ck ..... 948, 997
  - \char ..... 3142
  - \chardef ..... 2402, 3144, 3146
  - Chester, Robert of ..... 7
  - Claassens, Geert H. M. .... 8
  - class 1 feet ..... 120, 146
  - class 2 feet ..... 120, 121
  - \cleaders ..... 3604
  - \closeout ..... 587, 591, 2320
  - \clubpenalty ..... 778, 1109
  - \collation ..... 4235, 4253,  
4255, 4264, 4271, 4283, 4288,  
4290, 4300, 4309, 4311, 4317, 4326
  - \color@begingroup .....  
1323, 1527, 2090, 2626, 2832, 2960
  - \color@endgroup .....  
1324, 1527, 2094, 2627, 2832, 2979

- `\columnwidth` ..... 1321, 1498, 2624, 2962, 3005  
`\content` .. 1149, 1164, 1179, 1194,  
1209, 1231, 1246, 1261, 1276, 1291  
Copernicus, Nicolaus ..... 7  
`\copy` ..... 4109  
`\count` ..... 1466, 1471, 1488,  
1492, 1618, 1621, 1666, 1692,  
2682, 2687, 2703, 2706, 2749,  
2752, 2791, 2794, 4473, 4474, 4796  
`\countdef` ..... 2053  
`\Cparafootfmt` ..... 4677, 4692  
`\cr` ..... 1597, 1600  
`\CRITEXT` ..... 3307  
`\critext` 37, 651, 657, 3309, 3396, 3438  
`\cs` .. 1417, 1425, 1429, 1437–1439,  
1747, 1934, 1980, 2011, 2241–2243  
`\csdef` 1733–1738, 1741, 1778–1783,  
1786, 1798–1803, 1806, 1818–  
1823, 1826, 1838–1843, 1846,  
1858–1863, 1866, 1879–1884,  
1887, 1899–1904, 1907, 1919–  
1924, 1927, 1945–1950, 1953,  
1965–1970, 1973, 1996–2001, 2004  
`\csexpandonce` .....  
. 1155, 1158, 1170, 1173, 1185,  
1188, 1200, 1203, 1215, 1218,  
1237, 1252, 1267, 1282, 1297, 4356  
`\csexpandonce{@tag}` ..... 658  
`\csnumdef` ..... 760, 762  
`\csnumgdef` ..... 2047  
`\csuse` 1333–1335, 1542, 1549–1551,  
1583, 1585–1587, 1589, 1638–  
1640, 1681–1683, 2015, 2021,  
2028, 2032, 2034, 2037, 2041, 2045  
`\ctab` ..... 3227, 3752, 3843  
`\ctabtext` ..... 3231, 3762, 3847
- D**
- `\D@footnote` ... 3327, 3408, 3425, 3444  
`\date` ..... 3875, 4698  
`\dcolerr` ..... 3283, 3295  
`\dcoli` ... 3253, 3285, 3286, 3653, 3658  
`\dcolii` ..... 3254, 3286  
`\dcoliii` ..... 3255, 3286  
`\dcoliv` ..... 3256, 3287  
`\dcolix` ..... 3261, 3288  
`\dcolv` ..... 3257, 3287  
`\dcolvi` ..... 3258, 3287  
`\dcolvii` ..... 3259, 3288  
`\dcolviii` ..... 3260, 3288  
`\dcolx` ..... 3262, 3288  
`\dcolxi` ..... 3263, 3289  
`\dcolxii` ..... 3264, 3289  
`\dcolxiii` ..... 3265, 3289  
`\dcolxiv` ..... 3266, 3290  
`\dcolxix` ..... 3271, 3291  
`\dcolxv` ..... 3267, 3290  
`\dcolxvi` ..... 3268, 3290  
`\dcolxvii` ..... 3269, 3291  
`\dcolxviii` ..... 3270, 3291  
`\dcolxx` ..... 3272, 3291  
`\dcolxxi` ..... 3273, 3292  
`\dcolxxii` ..... 3274, 3292  
`\dcolxxiii` ..... 3275, 3292  
`\dcolxxiv` ..... 3276, 3293  
`\dcolxxix` ..... 3281, 3294  
`\dcolxxv` ..... 3277, 3293  
`\dcolxxvi` ..... 3278, 3293  
`\dcolxxvii` ..... 3279, 3294  
`\dcolxxviii` ..... 3280, 3294  
`\dcolxxx` ..... 3282, 3294  
`\DeclareOption` ..... 6, 7  
Dekker, Dirk-Jan ..... 6, 35, 204  
`\Dend` ..... 2341, 2349  
`\Dendnote` ..... 15, 2328  
`\Dfootgroup` ..... 2112  
`\Dfootins` 1457, 2112, 2123, 2163, 2176  
`\Dfootnote` 14, 1163, 2936, 3330, 3400,  
3408, 3417, 3425, 3444, 3993, 4028  
`\Dfootstart` ..... 2112  
`\dimen` 603, 604, 606–608, 610, 1467,  
1472, 1496–1498, 1501, 1602–  
1604, 1619, 1622, 1667, 1693,  
2683, 2688, 2689, 2704, 2707,  
2750, 2753, 2798–2800, 2803, 4097  
`\dimen@` ..... 2067, 2069  
`\disablel@dtabfeet` .....  
..... 3459, 3476, 3490, 3504,  
3517, 3532, 3660, 3667, 3672,  
3680, 3685, 3693, 3711, 3727, 3850  
`\displaystyle` ..... 3346, 3461,  
3463, 3492, 3494, 3519, 3521,  
3660, 3672, 3685, 3773, 3825, 3826  
`\displaywidowpenalty` ..... 779  
`\divide` ..... 942,  
955, 1498, 1603, 2800, 4389, 4415  
`\do@actions` ..... 843, 870  
`\do@actions@fixedcode` ..... 890, 903  
`\do@actions@next` ..... 870

- `\do@ballast` ..... 844, 858  
`\do@line` ..... 763, 804  
`\do@linehook` ..... 808, 825  
`\do@lockoff` ..... 527  
`\do@lockoffL` ..... 527  
`\do@lockon` ..... 498  
`\do@lockonL` ..... 498  
`\documentclass` ..... 3861,  
     3917, 4057, 4181, 4335, 4598, 4786  
`\doedindexlabel` 3034, 3054, 3079, 3320  
`\doendnotes` ..... 24, 2395  
`\doreinxtrafeeti` ... 2116, 2899, 2920  
`\doreinxtrafeetii` .. 2117, 2119, 2146  
`\dosplits` ..... 1594  
`\dots` 3225, 3995, 4030, 4153, 4156, 4313  
`\doublecolon` ..... 3944, 3953  
Downes, Michael ..... 34, 100, 102  
`\doxtrafeet` ..... 2100  
`\doxtrafeeti` ..... 2101, 2899, 2915  
`\doxtrafeetii` ..... 2102, 2104  
`\dp` ..... 1314, 1515,  
     1529, 2067, 2087, 2821, 2834, 4637  
`\dummy@edtext` ..... 639, 652  
`\dummy@ref` ..... 554, 564  
`\dummy@text` ..... 638, 651
- E**
- `\E@footnote` ... 3327, 3409, 3426, 3445  
`\edaftertab` .....  
     .. 31, 171, 3233, 3700, 3734, 3752  
`edarrayc` (environment) ..... 29, 3842  
`edarrayl` (environment) ..... 29, 3842  
`edarrayr` (environment) ..... 29, 3842  
`\EDATAB` ..... 3792, 3800  
`\edatab` ..... 3234, 3798  
`\edatabell` ..... 3235, 3782  
`\edatleft` ..... 31, 3236, 3584  
`\edatright` ..... 31, 3237, 3592  
`\edbeforetab` .....  
     .. 31, 171, 3232, 3700, 3734, 3752  
`\edfilldimen` .....  
     .... 3606, 3616, 3617, 3619, 3643  
`\edfont@info` ..... 694, 697, 701  
`\EDINDEX` ..... 3313  
`\edindex` .... 28, 3041, 3077, 3313,  
     3402, 3410, 3419, 3427, 3440,  
     3452, 3469, 3485, 3499, 3512, 3527  
`\edindexlab` ..... 28, 3029, 3035, 3038  
`\EDLABEL` ..... 3311
- `\edlabel` ..... 24, 646, 2225, 3035,  
     3311, 3430, 3449, 3451, 3468,  
     3484, 3498, 3511, 3526, 3659,  
     3666, 3671, 3679, 3684, 3692, 3898  
`\edmakelabel` ..... 25, 2315  
`\edpageref` ..... 24, 647, 2276  
`\edrowfill` . 30, 3240, 3431, 3434, 3641  
`\EDTAB` ..... 3788, 3824  
`\edtabcolsep` ..... 30, 3375,  
     3464, 3481, 3494, 3508, 3522,  
     3537, 3617, 3674, 3687, 3696, 3810  
`\EDTABINDENT` ..... 3805, 3818  
`\edtabindent` ..... 3648,  
     3652, 3657, 3668, 3681, 3694, 3814  
`\EDTABtext` ..... 3832  
`edtabularc` (environment) .... 29, 3846  
`edtabularl` (environment) .... 29, 3846  
`edtabularr` (environment) .... 29, 3846  
`\EDTEXT` ..... 3307  
`\edtext` ..... 14, 652, 673,  
     2453–2455, 2562, 2667, 3307,  
     3413, 3439, 3890, 3891, 3897,  
     3982, 3985, 3987, 3989, 3991,  
     3992, 3997, 3999, 4001, 4003,  
     4005, 4007, 4009, 4011, 4013,  
     4018, 4020, 4022, 4024, 4026,  
     4027, 4032, 4034, 4036, 4038,  
     4040, 4042, 4044, 4046, 4048,  
     4123, 4149, 4151–4153, 4166,  
     4251, 4254, 4262, 4270, 4276,  
     4282, 4288, 4289, 4291, 4296,  
     4299, 4302, 4308, 4310, 4316,  
     4325, 4498, 4499, 4509, 4516,  
     4519, 4522, 4523, 4533, 4534,  
     4538, 4540, 4544, 4546, 4548,  
     4552, 4554, 4555, 4558, 4561,  
     4563, 4569, 4582, 4711, 4713,  
     4715, 4717, 4719–4721, 4723,  
     4724, 4727, 4736, 4738, 4741,  
     4744–4746, 4748–4752, 4754,  
     4755, 4758, 4761, 4763, 4765–  
     4770, 4848, 4853, 4895, 4920,  
     4921, 4923, 4925, 4927, 4929,  
     4930, 4932–4934, 4938, 4940,  
     4942, 4943, 4946, 4947, 4951–  
     4955, 4957, 4959, 4962, 4965–4967  
`\edvertdots` ..... 32, 3239, 3603  
`\edvertline` ..... 32, 3238, 3601  
`\Eend` ..... 2345, 2349  
`\Eendnote` ..... 15, 2328

- \Efootgroup ..... 2113
- \Efootins 1458, 2113, 2124, 2164, 2177
- \Efootnote ..... 14, 1163,  
2937, 3331, 3401, 3409, 3418,  
3426, 3445, 3998, 4000, 4002,  
4004, 4006, 4008, 4010, 4012, 4014
- \Efootstart ..... 2113
- \emergencystretch ..... 4500, 4694
- \empty 19, 106, 147, 150, 307, 308, 345,  
667, 682, 692, 706, 710, 716,  
747, 893, 951, 967, 1086–1088,  
1099, 1131, 2227, 3157, 4091, 4862
- \enablel@dtabfeet ..... 3708,  
3724, 3742, 3750, 3760, 3768, 3850
- \end@lemmas .....  
637, 667, 668, 682, 683, 4862, 4863
- \endashchar ..... 19, 1338, 1410, 2387
- \endgraf ..... 758, 798, 802, 4084
- \endline@num ..... 333, 571, 577
- \endlock 13, 627, 645, 3199, 3203, 3205
- \endminipage ..... 2971
- \endnumbering .....  
.. 9, 118, 140, 162, 172, 3900,  
4052, 4174, 4328, 4586, 4775, 4969
- \endpage@num ..... 333, 570, 577
- \endprint ..... 24, 2349, 2398
- \endstanzaextra ..... 22, 3192, 4829
- \endsub ..... 12, 603, 644, 4105
- \endsubline@num ..... 333, 572, 578
- \enskip 2350, 3928, 3940, 3953, 3960,  
4089, 4092, 4212, 4218, 4480,  
4488, 4660, 4673, 4680, 4802, 4807
- \enspace ..... 2350, 2631,  
2724, 2769, 2842, 3927, 3952,  
3959, 4189, 4217, 4226, 4479,  
4487, 4659, 4669, 4679, 4801, 4806
- environments:
- edarrayc ..... 29, 3842
- edarrayl ..... 29, 3842
- edarrayr ..... 29, 3842
- edtabularc ..... 29, 3846
- edtabularl ..... 29, 3846
- edtabularr ..... 29, 3846
- ledgroup ..... 23, 2989
- ledgroupsize ..... 23, 3002
- minipage ..... 23
- Euclid ..... 7
- \ExecuteOptions ..... 8
- \exit ..... 4191, 4319
- \extensionchars .... 32, 104, 124, 168
- F**
- \f@encoding ..... 701
- \f@family ..... 701
- \f@series ..... 701
- \f@shape ..... 701
- \f@x@l@cks ..... 992, 997
- Fairbairns, Robin ..... 27
- \ffootnote ..... 4354, 4498
- \finishstage ..... 4104, 4105
- \first@linenum@out@false ... 582, 588
- \first@linenum@out@true ..... 582
- \firstlinenum ..... 10, 12, 241
- \firstsublinenum ..... 10, 12, 241
- \fix@page ..... 369, 422
- \flag@end ..... 596, 672, 687, 4867
- \flag@start ..... 596, 664, 679, 4859
- \flagstanza ..... 23, 3208, 4839
- \floatingpenalty ..... 1313
- \flush@notes ..... 768, 1129
- Folkerts, Menso ..... 7
- \fontencoding ..... 1145
- \fontfamily ..... 1145
- \fontseries ..... 1145
- \fontshape ..... 1145
- \footfootmarkA ..... 27, 2586
- \footfudgefiddle 34, 1494, 1498, 2800
- \footgroupA ..... 2902
- \footgroupB ..... 2903
- \footgroupC ..... 2904
- \footins .....  
1423, 2082, 2089, 2093, 2128, 2172
- \footinsA 2165, 2178, 2582, 2902, 2908
- \footinsB 2166, 2179, 2870, 2903, 2909
- \footinsC 2167, 2180, 2884, 2904, 2910
- \footnormal .....  
1454, 1476–1480, 2142, 3974, 3975
- \footnormalX 27, 2672, 2895–2897, 2914
- \footnote ..... 3895, 3905, 3984
- \footnoteA ..... 27, 2576, 2946
- \footnoteB . 27, 2861, 2947, 3906–3908
- \footnoteC ..... 27, 2879, 2948
- \footnoterule .. 1445, 2092, 2643, 2986
- \footnotesize ..... 1141, 2450,  
2451, 4469, 4470, 4606, 4607, 4792
- \footparagraph 16, 1482, 3972, 4095,  
4231, 4466, 4467, 4686–4688, 4795
- \footparagraphX ..... 27, 2784, 3868
- \footspitskips ..... 1308,  
1311, 1511, 1525, 1627, 1670,  
2616, 2712, 2758, 2817, 2830, 4623

`\footstartA` ..... 2902  
`\footstartB` ..... 2903  
`\footstartC` ..... 2904  
`\footthreecol` ..... 16, 1607, 3971  
`\footthreecolX` ..... 27, 2739  
`\foottwocol` ..... 16, 1655, 3970, 4232  
`\foottwocolX` ..... 27, 2693  
`\foottwocolX` ..... 2693  
`\forcsvlist` 1150, 1165, 1180, 1195,  
1210, 1232, 1247, 1262, 1277, 1292  
`\fullstop` .....  
19, 297, 1338, 1407, 1409, 1411,  
1413, 2386, 2390, 4217, 4226, 4872

## G

`\g@addto@macro` ..... 2134, 2135,  
2143, 2146, 2149, 2151, 2159,  
2915, 2920, 2923, 2926, 3041, 3077  
Gädeke, Nora ..... 8  
`\get@linelistfile` ..... 341, 357  
`\getline@num` ..... 812, 839  
`\gfootnote` ..... 4363, 4499  
`\ggfootfmt` ..... 4082, 4096  
`\ggfootstart` ..... 4099, 4100  
`\gl@p` 319, 348, 349, 668, 683, 696, 896,  
897, 1092, 1096, 1132, 2230, 4863  
`\gl@poff` ..... 319, 320  
`\goodbreak` ..... 4228

## H

`\hangafter` .... 3188, 3937, 4103, 4188  
`\hangingsymbol` ..... 23, 3131, 3138  
`\hb@xt@` ..... 820,  
822, 832, 834, 3643, 3648, 3652,  
3657, 3668, 3681, 3694, 3771, 3773  
`\hfilneg` ..... 1596  
`\Hilfsbox` ..... 3248  
`\hilfsbox` ..... 3248, 3303, 3304,  
3346, 3358, 3448, 3461, 3478,  
3492, 3506, 3519, 3534, 3659,  
3661, 3666, 3670, 3671, 3673,  
3679, 3683, 3684, 3686, 3692, 3695  
`\hilfscount` .... 3248, 3809–3811, 3817  
`\HILFSskip` ..... 3802  
`\Hilfsskip` ..... 3649,  
3653, 3654, 3658, 3661, 3662,  
3669, 3670, 3673–3675, 3682,  
3683, 3686–3688, 3695–3697,  
3802, 3808, 3810, 3816, 3820, 3821

`\hilfsskip` .....  
. 3248, 3447, 3448, 3463, 3480,  
3494, 3508, 3521, 3536, 3819–3821  
`\hphantom` ..... 3223  
`\Hy@temp@A` ..... 3065, 3066  
`\HyInd@ParenLeft` ..... 3066  
`\hyphenation` ..... 4699  
`\hyphenpenalty` ..... 4065

## I

`\if@fcolmade` ..... 2198, 2207  
`\if@firstcolumn` 973, 1048, 2200, 2509  
`\if@led@nofoot` ..... 2155, 2185  
`\if@nobreak` ..... 732  
`\ifautopar` ..... 749, 782  
`\ifbypage@` 176, 416, 427, 875, 1370, 4072  
`\ifbypstart@` ..... 176, 764, 1395  
`\ifcsdef` ..... 1584, 2026  
`\ifcseempty` 1333, 1549, 1586, 1638, 1681  
`\ifcsstring` ..... 2030  
`\ifdim` 604, 606, 608, 610, 2534, 3303, 3809  
`\iffirst@linenum@out@` ..... 582, 586  
`\ifhbox` ..... 1575, 1580  
`\ifhmode` ..... 2548, 2555, 2597, 2604  
`\ifinlist` . 1151, 1152, 1160, 1161,  
1166, 1167, 1175, 1176, 1181,  
1182, 1190, 1191, 1196, 1197,  
1205, 1206, 1211, 1212, 1220,  
1221, 1233, 1234, 1242, 1243,  
1248, 1249, 1257, 1258, 1263,  
1264, 1272, 1273, 1278, 1279,  
1287, 1288, 1293, 1294, 1302, 1303  
`\ifinserthangingsymbol` ..... 3134  
`\ifinstanza` ..... 749, 799, 3131, 3137  
`\ifl@d@dash` ..... 1342, 1410, 2387  
`\ifl@d@elin` ..... 1342,  
1388, 1412, 1413, 2377, 2389, 2390  
`\ifl@d@esl` ..... 1342, 1413, 2390  
`\ifl@d@pnum` .....  
1342, 1376, 1407, 1411, 2365, 2388  
`\ifl@d@ssub` ..... 1342, 1409, 2386  
`\ifl@dend@` ..... 2317, 2323  
`\ifl@dmemoir` ..... 22, 3314  
`\ifl@dpairing` ..... 108, 144, 1396  
`\ifl@dskipnumber` ..... 630, 934  
`\ifl@dstartendok` ..... 3613, 3623  
`\ifledfinal` ..... 5, 13, 32  
`\ifledplinum` ..... 1721, 4668  
`\ifledRcol` ..... 108, 785, 1397  
`\ifleftnoteup` ..... 2486, 2500

- \ifnolinenum . . . . . 4068, 4087
- \ifnoteschanged@ . . . . . 154, 337
- \ifnumberedpar@ . . . . . 721, 742,
  - 754, 1153, 1168, 1183, 1198,
  - 1213, 1235, 1250, 1265, 1280,
  - 1295, 2330, 2331, 2334, 2335,
  - 2338, 2339, 2342, 2343, 2346,
  - 2347, 2405, 2461, 2467, 2473,
  - 2561, 2567, 2655, 2666, 4355, 4364
- \ifnumbering . . . . . 107,
  - 116, 141, 164, 179, 738, 751, 791
- \ifnumberingR . . . . . 108, 786
- \ifnumberline . . . . . 689, 840, 845, 933
- \ifnumberpstart . . . . . 729, 749, 771, 799
- \ifnumequal . . . . . 765, 1583, 1585, 2028
- \ifodd . . . . . 983, 1058, 2519, 4442
- \ifpst@rtedL . . . . . 108
- \ifpstartnum . . . . . 1069, 1072, 1077
- \ifreportnoidxfile . . . . . 3047
- \ifrightrightnoteup . . . . . 2458, 2495
- \ifshowindexmark . . . . . 3060, 3093
- \ifsidepstartnum . . . . . 749, 1043
- \ifstreempty . . . . . 1710, 1756
- \ifstrequal . . . . . 1731,
  - 1776, 1796, 1816, 1836, 1856,
  - 1877, 1897, 1917, 1943, 1963, 1994
- \ifsublines@ . . . . .
  - . 295, 324, 405, 440, 445, 451,
  - 466, 484, 493, 507, 529, 576,
  - 578, 846, 882, 937, 2249, 2274, 4384
- \iftoggle . . . . . 1333,
  - 1549, 1638, 1681, 2014, 2019, 2024
- \ifvbox . . . . . 761, 2062, 2130
- \ifvmode . . . . . 2231
- \ifvoid . . . . . 2082,
  - 2109–2113, 2120–2124, 2128,
  - 2144, 2147, 2152, 2160–2167,
  - 2172–2180, 2902–2904, 2908–
  - 2910, 2918, 2921, 2927, 2939–
  - 2943, 2950–2952, 2974, 2996, 3023
- \IMM . . . . . 4832, 4833
- \indexentry . . . . . 3086
- \initnumbering@reg . . . . . 115
- \inplaceoflemmaseparator . . . . . 18, 1942
- \inplaceoflemmaseparator@A . . . . . 1936
- \inplaceoflemmaseparator@B . . . . . 1937
- \inplaceoflemmaseparator@C . . . . . 1938
- \inplaceoflemmaseparator@D . . . . . 1939
- \inplaceoflemmaseparator@E . . . . . 1940
- \inplaceoflemmaseparator@Z . . . . . 1941
- \inplaceofnumber . . . . . 17, 1849, 1933
- \inplaceofnumber@A . . . . . 1849
- \inplaceofnumber@B . . . . . 1850
- \inplaceofnumber@C . . . . . 1851
- \inplaceofnumber@D . . . . . 1852
- \inplaceofnumber@E . . . . . 1853
- \inplaceofnumber@Z . . . . . 1854
- \InputIfFileExists . . . . . 358
- \insert . . . . . 1306,
  - 1508, 1625, 1668, 2120–2124,
  - 2128, 2130, 2148, 2614, 2710,
  - 2756, 2814, 2908–2910, 2922, 4620
- \insert@count . . . . . 552, 553,
  - 598, 662, 677, 1151, 1152,
  - 1156, 1160, 1161, 1166, 1167,
  - 1171, 1175, 1176, 1181, 1182,
  - 1186, 1190, 1191, 1196, 1197,
  - 1201, 1205, 1206, 1211, 1212,
  - 1216, 1220, 1221, 1233, 1234,
  - 1238, 1242, 1243, 1248, 1249,
  - 1253, 1257, 1258, 1263, 1264,
  - 1268, 1272, 1273, 1278, 1279,
  - 1283, 1287, 1288, 1293, 1294,
  - 1298, 1302, 1303, 2464, 2470,
  - 2476, 2570, 2658, 4358, 4367, 4857
- \inserthangingsymbol . . . . . 820, 3135
- \inserthangingsymbolfalse . . . . . 816
- \inserthangingsymboltrue . . . . . 814
- \inserthangingsymbol . . . . . 3134
- \insertlines@list . . . . .
  - . . . 147, 328, 354, 560, 1088, 1092
- \insertparafootsep . . . . . 1545, 1582, 2840
- \inserts@list . . . . . 746, 1083,
  - 1086, 1096, 1131, 1132, 1151,
  - 1152, 1155, 1160, 1161, 1166,
  - 1167, 1170, 1175, 1176, 1181,
  - 1182, 1185, 1190, 1191, 1196,
  - 1197, 1200, 1205, 1206, 1211,
  - 1212, 1215, 1220, 1221, 1233,
  - 1234, 1237, 1242, 1243, 1248,
  - 1249, 1252, 1257, 1258, 1263,
  - 1264, 1267, 1272, 1273, 1278,
  - 1279, 1282, 1287, 1288, 1293,
  - 1294, 1297, 1302, 1303, 2463,
  - 2469, 2475, 2569, 2657, 4357, 4366
- \instanzafalse . . . . . 3133, 3205
- \instanzatrue . . . . . 3194
- \interfootnotelinepenalty . . . . . 1312
- \interlinepenalty 779, 1116, 1312, 3199

- `\interparanoteglua` ..... 3355, 3383, 3387, 3392, 3453,  
                   ..... 17, 1978, 1983, 1985, 4199  
`\ipn@skip` ..... 1979, 1982, 1984  
`\itshape` ..... 4104, 4213, 4219, 4227
- J**
- Jayaditya ..... 8
- K**
- Kabelschacht, Alois ..... 87  
 Krukov, Alexej ..... 64
- L**
- `\l@d@wrindexhyp` ..... 3058, 3091  
`\l@d@add` ..... 711, 713, 717, 719  
`\l@d@dashfalse` ..... 1351, 1369, 2360  
`\l@d@dashtrue` .....  
                   1373, 1379, 1391, 2363, 2368, 2380  
`\l@d@elinfalse` ..... 1347, 1376, 2365  
`\l@d@elintrue` .. 1376, 1378, 2365, 2367  
`\l@d@end` .. 2317, 2319, 2320, 2326,  
                   2329, 2333, 2337, 2341, 2345, 2402  
`\l@d@err@UnequalColumns` ..... 3389  
`\l@d@eslfalse` .....  
                   ..... 1349, 1385, 1388, 2374, 2377  
`\l@d@esltrue` ... 1388, 1390, 2377, 2379  
`\l@d@index` ..... 3043, 3045, 3316  
`\l@d@makecol` ..... 2074, 2137, 2216  
`\l@d@nums` 661, 694, 697, 705, 706, 719,  
                   1155, 1170, 1185, 1200, 1215,  
                   1237, 1252, 1267, 1282, 1297,  
                   2330, 2334, 2338, 2342, 2346, 4356  
`\l@d@pnumfalse` ..... 1343, 1369, 2360  
`\l@d@pnumtrue` ..... 1372, 2362  
`\l@d@reinserts` ..... 2127, 2138  
`\l@d@section` ..... 2326, 2349  
`\l@d@set` ..... 473, 624  
`\l@d@ssubfalse` ..... 1345, 1381, 2370  
`\l@d@ssubtrue` ..... 1383, 2372  
`\l@d@wrindexm@m` ..... 3057, 3058  
`\l@d@ampcount` ..... 3243,  
                   3385, 3387, 3392, 3657, 3667,  
                   3668, 3680, 3681, 3716, 3732, 3770  
`\l@d@begin@stack` .... 3109, 3119–3121  
`\l@d@bfnote` ..... 2562, 2566  
`\l@d@checkcols` ..... 3342, 3354, 3382  
`\l@d@checkstartend` ..... 3612, 3623  
`\l@d@chset@num` ..... 372, 375, 473  
`\l@d@colcount` ..... 3243, 3285,  
                   3297, 3298, 3341, 3343, 3353,  
                   3355, 3383, 3387, 3392, 3453,  
                   3455, 3470, 3472, 3486, 3487,  
                   3500, 3501, 3513, 3514, 3528,  
                   3529, 3579, 3580, 3657, 3667,  
                   3668, 3680, 3681, 3712, 3714,  
                   3728, 3730, 3770, 3776, 3806, 3815  
`\l@d@collect@@body` ..... 3111, 3118  
`\l@d@collect@body` .....  
                   .... 3106, 3842–3844, 3846–3848  
`\l@d@colwidth` .... 3285, 3303, 3304,  
                   3447, 3578, 3643, 3669, 3682,  
                   3771, 3773, 3778, 3787, 3808, 3809  
`\l@d@csnote` ..... 2455, 2458  
`\l@d@csnotetext` ..... 826, 2481,  
                   2502, 2507, 2510, 2512, 2520, 2522  
`\l@d@doreinextrafeet` 2115, 2129, 2135  
`\l@d@dofootinsert` ..... 2075, 2080  
`\l@d@dextrafeet` ..... 2097, 2100, 2134  
`\l@d@dedbeginmini` .... 2149, 2929, 2932  
`\l@d@dedendmini` ..... 2151, 2930, 2932  
`\l@d@emptyd@ta` ..... 809, 826  
`\l@d@end@close` ..... 2319, 2395  
`\l@d@end@false` ..... 2317, 2320  
`\l@d@end@open` ..... 2319, 2324  
`\l@d@end@stuff` ... 125, 169, 2322, 2401  
`\l@d@end@true` ..... 2317, 2319  
`\l@d@envbody` .... 3101, 3104, 3107–3109  
`\l@d@fambeginmini` ... 2923, 2929, 2945  
`\l@d@famendmini` ..... 2926, 2930, 2945  
`\l@d@feetbeginmini` .....  
                   ..... 2929, 2966, 2992, 3019  
`\l@d@feetendmini` 2929, 2977, 2999, 3026  
`\l@d@getline@margin` ..... 202  
`\l@d@getlock@disp` ..... 246, 274  
`\l@d@getref@num` ..... 2276,  
                   2277, 2279, 2280, 2282, 2283, 2290  
`\l@d@getsidenote@margin` ..... 2412  
`\l@d@gobblearg` ..... 3333, 3414–3418  
`\l@d@gobbledarg` ..... 3333, 3397–3401  
`\l@d@label@parse` ..... 2296, 2299  
`\l@d@dld@ta` ... 820, 826, 972, 1049, 1061  
`\l@d@lp@rbox` .... 832, 2443, 2485, 2491  
`\l@d@lsn@te` ..... 821, 831  
`\l@d@lsnote` ..... 2453, 2458  
`\l@d@make@labels` 2235, 2238, 2260, 2269  
`\l@d@memoirfalse` ..... 23  
`\l@d@memoirtrue` ..... 23  
`\l@d@modforcritext` ..... 3395, 3851  
`\l@d@modforedtext` ..... 3412, 3854

- `\l@dnnullfills` ..... 3429,  
3701, 3719, 3735, 3745, 3753, 3763
- `\l@dnumpstartsL` ..... 108, 130
- `\l@dold@footnotetext` .... 2542, 2544
- `\l@dold@xympar` ..... 2403
- `\l@doldold@footnotetext` . 2559, 2574
- `\l@dp@rsefootspec` ..... 1352
- `\l@dpairingfalse` ..... 108
- `\l@dpairingtrue` ..... 108
- `\l@dparsedendline` ..... 1352, 4630
- `\l@dparsedendpage` ..... 1352
- `\l@dparsedendsub` ..... 1352
- `\l@dparsedstartline` .....  
..... 1352, 4625, 4630, 4632
- `\l@dparsedstartpage` . 1352, 4628, 4633
- `\l@dparsedstartsub` ..... 1352
- `\l@dparsefootspec` ..... 1352, 4624
- `\l@dpush@begins` ..... 3115, 3119
- `\l@drd@eta` ... 822, 826, 972, 1051, 1059
- `\l@dref@undefined` .....  
..... 2276, 2279, 2282, 2285
- `\l@drestorefills` ..... 3429,  
3705, 3721, 3739, 3747, 3757, 3765
- `\l@drestoreforcritext` ... 3395, 3852
- `\l@drestoreforedtext` .... 3412, 3855
- `\l@drp@rbox` ..... 834, 2443, 2494
- `\l@drsn@te` ..... 823, 831
- `\l@drsnote` ..... 2454, 2458
- `\l@dssetmaxcolwidth` .. 3302, 3348, 3360
- `\l@dskipnumberfalse` ..... 630, 935
- `\l@dskipnumbertrue` ..... 630, 926
- `\l@dstartendokfalse` . 3627, 3631, 3635
- `\l@dstartendoktrue` ..... 3625
- `\l@dtabaddcols` ..... 3611, 3642
- `\l@dtabnoexpands` ..... 653, 3216
- `\l@dunboxmpfoot` 2975, 2983, 2997, 3024
- `\l@dunhbox@line` ..... 804
- `\l@dzeropenalties` ..... 757, 777
- Lück, Uwe ..... 6
- `\label` ..... 25, 1416, 1430, 3883
- `\label@refs` ..... 2228, 2230, 2235,  
2238, 2246, 2247, 2251, 2253, 2255
- `\labelref@list` . 2221, 2227, 2230, 2274
- `\labelrefsparseline` ..... 2241
- `\labelrefsparsesubline` ..... 2241
- `\last@page@num` ..... 422
- `\lastbox` ... 797, 811, 1537, 1574, 1579
- `\lastkern` ..... 2534
- `\lastskip` ..... 603, 607
- Lavagnino, John ..... 5, 7
- `\lbreak` ..... 4833, 4896
- `\ldots` ..... 4733, 4735, 4761
- Leal, Jeronimo@Leal, Jerónimo ..... 6
- `\led@err@AutoparNotNumbered` ....  
..... 66, 787, 792
- `\led@err@HighEndColumn` .... 93, 3632
- `\led@err@LineationInNumbered` 43, 180
- `\led@err@LowStartColumn` ... 93, 3628
- `\led@err@NumberingNotStarted` 27, 158
- `\led@err@NumberingShouldHaveStarted`  
..... 27, 171
- `\led@err@NumberingStarted` ... 27, 117
- `\led@err@PendNoPstart` ..... 66, 755
- `\led@err@PendNotNumbered` .... 66, 752
- `\led@err@PstartInPstart` .... 66, 743
- `\led@err@PstartNotNumbered` .. 66, 739
- `\led@err@ReverseColumns` ... 93, 3636
- `\led@err@TooManyColumns` ... 93, 3299
- `\led@err@UnequalColumns` ..... 93
- `\led@mess@NotesChanged` ..... 33, 155
- `\led@mess@SectionContinued` .. 41, 167
- `\led@warn@BadAction` ..... 80, 928
- `\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, 2435
- `\led@warn@BadSubblockdisp` .... 46, 278
- `\led@warn@DuplicateLabel` ... 82, 2263
- `\led@warn@NoIndexFile` ..... 91, 3048
- `\led@warn@NoLineFile` ..... 54, 363
- `\led@warn@NoMarginpars` .... 87, 2406
- `\led@warn@RefUndefined` .... 82, 2287
- `\ledfinalfalse` ..... 7
- `\ledfinaltrue` ..... 6
- `\ledfootinsdim` .....  
..... 1454, 1467, 1472, 2683, 2689
- `ledgroup` (environment) ..... 23, 2989
- `ledgroupsize` (environment) . 23, 3002
- `\ledleftnote` ..... 26, 2453
- `\ledlinenum` ..... 290
- `\ledllfill` ..... 822, 836, 3006, 3010
- `\ledlsnotefontsetup` ... 27, 2446, 2484
- `\ledlsnotesep` ..... 27, 832, 2446
- `\ledlsnotewidth` ..... 26, 2446, 2484
- `\Ledmac` ..... 1436



- \ledmac@error ..... 1585, 2028, 2047, 2273, 4072–4074, 4077, 4204, 4411–4413, 4444
  - \ledmac@warning ..... 25, 47, 49, 51, 53, 55, 57, 60, 63, 65, 81, 83, 85, 88, 90, 92, 2246
  - \ledplinenumfalse ..... 4626
  - \ledplinenumtrue 1722, 4624, 4629, 4631
  - \ledrightnote ..... 26, 2453, 3896
  - \ledrlfill ..... 822, 836, 3007, 3014
  - \ledrsnotefontsetup ... 27, 2446, 2493
  - \ledrsnotesep ..... 27, 834, 2446
  - \ledrsnotewidth . 26, 2446, 2493, 3872
  - \ledsetnormalparstuff ..... 1326, 1546, 2630, 2841, 3926, 3958, 4210, 4216, 4477, 4485, 4657, 4665, 4678, 4800, 4805
  - \ledsidenote ..... 26, 2453, 4712
  - \left ..... 3586, 3589, 3594, 3597
  - \leftctab ..... 3656, 3754
  - \lefthyphenmin ..... 4340
  - \leftlinenum ..... 12, 290, 974, 986, 4077, 4446
  - \leftltab ..... 3647, 3736
  - \leftnoteuptrue ..... 2501
  - \leftpstartnum ..... 1043
  - \leftrtab ..... 3651, 3702
  - Leibniz ..... 8
  - \lemma ..... 15, 703, 3995, 4030, 4153, 4156, 4252, 4263, 4288, 4313, 4510, 4516, 4519, 4534, 4761
  - \lemmafnt ..... 4480, 4495, 4660, 4680
  - \lemmaseparator ..... 17, 1869, 1930
  - \lemmaseparator@A ..... 1869
  - \lemmaseparator@B ..... 1870
  - \lemmaseparator@C ..... 1871
  - \lemmaseparator@D ..... 1872
  - \lemmaseparator@E ..... 1873, 1874
  - \lemmaseparator@Z ..... 1875
  - \letsforverteilen ..... 3437, 3462, 3479, 3493, 3507, 3520, 3535
  - Levy, Silvio ..... 198
  - \line .... 2245, 2247, 2248, 2253, 2255
  - \line@list 150, 328, 353, 578, 692, 696
  - \line@list@stuff ..... 124, 168, 584
  - \line@margin .... 202, 979, 1054, 4438
  - \line@num 132, 294, 322, 377, 411, 417, 428, 458, 459, 461, 469, 474, 475, 487, 571, 575, 852, 876, 885, 950, 952, 953, 962, 963, 1585, 2028, 2047, 2273, 4072–4074, 4077, 4204, 4411–4413, 4444
  - \line@set ..... 707, 708
  - \lineation ..... 11, 44, 47, 178, 3966, 4080, 4237, 4464, 4602
  - \linenum ..... 15, 704, 2312, 3332, 3403, 3420, 3446, 4847, 4858
  - \linenum@out ... 581, 587, 589, 591, 592, 595, 597, 600, 605, 609, 612, 617, 624, 627, 628, 634, 2226
  - \linenumberlist ..... 12, 19, 951, 963
  - \linenumberstyle ..... 13, 281, 4911
  - \linenumincrement ..... 10, 12, 241
  - \linenummargin ..... 12, 49, 202, 3967, 4079, 4238, 4465, 4603
  - \linenumr@p ..... 281
  - \linenumrep ..... 281, 294, 1408, 1412, 2273, 2385, 2389, 4872
  - \linenums ..... 4070
  - \linenumsep ..... 12, 290, 1073, 1078, 2448, 2449, 4185, 4196, 4197
  - \lineref .... 24, 648, 2279, 3038, 3910
  - \linewidth ..... 820
  - \list@clear ..... 308, 353–356, 746
  - \list@clearing@reg ..... 340, 352
  - \list@create ..... 307, 328–331, 637, 1083, 2221
  - \listadd .. 1150, 1165, 1180, 1195, 1210, 1232, 1247, 1262, 1277, 1292
  - \lmarpar ..... 4498, 4517
  - \lock@disp ..... 246, 1015, 1019, 1023, 4423, 4427, 4431
  - \lock@off ..... 500, 501, 527, 628
  - \lock@on ..... 498, 627
  - \lockdisp ..... 13, 51, 246
  - Lorch, Richard ..... 8
  - \lp@rbox ..... 4352, 4375
  - \ltab ..... 3228, 3734, 3842
  - \ltabtext ..... 3230, 3744, 3846
  - Luecking, Dan ..... 40
- M
- \m@m@makecolfloats ..... 2057, 2076
  - \m@m@makecolintro ..... 2057
  - \m@m@makecoltext ..... 2057, 2077
  - \m@mdodoreinextrafeet ..... 2135
  - \m@mdoextrafeet ..... 2134
  - \m@mmf@check ..... 2533, 2550, 2599

- \m@mmf@prepare ..... 2530, 2545, 2554, 2580,  
2592, 2603, 2865, 2876, 2883, 2892
  - \m@th ..... 3604
  - \Ma ..... 4342, 4540, 4567, 4576
  - \makeboxofhboxes .....  
.... 1556, 1566, 1571, 2847, 2856
  - \makeindex ..... 3041, 3077
  - \makememindexhook ..... 3041
  - \maketitle ..... 3877, 4701
  - \managestanza@modulo .... 3163, 3180
  - \marg ..... 1934
  - \marginparwidth ..... 2446, 2447
  - \mathchardef ..... 3158, 4832
  - \maxdepth ..... 2078
  - \maxdimen 1512, 1526, 2818, 2831, 4634
  - Mayer, Gyula ..... 8
  - \measurebody .. 3704, 3710, 3738, 3756
  - \measuremcell ..... 3340, 3366
  - \measuremrow ..... 3364, 3715
  - \measuretbody .. 3720, 3726, 3746, 3764
  - \measurercell ..... 3352, 3371
  - \measuretrrow ..... 3369, 3731
  - \medskip ..... 4513
  - \message ..... 42, 123
  - \mid ..... 4567, 4576
  - Middleton, Thomas ..... 8, 52
  - minipage (environment) ..... 23
  - Mittelbach, Frank ..... 7
  - \morenoexpands ... 34, 640, 4348, 4652
  - \moveleft ..... 3649, 3654, 3662
  - \moveright ..... 3675, 3688, 3697
  - \mpAfootgroup ..... 2939
  - \mpAfootins ..... 1224, 2939
  - \mpAfootnote ..... 1230, 2933
  - \mpBfootgroup ..... 2940
  - \mpBfootins ..... 1224, 2940
  - \mpBfootnote ..... 1230, 2934
  - \mpCfootgroup ..... 2941
  - \mpCfootins ..... 1224, 2941
  - \mpCfootnote ..... 1230, 2935
  - \mpDfootgroup ..... 2942
  - \mpDfootins ..... 1224, 2942
  - \mpDfootnote ..... 1230, 2936
  - \mpEfootgroup ..... 2943
  - \mpEfootins ..... 1224, 2943
  - \mpEfootnote ..... 1230, 2937
  - \mpfootgroupA ..... 2950
  - \mpfootgroupB ..... 2951
  - \mpfootgroupC ..... 2952
  - \mpfootinsA ..... 2588, 2950
  - \mpfootinsB ..... 2872, 2951
  - \mpfootinsC ..... 2888, 2952
  - \mpfootnoteA ..... 2588, 2946
  - \mpfootnoteB ..... 2872, 2947
  - \mpfootnoteC ..... 2888, 2948
  - \mpnormalfootgroup ..... 1448, 1470
  - \mpnormalfootgroupX ..... 2648, 2686
  - \mpnormalvfootnote .....  
..... 1317, 1469, 1612, 1660
  - \mpnormalvfootnoteX .....  
..... 2620, 2685, 2698, 2744
  - \mppara@footgroup ..... 1491, 1561
  - \mppara@footgroupX ..... 2793, 2845
  - \mppara@vfootnote ..... 1490, 1521
  - \mppara@vfootnoteX ..... 2792, 2813
  - \mpthreecolfootgroup .... 1613, 1647
  - \mpthreecolfootgroupX ... 2745, 2772
  - \mpthreecolfootsetup .... 1614, 1620
  - \mpthreecolfootsetupX ... 2746, 2748
  - \mptwocolfootgroup ..... 1661, 1691
  - \mptwocolfootgroupX ..... 2699, 2727
  - \mptwocolfootsetup ..... 1662, 1691
  - \mptwocolfootsetupX ..... 2700, 2702
  - \mpvAfootnote ..... 1236, 1240
  - \mpvBfootnote ..... 1251, 1255
  - \mpvCfootnote ..... 1266, 1270
  - \mpvDfootnote ..... 1281, 1285
  - \mpvEfootnote ..... 1296, 1300
  - \mpvfootnoteA ..... 2592
  - \mpvfootnoteB ..... 2876
  - \mpvfootnoteC ..... 2892
  - \multfootsep ..... 27, 2527, 2537
  - \multiplefootnotemarker .....  
..... 2527, 2531, 2532, 2534
- N
- \n@num ..... 548, 634
  - \n@num@reg ..... 548
  - \NeedsTeXFormat ..... 2
  - \new@line ..... 595, 822
  - \newbox ..... 721, 724, 2443,  
2444, 3248, 3250, 3839, 3840, 4352
  - \newcommandx 1329, 1535, 1544, 1629,  
1672, 1709, 1730, 1755, 1775,  
1795, 1815, 1835, 1855, 1876,  
1896, 1916, 1930, 1942, 1962, 1993
  - \newcounter ..... 232,  
234, 236, 238, 727, 859, 2583,  
2867, 2884, 3031, 3166, 3609, 4836

- \newif ..... 5, 22, 107, 108, 110,  
113, 114, 176, 177, 324, 337,  
582, 630, 689, 722, 729, 782,  
1044, 1069, 1342, 1344, 1346,  
1348, 1350, 1721, 2155, 2318,  
2458, 2500, 3132, 3134, 3623, 4068
- \newinsert . 1224–1228, 1456–1458,  
2582, 2593, 2870, 2877, 2886, 2893
- \newlength ..... 290, 3150
- \newlinechar .....  
.... 2328, 2332, 2336, 2340, 2344
- \newparafootfmt .... 4215, 4233, 4266
- \newread ..... 338
- \newtoggle .....  
1702–1707, 1749–1754, 1767, 1768
- \newtwocolfootfmt ..... 4221, 4234
- \newwrite ..... 581, 2317
- \NEXT ..... 3336,  
3341, 3344, 3349, 3350, 3353,  
3356, 3361, 3362, 3365, 3367,  
3368, 3370, 3372, 3373, 3378,  
3541, 3544, 3546, 3547, 3549,  
3551, 3552, 3555, 3557, 3558,  
3560, 3562, 3563, 3566, 3568,  
3569, 3571, 3573, 3574, 3579,  
3581, 3582, 3776, 3779, 3780,  
3785, 3789, 3790, 3806, 3812, 3813
- \Next ..... 3378, 3454,  
3456, 3465, 3466, 3471, 3473,  
3482, 3483, 3486, 3488, 3495,  
3496, 3500, 3502, 3509, 3510,  
3513, 3515, 3523, 3524, 3528,  
3530, 3538, 3539, 3794, 3796, 3797
- \next@action ..... 81, 349,  
865, 873, 874, 879, 880, 888, 897
- \next@actionline .....  
.... 346, 348, 864, 872, 894, 896
- \next@insert .....  
747, 1087, 1090, 1092, 1095, 1099
- \next@page@num .....  
.... 137, 380, 382, 420, 432, 481
- \no@expands ..... 640, 659, 4346
- \noalign ..... 1599
- \nobrak ..... 4650, 4651
- \noendnotes ..... 24,  
2401, 3870, 3920, 4060, 4194, 4338
- \noindent ..... 799, 1041, 1513,  
1527, 1559, 1569, 2819, 2832,  
2850, 2859, 4509, 4516, 4635, 4708
- \nolemmaseparator ..... 18, 1930
- \nolemmaseparatorthisnote@ .... 1768
- \nolinenums ..... 4067, 4069
- \nolinenumsfalse ..... 4070
- \nolinenumstrue ..... 4069
- \nonumberinfootnote ... 17, 1746, 1755
- \nonumberthisnote@ ..... 1767
- \nonumparafootfmt .. 4208, 4209, 4247
- \normal@footnotemarkX ... 2606, 2674
- \normal@pars .....  
... 143, 748, 802, 1327, 1630,  
1673, 2717, 2762, 3932, 3946, 4222
- \normalbfnoteX ..... 2654, 2667
- \normalbodyfootmarkX .... 2611, 2675
- \normalcolor 1450, 1563, 1649, 1696,  
2091, 2650, 2733, 2778, 2853, 2985
- \normalfont 292, 1142, 2528, 2612, 4228
- \normalfootfmt . 1326, 1462, 3925, 3973
- \normalfootfmtX ..... 2629, 2678
- \normalfootfootmarkX .... 2634, 2679
- \normalfootgroup ..... 1446, 1463
- \normalfootgroupX ..... 2645, 2680
- \normalfootnoterule ..... 1445, 1465
- \normalfootnoteruleX 2643, 2681, 2790
- \normalfootstart ... 1415, 1441, 1460
- \normalfootstartX ..... 2637, 2673
- \normalvfootnote ..... 1305, 1461
- \normalvfootnoteX ..... 2613, 2676
- \nospeak ..... 4114, 4125
- \nospeaker ..... 4113
- \note 4236, 4277, 4292, 4297, 4303, 4314
- \notefontsetup ..... 19, 1141,  
1307, 1320, 1495, 1510, 1524,  
1558, 1568, 1626, 1643, 1669,  
1686, 1984, 2349, 2615, 2623,  
2711, 2727, 2757, 2772, 2797,  
2816, 2829, 2849, 2858, 4083, 4622
- \notenumfont ..... 19,  
1142, 2034, 2037, 2041, 2045,  
2349, 2631, 2724, 2769, 2842,  
3927, 3939, 3952, 3959, 3965,  
4217, 4226, 4469, 4479, 4487,  
4606, 4659, 4667, 4679, 4801, 4806
- \noteschanged@false ..... 337, 359
- \noteschanged@true .....  
.... 148, 151, 337, 364, 693, 1089
- \notetextfont 4470, 4481, 4489, 4607,  
4661, 4674, 4681, 4792, 4803, 4808
- \nulledindex .... 3313, 3402, 3419,  
3452, 3469, 3485, 3499, 3512, 3527
- \nullsetzen ..... 3576, 3713, 3729

- \num@lines . 721, 758, 1106, 1112, 1115
  - \numberedpar@false . . . . . 721
  - \numberedpar@true . . . . . 721, 750
  - \numberingfalse . . . . . 107, 142
  - \numberingtrue . . . . . 107, 120, 162
  - \numberit . . . . . 4838, 4843
  - \numberlinefalse . . . . . 11
  - \numberlinetrue . . . . . 11, 690
  - \numberonlyfirstinline . . . . 17, 1702
  - \numberpstartfalse . . . . . 11, 726
  - \numberpstarttrue . . . . . 11, 726
  - \numlabfont . . 19, 290, 4074, 4077, 4204
- O**
- \oarg . . . . . 1747, 1934
  - \oldprintlines . . . . . 4874
  - \one@line . . . . . 721, 810, 811, 822
  - \openout . . . . . 589, 592, 2319
  - \options . . . 1150–1152, 1160, 1161,  
1165–1167, 1175, 1176, 1180–  
1182, 1190, 1191, 1195–1197,  
1205, 1206, 1210–1212, 1220,  
1221, 1232–1234, 1242, 1243,  
1247–1249, 1257, 1258, 1262–  
1264, 1272, 1273, 1277–1279,  
1287, 1288, 1292–1294, 1302, 1303
  - \os . . . . . 4063, 4088, 4157, 4167
  - \overfullrule . . . . . 4339
- P**
- \PackageError . . . . . 26
  - \PackageWarning . . . . . 25
  - \page@action . . . . . 381, 479, 565
  - \page@num . . . . . 333,  
344, 419, 430, 570, 575, 874,  
981, 1056, 1583, 1592, 2517, 4440
  - \page@start . . . . . 601, 2081
  - \pagelinesep . . . . . 28, 3029, 3038
  - \pageno . . . . . 83, 85, 2053
  - \pageparbreak . . . . . 34, 1041
  - \pageref . . . . . 25
  - \pagestyle . . . . . 4244, 4879
  - \par@line . . . . .  
. 721, 759, 1107, 1108, 1111, 1115
  - \para@footgroup . . . . . 1487, 1554
  - \para@footgroupX . . . . . 2789, 2845
  - \para@footsetup . . . . . 1489, 1495
  - \para@footsetupX . . . . . 2795, 2797
  - \para@vfootnote . . . . 1485, 1507, 4619
  - \para@vfootnoteX . . . . . 2787, 2813
  - \parafootfmt . . . . . 1486, 1544, 3957
  - \parafootfmtX . . . . . 2788, 2839
  - \parafootftmsep . . . . 1986, 1987–1992
  - \parafootsep . . . . . 18, 1987
  - \parafootsep@A . . . . . 1987
  - \parafootsep@B . . . . . 1988
  - \parafootsep@C . . . . . 1989
  - \parafootsep@D . . . . . 1990
  - \parafootsep@E . . . . . 1991
  - \parafootsep@Z . . . . . 1992
  - \parafootstart . . . . . 1484, 1503
  - \parafootstartX . . . . . 2786, 2805
  - \pausenumbering . . . . . 11, 161
  - \pend . . 9, 73, 76, 744, 751, 800, 1041,  
3203, 3205, 3899, 4015, 4050,  
4105, 4113, 4132, 4145, 4173,  
4257, 4264, 4285, 4320, 4327,  
4512, 4571, 4585, 4885, 4897, 4902
  - \phantom . . . . . 3222
  - \phantomsection . . . . . 1416, 1430
  - Pigman, III<sup>rd</sup>, G. W. . . . . 192
  - Plato of Tivoli . . . . . 7
  - \postbodyfootmark . . . . . 2595, 2609
  - \postdisplaypenalty . . . . . 780
  - \prebodyfootmark . . . . . 2595, 2607
  - \predisplaypenalty . . . . . 779
  - \prevgraf . . . . . 758
  - \previous@A@number . . . . . 4644
  - \previous@B@number . . . . . 4645
  - \previous@C@number . . . . . 4646
  - \previous@page . . . . . 4628, 4633, 4647
  - \prevline . . . . . 1584, 2027
  - \prevpage@num . . . . . 1582
  - \printendlines . . . . . 2349, 2383
  - \printlinefootnote . . . . .  
1331, 1547, 1636, 1679, 2010, 2013
  - \printlines . . . . 1394, 2037, 2041, 2045,  
3927, 3939, 3952, 3959, 4088,  
4217, 4226, 4479, 4487, 4659,  
4667, 4679, 4801, 4806, 4874, 4909
  - \printnpnum . . . . . 24, 2385, 2388, 2393
  - \printstanzalines . . . . . 4871, 4909
  - \processl@denvbody . . . . .  
. . . . . 3108, 3112, 3113, 3128
  - \ProcessOptions . . . . . 9
  - \protected@write . . . . .  
2237, 3062, 3067, 3070, 3085, 3094
  - \protected@xdef . . . . .  
2578, 2590, 2863, 2874, 2881, 2890

- `\ProvidesPackage` ..... 3  
`\pst@rtedLfalse` ..... 108, 129, 145  
`\pst@rtedLtrue` ..... 108, 165  
`\pstart` ..... 9, 67, 70, 71,  
     76, 726, 799, 1041, 3184, 3886,  
     3981, 4017, 4102, 4107, 4111,  
     4113, 4114, 4188, 4250, 4260,  
     4507, 4515, 4573, 4885, 4893, 4899  
`\pstartline` ..... 762, 765  
`\pstartnum` ..... 1043  
`\pstartnumfalse` ..... 1074, 1081  
`\pstartnumtrue` ..... 772, 1070
- Q**
- `\quad` ..... 4300, 4317
- R**
- `\raggedright` ..... 1634,  
     1677, 2451, 2722, 2767, 3936, 3950  
`\raw@text` ..... 721, 749, 761, 810  
`\rbracket` ..... 19, 1338,  
     1869–1875, 3940, 4212, 4218,  
     4227, 4488, 4650, 4673, 4802, 4807  
`\rd@ta` ..... 4379, 4444  
`\read@linelist` ..... 338, 585  
`\ref` ..... 25, 3911  
`\refstepcounter` ..... 4842  
`\Relax` ..... 3336, 3793, 3800  
`\rem@inder` ..... 963, 965–967  
`\removeboxes` .....  
     .... 1557, 1567, 1571, 2848, 2857  
`\removelastskip` ..... 3453, 3470  
`\RequirePackage` ..... 11, 12  
`\resumenumbering` ..... 11, 161  
`\right` ..... 3587, 3590, 3595, 3598  
`\rightctab` ..... 3665, 3755  
`\rightlinenum` .....  
     12, 290, 976, 984, 4072, 4204, 4443  
`\rightltab` ..... 3678, 3737  
`\rightnoteuptrue` ..... 2459  
`\rightpstartnum` .... 1051, 1059, 1076  
`\rightrtab` ..... 3691, 3703  
`\rightstartnum` ..... 1043  
`\rigidbalance` ... 1594, 1646, 1653,  
     1689, 1700, 2730, 2737, 2775, 2782  
`\rlap` .. 976, 984, 1051, 1059, 4185, 4443  
`\rmarpar` ..... 4499, 4508, 4549, 4577  
`\rmfamily` ..... 3924, 3944, 4091  
Robinson, Peter ..... 6  
`\roman` ..... 3610
- `\rtab` ..... 3226, 3700, 3844  
`\rtabtext` ..... 3229, 3718, 3848
- S**
- Sacrobosco ..... 8  
`\savel@dcnote` ..... 2502  
`\sc@n@list` ..... 964, 966  
`\scf` ..... 4201,  
     4298, 4303, 4306, 4307, 4314, 4315  
Schöpf, Rainer ..... 7  
`\section@num` .....  
     104, 121, 123, 124, 166–168, 2326  
`\select@lemmafont` ..... 642, 1143  
`\select@lemmafont` ..... 20,  
     1143, 1332, 1548, 1637, 1680,  
     2350, 3928, 3940, 3953, 3960,  
     4212, 4218, 4488, 4673, 4802, 4807  
`\sen` ..... 4106,  
     4148, 4149, 4151, 4153, 4158–4163  
`\senspeak` ..... 4107, 4147  
`\set@line` ..... 661, 676, 691, 4856  
`\set@line@action` 374, 464, 471, 482, 567  
`\setl@dlp@rbox` . 2479, 2483, 2510, 2522  
`\setl@drp@rbox` . 2480, 2492, 2512, 2520  
`\setl@drpr@rbox` ..... 2483  
`\setline` ... 13, 63, 613, 650, 765, 4268  
`\setlinenum` ..... 13, 65, 620, 4844  
`\setlp@rbox` ..... 4372, 4377  
`\setmcellcenter` ..... 3511, 3567  
`\setmcellleft` ..... 3484, 3556  
`\setmcellright` ..... 3451, 3545  
`\setmrowcenter` ..... 3565, 3759  
`\setmrowleft` ..... 3554, 3741  
`\setmrowright` ..... 3543, 3707  
`\setprintendlines` ..... 2359, 2384  
`\setprintlines` ..... 1368, 1406  
`\setstanzaindents` ... 21, 3163, 4820  
`\setstanzapenalties` .....  
     .... 22, 3163, 4825, 4826  
`\setstanzavalues` 3153, 3163, 3164, 3214  
`\settccllcenter` ..... 3526, 3572  
`\settccllleft` ..... 3498, 3561  
`\settccllright` ..... 3468, 3550  
`\settoggle` .....  
     1711–1715, 1718, 1757–1761, 1764  
`\setthrowcenter` ..... 3570, 3767  
`\setthrowleft` ..... 3559, 3749  
`\setthrowright` ..... 3548, 3723  
Setzen ..... 3784, 3793, 3795  
Shakespeare, William ..... 195

- `\showlemma` ..... [13](#), [32](#), 666, 681
  - `\sidenote@margin` ..... [2412](#), 2515
  - `\sidenotemargin` ..... [26](#), [2412](#), 4604
  - `\sidenotemmargin` ..... 90
  - `\skip` ..... 1423, 1442, 1449, 1468,  
1473, 1505, 1562, 1648, 1695,  
2089, 2638, 2649, 2684, 2690,  
2732, 2777, 2806, 2810, 2852,  
2984, 4098, 4099, 4613–4615, 4811
  - `\skip@lockoff` ..... 501, [527](#)
  - `\skipnumbering` ..... [14](#), [630](#)
  - `\skipnumbering@reg` ..... [630](#)
  - `\sl` ..... 640
  - `\spacedcolon` ..... 3924, 3928, 3960
  - `\spacefactor` .....  
2535, 2538, 2549, 2555, 2598, 2604
  - `\spaceskip` ..... 1309, 2617, 2713
  - `\speak` ..... 4111, 4134
  - `\speaker` ..... 4188, 4270, 4287, 4322
  - `\splitmaxdepth` ..... 1314
  - `\splitoff` ..... [1594](#)
  - `\splittopskip` .... 807, 1314, 1596,  
1644, 1646, 1651, 1653, 1687,  
1689, 1698, 1700, 2728, 2730,  
2735, 2737, 2773, 2775, 2780, 2782
  - `\spreadmath` ..... [30](#), [3772](#)
  - `\spreadtext` ..... [30](#), [3770](#)
  - `\ss` ..... 3217
  - `\stage` ..... 4102, 4123, 4185, 4191
  - `\stanza` [21](#), [3192](#), 4916, 4919, 4936, 4950
  - `\stanza@count` ..... [3144](#), 3155,  
3158, 3160, 3177, 3189, 3196, 3204
  - `\stanza@hang` ..... [3175](#), 3198
  - `\stanza@line` ..... [3175](#), 3204, 3206
  - `\stanza@modulo` .....  
.... 3167, 3170–3172, 3181, 3196
  - `\stanzaindentbase` .... [21](#), [3144](#),  
3178, 3182, 3187, 3208, 4815, 4839
  - `\startlock` ..... [13](#), [627](#), 645, 3185
  - `\startstanzahook` ..... [23](#), [3192](#), 4842
  - `\startsub` ..... [12](#), [603](#), 644, 4102
  - `\stepcounter` ..... 2577, 2589,  
2862, 2873, 2880, 2889, 3034, 3618
  - `\stepl@dcolcount` .... [3297](#), 3347,  
3359, 3460, 3477, 3491, 3505,  
3518, 3533, 3577, 3777, 3786, 3807
  - `\strip@pt` ..... 1501, 2803
  - `\strip@szacnt` ..... [3153](#)
  - `\sub@action` ..... 390, [491](#), 566
  - `\sub@change` ..... 138,  
384, 385, 391, 441, 443, 446, 448
  - `\sub@lock` ..... 135, [326](#),  
399, 401, 403, 406, 509, 510,  
512, 513, 531, 532, 534, 847,  
918, 920, 921, 923, 998, 1034,  
1036, 1038, 4396, 4457, 4459, 4461
  - `\sub@off` ..... [440](#), 609
  - `\sub@on` ..... [440](#), 605
  - `\subline` ..... 2250–2253
  - `\subline@num` ..... 133, 296,  
297, [323](#), 407, 411, 417, 428,  
452, 453, 455, 467, 485, 572,  
576, 848, 853, 876, 883, 938–  
940, 2274, 4075, 4078, 4385–4387
  - `\sublinenumberstyle` ..... [13](#), [281](#)
  - `\sublinenumincrement` .... [10](#), [12](#), [241](#)
  - `\sublinenumr@p` ..... [281](#)
  - `\sublinenumrep` ..... [281](#),  
297, 1409, 1413, 2274, 2386, 2390
  - `\sublineref` ..... [24](#), 649, [2282](#)
  - `\sublines@false` ... 136, [324](#), 388, 908
  - `\sublines@true` ..... [324](#), 386, 906
  - `\sublock@disp` ..... [272](#),  
1000, 1004, 1008, 4398, 4402, 4406
  - `\sublockdisp` ..... 53, [272](#)
  - `\subsection` ..... 3883
  - `\subsubsection` ..... 2009
  - Sullivan, Wayne ..... 7, 8, 21,  
33, 40, 45, 101, 102, 124, 155, 198
  - `\symlinenum` ..... [17](#), [1724](#)
  - `\symlinenum@A` ..... 1724
  - `\symlinenum@B` ..... 1725
  - `\symlinenum@C` ..... 1726
  - `\symlinenum@D` ..... 1727
  - `\symlinenum@E` ..... 1728
  - `\symlinenum@Z` ..... 1729
  - `\symlinenum` ..... [1721](#), 1724–1729
  - `\sza@penalty` ..... [3175](#), 3202, 3203
- T**
- `\tabellzwischen` ..... [3775](#), 3783
  - `\tabelskip` 3787, 3827–3829, 3835–3837
  - `\tabHilfbox` ..... 3826,  
3828, 3830, 3834, 3836, 3838, [3839](#)
  - `\tabhilfbox` ..... 3825,  
3827, 3829, 3833, 3835, 3837, [3839](#)
  - `\tableofcontents` ..... 3878
  - Tapp, Christian ..... 6
  - `\textbf` ..... 4534, 4839, 4885, 4895

- `\textheight` ..... 2203, 4789  
`\textnormal` ..... 1338–1340,  
     4253, 4271, 4290, 4300, 4301,  
     4309, 4311, 4313, 4317, 4326, 4651  
`\textrm` ..... 4106, 4107, 4298  
`\textsc` ..... 4712  
`\textsf` ..... 3880, 3887  
`\textsuperscript` .....  
     .... 2528, 2586, 2612, 2635, 4535  
`\textwidth` ..... 2962, 3004, 4790  
`\thanks` ..... 3874  
`\theadcolcount` .... 3609, 3616, 3619  
`\thefootnoteA` 27, 2578, 2583, 2586, 2590  
`\thefootnoteB` .. 2863, 2867, 2874, 3867  
`\thefootnoteC` ..... 2881, 2884, 2890  
`\thelabidx` ..... 3035, 3038  
`\thempfn` ..... 2964, 2990, 3017  
`\thempfootnote` .... 2964, 2990, 3017  
`Theodosius` ..... 8  
`\thepage` ..... 595, 2238, 3038  
`\thepageline` .....  
     3037, 3063, 3068, 3071, 3086, 3095  
`\thepstart` ..... 11,  
     726, 728, 749, 799, 1072, 1079, 1403  
`\thepstartL` ..... 1400  
`\thepstartR` ..... 1398  
`\thestanzanum` ..... 4839  
`\thinspace` ..... 1340, 3924, 3944  
`\thr@@` ..... 223, 512, 521, 532,  
     539, 913, 921, 1619, 1622, 1646,  
     1653, 2433, 2750, 2753, 2775, 2782  
`\threecolfootfmt` ... 1609, 1629, 3931  
`\threecolfootfmtX` ..... 2741, 2761  
`\threecolfootgroup` ..... 1610, 1643  
`\threecolfootgroupX` ..... 2742, 2772  
`\threecolfootsetup` ..... 1611, 1617  
`\threecolfootsetupX` ..... 2743, 2748  
`\threecolvfootnote` ..... 1608, 1624  
`\threecolvfootnoteX` ..... 2740, 2755  
`\tiny` ..... 4201  
`\title` ..... 3873, 4697  
`\togglefalse` 1160, 1161, 1175, 1176,  
     1190, 1191, 1205, 1206, 1220,  
     1221, 1242, 1243, 1257, 1258,  
     1272, 1273, 1287, 1288, 1302, 1303  
`\toggletrue` 1151, 1152, 1166, 1167,  
     1181, 1182, 1196, 1197, 1211,  
     1212, 1233, 1234, 1248, 1249,  
     1263, 1264, 1278, 1279, 1293, 1294  
`\tolerance` ..... 1633,  
     1676, 2721, 2766, 3935, 3949, 4224  
`\twocolfootfmt` ..... 1657, 1665, 3945  
`\twocolfootfmtX` ..... 2695, 2716  
`\twocolfootgroup` ..... 1658, 1665  
`\twocolfootgroupX` ..... 2696, 2727  
`\twocolfootsetup` ..... 1659, 1665  
`\twocolfootsetupX` ..... 2697, 2702  
`\twocolvfootnote` ..... 1656, 1665  
`\twocolvfootnoteX` ..... 2694, 2709
- ## U
- `\underbrace` ..... 3224  
`\unhbox` 804, 1538, 1557, 1559, 1567,  
     1569, 1576, 1580, 2848, 2850,  
     2857, 2859, 3829, 3830, 3837, 3838  
`\unkern` ..... 2536  
`\unpenalty` ..... 1541, 1573  
`\unvbox` ..... 811,  
     1319, 1446, 1452, 1523, 1536,  
     1555, 1565, 1605, 2068, 2088,  
     2093, 2106, 2120–2124, 2128,  
     2130, 2148, 2186, 2622, 2646,  
     2652, 2828, 2846, 2855, 2901,  
     2908–2910, 2917, 2922, 2981, 2987  
`\unvxh` 1514, 1528, 1535, 2820, 2833, 4636  
`\usepackage` ..... 3862, 3918, 4058,  
     4182, 4336, 4344, 4599, 4600, 4787  
`\usingcritext` ..... 3850  
`\usingedtext` ..... 3850
- ## V
- `\vAfootnote` ..... 1154, 1158  
`\valign` ..... 1597  
`\value` ..... 3171, 3176, 3615  
`Vamana` ..... 8  
`\variab` ..... 3380, 3706,  
     3722, 3740, 3748, 3758, 3766, 3799  
`\vbadness` ..... 806, 1596  
`\vbfnoteX` ..... 2656, 2661  
`\vBfootnote` ..... 1169, 1173  
`\vbox` ..... 749, 1318,  
     1512, 1522, 1526, 1536, 2065,  
     2085, 2105, 2211, 2215, 2487,  
     2489, 2491, 2496, 2498, 2621,  
     2818, 2827, 2831, 2900, 2916,  
     2959, 3586, 3589, 3594, 3597,  
     3601, 3603, 3604, 3648, 3652,  
     3657, 3668, 3681, 3694, 4375, 4634  
`\vCfootnote` ..... 1184, 1188

|                                       |                                |                                           |                                |
|---------------------------------------|--------------------------------|-------------------------------------------|--------------------------------|
| <code>\vDfootnote</code> . . . . .    | 1199, 1203                     | <code>\xedlabel</code> . . . . .          | <u>3311</u> , 3449             |
| <code>\vEfootnote</code> . . . . .    | 1214, 1218                     | <code>\xedtext</code> . . . . .           | <u>3307</u> , 3439             |
| <code>\vffootnote</code> . . . . .    | 4356, 4360, 4377               | <code>\xleft@appenditem</code> . . . . .  | <u>315</u>                     |
| <code>\vfil</code> . . . . .          | 1597, 2215,                    | <code>\xlineref</code> . . . . .          | <u>25</u> , <u>2279</u>        |
| 3587, 3590, 3595, 3598, 3601, 3604    |                                | <code>\xpagerref</code> . . . . .         | <u>25</u> , <u>2276</u>        |
| <code>\vfootnoteA</code> . . . . .    | 2580                           | <code>\xparafootfmt</code> . . . . .      | 4799, 4810, 4907               |
| <code>\vfootnoteB</code> . . . . .    | 2865                           | <code>\xright@appenditem</code> . . . . . | <u>309</u> , 480,              |
| <code>\vfootnoteC</code> . . . . .    | 2883                           | 481, 483, 490, 492, 494, 496,             |                                |
| <code>\vgfootnote</code> . . . . .    | 4365, 4369, 4379               | 506, 508, 517, 528, 530, 537,             |                                |
| <code>\vl@dbfnote</code> . . . . .    | 2566                           | 550, 551, 560, 574, 1151, 1152,           |                                |
| <code>\vl@dcsnote</code> . . . . .    | 2474, <u>2479</u>              | 1154, 1160, 1161, 1166, 1167,             |                                |
| <code>\vl@dlsnote</code> . . . . .    | 2462, <u>2479</u>              | 1169, 1175, 1176, 1181, 1182,             |                                |
| <code>\vl@drsnote</code> . . . . .    | 2468, <u>2479</u>              | 1184, 1190, 1191, 1196, 1197,             |                                |
| <code>\vnumfootnoteX</code> . . . . . | <u>2665</u> , 2677             | 1199, 1205, 1206, 1211, 1212,             |                                |
| <code>\vrule</code> . . . . .         | 3586, 3589, 3594, 3597, 3601   | 1214, 1220, 1221, 1233, 1234,             |                                |
| <code>\vsize</code> . . . . .         | 1454, 2688, 4097               | 1236, 1242, 1243, 1248, 1249,             |                                |
| <code>\vsplit</code> . . . . .        | 810, 1604, 2186                | 1251, 1257, 1258, 1263, 1264,             |                                |
|                                       |                                | 1266, 1272, 1273, 1278, 1279,             |                                |
|                                       |                                | 1281, 1287, 1288, 1293, 1294,             |                                |
|                                       |                                | 1296, 1302, 1303, 2272, 2462,             |                                |
|                                       |                                | 2468, 2474, 2568, 2656, 4356, 4365        |                                |
|                                       |                                | <code>\xspaceskip</code> . . . . .        | 1309, 2617, 2713               |
|                                       |                                | <code>\xsublineref</code> . . . . .       | <u>25</u> , <u>2282</u>        |
|                                       |                                | <code>\xxref</code> . . . . .             | <u>25</u> , <u>2307</u>        |
|                                       |                                |                                           |                                |
|                                       |                                | <b>Y</b>                                  |                                |
|                                       |                                | <code>\yparafootfmt</code> . . . . .      | 4804, 4891                     |
|                                       |                                |                                           |                                |
|                                       |                                | <b>Z</b>                                  |                                |
| <code>\x@lemma</code> . . . . .       | 668–670, 683–685, 4863–4865    | <code>\z@skip</code> . . . . .            | 1309, 1315, 2617, 2713         |
| <code>\xcritext</code> . . . . .      | <u>3307</u> , 3438             | <code>\zz@@@</code> . . . . .             | <u>2222</u> , 2228, 2309, 2311 |
| <code>\xedindex</code> . . . . .      | <u>3313</u> , 3410, 3427, 3440 |                                           |                                |

## Change History

|                                               |   |                                               |
|-----------------------------------------------|---|-----------------------------------------------|
| v0.1                                          |   | phy. Redefines the command                    |
| General: First public release . . . . .       | 1 | <code>\hangingsymbol</code> to define the     |
| v0.10                                         |   | character. . . . . 1                          |
| General: Corrections to <code>\section</code> |   |                                               |
| and other titles in numbered                  |   | v0.12                                         |
| sections . . . . .                            | 1 | General: For compatibility with led-          |
| v0.11                                         |   | par, possibility to use <code>\autopar</code> |
| General: Makes it possible to add             |   | on the right side. . . . . 1                  |
| a symbol on each verse's hang-                |   | Possibility to number the                     |
| ing, as in French typogra-                    |   | pstart with the commands                      |
|                                               |   | <code>\numberpstarttrue</code> . . . . . 1    |



|                                                                                                                                                                                             |       |         |                                                                                                                                                                |         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| Possibility to number <code>\pstart</code> . . . . .                                                                                                                                        | 11    | v0.18   | General: Debug in lineation command . . . . .                                                                                                                  | 11      |
| <code>\ifledRcol</code> : Added <code>\ifledRcol</code> and <code>\ifnumberingR</code> for/from <code>ledpar</code> . . . . .                                                               | 44    |         | New generic commands to customize footnote display. . . . .                                                                                                    | 17, 108 |
| v0.12.1                                                                                                                                                                                     |       | v0.18.1 | General: <code>\inplaceoflemmaseparator</code> added. . . . .                                                                                                  | 18      |
| General: Don't number <code>\pstarts</code> of stanza. . . . .                                                                                                                              | 1     |         | <code>\lemma</code> can contain commands. . . . .                                                                                                              | 15      |
| The numbering of <code>\pstarts</code> restarts on each <code>\beginnumbering</code> . . . . .                                                                                              | 1     |         | <code>\nonumberthisnote</code> deleted. . . . .                                                                                                                | 17      |
| v0.13                                                                                                                                                                                       |       |         | Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code> . . . . .                                                                         | 15, 88  |
| General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every $n$ verses. . . . .                                                                                | 1, 22 | v0.2    | General: Added <code>tabmac</code> code, and extended indexing . . . . .                                                                                       | 1       |
| <code>\managestanza@modulo</code> : New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every $n$ verses. . . . .                                                     | 156   |         | <code>\ifl@dmemoir</code> : Added <code>\ifl@dmemoir</code> for memoir class having been used . . . . .                                                        | 40      |
| v0.13.1                                                                                                                                                                                     |       |         | <code>\ledmac@error</code> : Added <code>\ledmac@error</code> and replaced error messages . . . . .                                                            | 41      |
| 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     |         | <code>\morenoexpands</code> : Added <code>\l@dtabnoexpands</code> to <code>\no@expands</code> . . . . .                                                        | 69      |
| v0.14                                                                                                                                                                                       |       | v0.2.1  | <code>\@lab</code> : Removed page setting from <code>\@lab</code> . . . . .                                                                                    | 126     |
| General: Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph. . . . .                                                                   | 1     |         | General: Added text about normal labeling . . . . .                                                                                                            | 25      |
| <code>\edlabel</code> : Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph. . . . .                                                    | 124   |         | Bug fixes and match with <code>mempatch v1.8</code> . . . . .                                                                                                  | 1       |
| v0.15                                                                                                                                                                                       |       |         | Major changes to insert code when memoir is loaded . . . . .                                                                                                   | 121     |
| General: Line numbering can be reset at each <code>pstart</code> . . . . .                                                                                                                  | 46    |         | <code>\doxtrafeet</code> : Renamed <code>\doxtrafeet</code> to <code>\l@ddoxtrafeet</code> . . . . .                                                           | 120     |
| New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . . . .                                                                                          | 155   |         | <code>\edlabel</code> : Tweaked <code>\edlabel</code> to get correct page numbers . . . . .                                                                    | 124     |
| Possibility to print <code>\pstart</code> number in side. . . . .                                                                                                                           | 11    |         | <code>\l@d@makecol</code> : Rewrote <code>\@makecol</code> , calling it <code>\l@d@makecol</code> . . . . .                                                    | 119     |
| <code>\affixline@num</code> : Line numbering can be disabled. . . . .                                                                                                                       | 80    |         | <code>\l@ddodoreinxtrafeet</code> : Renamed <code>\dodoreinxtrafeet</code> to <code>\l@ddodoreinxtrafeet</code> . . . . .                                      | 121     |
| <code>\printlines</code> : Line numbering can be reset at each <code>pstart</code> . . . . .                                                                                                | 96    |         | <code>\l@ddofootinsert</code> : Renamed <code>\dofootinsert</code> as <code>\l@ddofootinsert</code> . . . . .                                                  | 120     |
| v0.17                                                                                                                                                                                       |       |         | <code>\m@m@makecolintro</code> : Added <code>\m@m@makecolfloats</code> , <code>\m@m@makecoltext</code> and <code>\m@m@makecolintro</code> . . . . .            | 119     |
| General: New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . . . .                                                                             | 155   |         | <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 . . . . .                                                     | 69  | v0.31  | General: Added remarks about ledmac/parallel package incompatibility . . . . .                          | 35  |
| \zz@@@: Minor change to \zz@@@ .                                                                        | 124 |        |                                                                                                         |     |
| v0.2.2                                                                                                  |     |        |                                                                                                         |     |
| General: Added the Dekker example . . . . .                                                             | 204 | v0.4   | \@iiiminipage: Modified kernel \@iiiminipage and \endminipage to cater for critical footnotes . . . . . | 147 |
| Improved paragraph footnotes .                                                                          | 1   |        | General: Added \showlemma to \edtext (and \critext) . . .                                               | 71  |
| New Dekker example . . . . .                                                                            | 1   |        | Added minipage, etc., support .                                                                         | 1   |
| \footfudgefiddle: Added                                                                                 |     |        | ledgroup: Added ledgroup environment . . . . .                                                          | 148 |
| \footfudgefiddle . . . . .                                                                              | 100 |        | ledgroupsized: Added ledgroupsized environment . . . . .                                                | 149 |
| \l@d@section: Used \providecommand for \@gobblethree to avoid clash with the amsfonts package . . . . . | 130 |        | \footnormal: Added minpage footnote setup to \footnormal . .                                            | 98  |
| \line@list@stuff: Added initial write of page number in \line@list@stuff . . . . .                      | 64  |        | \ifledfinal: Added final/draft options . . . . .                                                        | 40  |
| \para@footsetup: Added                                                                                  |     |        | \l@dfeetendmini: Added                                                                                  |     |
| \footfudgefiddle to                                                                                     |     |        | \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code                                        | 147 |
| \para@footsetup . . . . .                                                                               | 100 |        | \mpEfootins: Added \mpAfootins and friends . . . . .                                                    | 90  |
| \para@footsetupX: Added                                                                                 |     |        | \mpEfootnote: Added \mpAfootnote and friends . . . . .                                                  | 90  |
| \footfudgefiddle to                                                                                     |     |        | \mpfootinsA: Familiar footnotes extended for minipages . . .                                            | 137 |
| \para@footsetupX . . . . .                                                                              | 143 |        | \mpfootinsB: Familiar footnotes extended for minipages . . .                                            | 145 |
| v0.3                                                                                                    |     |        | \mpfootinsC: Familiar footnotes extended for minipages . . .                                            | 145 |
| \@l@reg: Added a bunch of code to \@l for handling \setlinenum                                          | 57  |        | \mpnormalfootgroup: Added                                                                               |     |
| \@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines               | 126 |        | \mpnormalfootgroup . . . . .                                                                            | 97  |
| General: Added the Braonain example . . . . .                                                           | 208 |        | \mpnormalvfootnote: Added                                                                               |     |
| Includes edstanza and more . . .                                                                        | 1   |        | \mpnormalvfootnote . . . . .                                                                            | 93  |
| \ledlinenum: Added \linenumr@p and \sublinenumr@p to \leftlinenum and \rightlinenum . . . . .           | 50  |        | \showlemma: Added \showlemma .                                                                          | 40  |
| \linenumberlist: Added                                                                                  |     |        |                                                                                                         |     |
| \linenumberlist mechanism .                                                                             | 40  | v0.4.1 | \@opxtrafeetii: Added \@opxtrafeetii . . . . .                                                          | 120 |
| \printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines . . . . .                        | 131 |        | General: Added code for changing \doclearpage . . . . .                                                 | 122 |
| \printlines: Added \linenumr@p and \sublinenumr@p to \printlines . . . . .                              | 96  |        | Let ledmac take advantage of memoir's indexing . . . . .                                                | 150 |
| \sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle . . .                                    | 49  |        | Not released. Minor editorial improvements and code tweaks . .                                          | 1   |
| v0.3.1                                                                                                  |     |        | Only change \@footnotetext and \@footnotemark if memoir not used . . . . .                              | 135 |
| General: Not released. Added remarks about the parallel package . . . . .                               | 1   |        |                                                                                                         |     |

|                                      |      |                                          |     |
|--------------------------------------|------|------------------------------------------|-----|
| \addfootins: Added \addfootins       | 122  | \page@start: Made \page@start a          |     |
| \addfootinsX: Added minpage          |      | no-op                                    | 64  |
| setup to \addfootinsX                | 146  | \v1@dbfnote: Changed \l@dbfnote          |     |
| \doxtrafeetii: Changed               |      | and \v1@dbfnote as originals             |     |
| \doxtrafeetii code for easier        |      | could give incorrect markers in          |     |
| extensions                           | 120  | the footnotes                            | 136 |
| \ledfootinsdim: Added \ledfootinsdim | v0.7 |                                          |     |
|                                      | 98   | \@l@reg: Added \@l@reg                   | 57  |
| v0.5                                 |      | \@ref@reg: Added \@ref@reg               | 62  |
| \@footnotetext: Enabled regular      |      | General: Added bits about ledpar         |     |
| \footnote in numbered text           | 136  | package                                  | 35  |
| \@xympar: Eliminated \marginpar      |      | ledmac having been available for         |     |
| disturbance                          | 132  | 2 years, deleted the commented           |     |
| General: Added left and right side   |      | out original edmac texts                 | 1   |
| notes                                | 132  | Maïeul Rouquette new main-               |     |
| Added sidenotes, familiar foot-      |      | tainer                                   | 1   |
| notes in numbered text               | 1    | Made macros of all messages              | 41  |
| v0.5.1                               |      | Replaced all \interAfootnotelinepenalty, |     |
| General: Added moveable side note    | 132  | etc., by just \interfootnotelinepenalty  | 1   |
| Fixed right line numbers killed in   | v0.5 |                                          | 1   |
| \affixline@num: Changed              |      | Tidying up for ledpar and                |     |
| \affixline@num to cater for          |      | ledarab packages                         | 1   |
| sidenotes                            | 80   | \affixline@num: Added skipnu-            |     |
| ledgroupsize: Only change            |      | mering to \affixline@num                 | 80  |
| \hsize in ledgroupsize envi-         |      | \do@actions@fixedcode: Added             |     |
| ronment otherwise page number        |      | \do@actions@fixedcode                    | 79  |
| can be in wrong place                | 149  | \do@actions@next: Added number           |     |
| \l@getsidenote@margin: Added         |      | skipping to \do@actions                  | 78  |
| \sidenotemargin and                  |      | \do@linehook: Added \do@linehook         |     |
| \sidenote@margin                     | 132  | for use in \do@line                      | 76  |
| v0.6                                 |      | \endnumbering: Changed                   |     |
| \@l@reg: Added \fix@page to \@l      | 57   | \endnumbering for ledpar                 | 45  |
| Extended \@l to include the page     |      | \f@x@l@cks: Added \ch@cksub@l@ck,        |     |
| number                               | 57   | \ch@ck@l@ck and \f@x@l@cks               | 82  |
| \@lopR: Added \@pend, \@pendR,       |      | \footsplitskips: Added                   |     |
| \@lopL and \@lopR in anticipa-       |      | \footsplitskips for use in               |     |
| tion of parallel processing          | 59   | many footnote styles                     | 93  |
| General: Changed version of the      |      | \get@linelistfile: Added                 |     |
| Dekker example                       | 204  | \get@linelistfile                        | 56  |
| Fixed long paragraphs looping        | 1    | \ifledRcol: Added \l@dnumstartsL,        |     |
| Fixed minor typos                    | 1    | \ifl@dpairing and \ifpst@rted            |     |
| Prepared for ledpar package          | 1    | for/from ledpar                          | 44  |
| \fix@page: Added \last@page@num      |      | \initnumbering@reg: Added                |     |
| and \fix@page                        | 58   | \initnumbering@reg                       | 44  |
| \footnoteA: Modified \footnoteA      |      | \l@dcnotetext: Added \l@emptyd@ta        |     |
| and friends to include               |      |                                          | 77  |
| \@thefnmarkA etc                     | 137  | \l@ddofootinsert: Deleted                |     |
| \new@line: Extended \new@line to     |      | \page@start from \l@ddofootinsert        |     |
| output page numbers                  | 64   |                                          | 120 |

|                                                             |                                          |                                    |                                                               |
|-------------------------------------------------------------|------------------------------------------|------------------------------------|---------------------------------------------------------------|
| <code>\l@dgetline@margin:</code>                            | Added                                    | <code>\setprintlines:</code>       | Added <code>\setprintlines</code>                             |
| <code>\l@dgetline@margin</code>                             | ..... 47                                 |                                    | for use by <code>\printlines</code> ..... 95                  |
| <code>\l@dgetlock@disp:</code>                              | Added                                    | <code>\skipnumbering@reg:</code>   | Added                                                         |
| <code>\l@dgetlock@disp</code>                               | ..... 48                                 |                                    | <code>\skipnumbering</code> and supports 66                   |
| <code>\l@dgetsidenote@margin:</code>                        | Added                                    | <code>\sublinenumincrement:</code> | Added                                                         |
| <code>\l@dgetsidenote@margin</code>                         | .. 132                                   |                                    | <code>\firstlinenum</code> , <code>\linenumincrement</code> , |
| <code>\l@drsn@te:</code>                                    | Added <code>\l@dlsn@te</code>            |                                    | <code>\firstsublinenum</code> and                             |
|                                                             | and <code>\l@drsn@te</code> for use in   |                                    | <code>\linenumincrement</code> ..... 48                       |
| <code>\do@line</code>                                       | ..... 77                                 | <code>\sublinenumr@p:</code>       | Using <code>\linenumrep</code>                                |
| <code>\l@dunboxmpfoot:</code>                               | Added                                    |                                    | instead of <code>\linenumr@p</code> ..... 49                  |
| <code>\l@dunboxmpfoot</code>                                | containing                               |                                    | Using <code>\sublinenumrep</code> instead of                  |
|                                                             | some common code ..... 148               |                                    | <code>\sublinenumr@p</code> ..... 49                          |
| <code>\l@dzeropenalties:</code>                             | Added                                    | <code>\vnumfootnoteX:</code>       | Removed                                                       |
| <code>\l@dzeropenalties</code>                              | ..... 75                                 |                                    | extraneous space from                                         |
| <code>\ledlinenum:</code>                                   | Added <code>\ledlinenum</code>           |                                    | <code>\vnumfootnoteX</code> ..... 139                         |
|                                                             | for use by <code>\leftlinenum</code> and | v0.8                               |                                                               |
| <code>\rightlinenum</code>                                  | ..... 50                                 |                                    | General: Bug on endnotes fixed :                              |
| <code>\line@list@stuff:</code>                              | Deleted                                  |                                    | in a <code>//</code> text, all endnotes will                  |
| <code>\page@start</code> from <code>\line@list@stuff</code> | ..... 64                                 |                                    | print and be placed at the ends                               |
| <code>\list@clearing@reg:</code>                            | Added                                    | v0.8.1                             | of columns () ..... 1                                         |
| <code>\list@clearing@reg</code>                             | ..... 55                                 |                                    |                                                               |
| <code>\n@num@reg:</code>                                    | Added <code>\n@num</code> .... 62        |                                    | General: Bug on <code>\edtext</code> ; <code>\critex</code>   |
| <code>\normalbfnoteX:</code>                                | Removed                                  |                                    | ; <code>\lemma</code> fixed : we can now us                   |
|                                                             | extraneous space from                    |                                    | non switching commands ..... 1                                |
| <code>\normalbfnoteX</code>                                 | ..... 139                                | v0.9                               |                                                               |
| <code>\resumenumbering:</code>                              | Changed                                  |                                    | General: No more ledpatch. All                                |
| <code>\resumenumbering</code> for ledpar .                  | 45                                       |                                    | patches are now in the main                                   |
| <code>\setprintendlines:</code>                             | Added                                    | v0.9.1                             | file. .... 1                                                  |
| <code>\setprintendlines</code> for use by                   |                                          |                                    |                                                               |
| <code>\printendlines</code>                                 | ..... 130                                |                                    | General: Fix some bugs linked to in-                          |
|                                                             |                                          |                                    | tegrating ledpatch on the main                                |
|                                                             |                                          |                                    | file. .... 1                                                  |