

**ledmac** (deprecated)  
A presumptuous attempt to port  
EDMAC, TABMAC and EDSTANZA to LaTeX\*

Peter Wilson  
Herries Press<sup>†</sup>

Maïeul Rouquette<sup>‡</sup>

based on the original work by

John Lavagnino, Dominik Wujastyk, Herbert Breger and Wayne Sullivan

**This is documentation of deprecated ledmac package. If you are beginning a new project, we suggest that you use reledmac instead. If for old projects you can't migrate to reledmac, you can continue to use this documentation and the ledmac package. You should add noeledmac option when loading package, to disable message about eledmac.**

**Abstract**

For over ten years EDMAC, a set of PLAIN T<sub>E</sub>X 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<sub>E</sub>X macros, TABMAC, provides for tabular material. Another set of PLAIN T<sub>E</sub>X 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.

You can subscribe to the eledmac email list in:  
<https://lists.berlios.de/pipermail/ledmac-users/>

---

\*This file (ledmac.dtx) has version number ?, last revised ?.

<sup>†</sup>herries dot press at earthlink dot net

<sup>‡</sup>maieul at maieul dot net

## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Overview . . . . .	5
1.2	History . . . . .	7
1.2.1	EDMAC . . . . .	7
1.2.2	ledmac . . . . .	8
<b>2</b>	<b>The ledmac package</b>	<b>8</b>
<b>3</b>	<b>Numbering text lines and paragraphs</b>	<b>9</b>
3.1	Lineation commands . . . . .	11
3.2	Changing the line numbers . . . . .	12
<b>4</b>	<b>The apparatus</b>	<b>14</b>
4.1	Alternate footnote formatting . . . . .	16
4.2	Creating a new series . . . . .	17
<b>5</b>	<b>Fonts</b>	<b>17</b>
<b>6</b>	<b>Verse</b>	<b>20</b>
<b>7</b>	<b>Grouping</b>	<b>22</b>
<b>8</b>	<b>Crop marks</b>	<b>22</b>
<b>9</b>	<b>Endnotes</b>	<b>23</b>
<b>10</b>	<b>Cross referencing</b>	<b>23</b>
<b>11</b>	<b>Side notes</b>	<b>25</b>
<b>12</b>	<b>Familiar footnotes</b>	<b>26</b>
<b>13</b>	<b>Indexing</b>	<b>27</b>
<b>14</b>	<b>Tabular material</b>	<b>27</b>
<b>15</b>	<b>Miscellaneous</b>	<b>31</b>
15.1	Hints . . . . .	31
15.2	Known and suspected limitations . . . . .	36
15.3	Use with other packages . . . . .	37
15.4	Parallel typesetting . . . . .	38
15.5	Notes for EDMAC users . . . . .	40
<b>16</b>	<b>Implementation overview</b>	<b>42</b>

<i>Contents</i>	3
<b>17 Preliminaries</b>	<b>42</b>
17.1 Messages . . . . .	44
<b>18 Sectioning commands</b>	<b>46</b>
<b>19 Line counting</b>	<b>49</b>
19.1 Choosing the system of lineation . . . . .	49
19.2 List macros . . . . .	53
19.3 Line-number counters and lists . . . . .	54
19.4 Reading the line-list file . . . . .	58
19.5 Commands within the line-list file . . . . .	59
19.6 Writing to the line-list file . . . . .	66
<b>20 Marking text for notes</b>	<b>69</b>
20.1 <code>\edtext</code> and <code>\critext</code> themselves . . . . .	70
20.2 Substitute lemma . . . . .	75
20.3 Substitute line numbers . . . . .	75
<b>21 Paragraph decomposition and reassembly</b>	<b>76</b>
21.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code> . . . . .	76
21.2 Processing one line . . . . .	79
21.3 Line and page number computation . . . . .	80
21.4 Line number printing . . . . .	83
21.5 <code>Pstart</code> number printing in side . . . . .	87
21.6 Add insertions to the vertical list . . . . .	88
21.7 Penalties . . . . .	89
21.8 Printing leftover notes . . . . .	90
<b>22 Footnotes</b>	<b>91</b>
22.1 Fonts . . . . .	91
22.2 Outer-level footnote commands . . . . .	92
22.3 Normal footnote formatting . . . . .	94
22.4 Standard footnote definitions . . . . .	99
22.5 Paragraphed footnotes . . . . .	101
22.5.1 Insertion of footnotes separator . . . . .	107
22.6 Columnar footnotes . . . . .	107
<b>23 Output routine</b>	<b>111</b>
<b>24 Cross referencing</b>	<b>118</b>
<b>25 Endnotes</b>	<b>122</b>
<b>26 Side notes</b>	<b>125</b>

<b>27 Familiar footnotes</b>	<b>129</b>
27.1 The A series footnotes . . . . .	131
27.2 Footnote formats . . . . .	131
27.2.1 Two column footnotes . . . . .	134
27.2.2 Three column footnotes . . . . .	135
27.2.3 Paragraphed footnotes . . . . .	137
27.3 Other series footnotes . . . . .	139
<b>28 Minipages and such</b>	<b>140</b>
<b>29 Indexing</b>	<b>143</b>
<b>30 Macro as environment</b>	<b>146</b>
<b>31 Verse</b>	<b>149</b>
<b>32 Arrays and tables</b>	<b>152</b>
<b>33 The End</b>	<b>172</b>
<b>A Examples</b>	<b>173</b>
A.1 Simple example . . . . .	181
A.2 General example of features . . . . .	182
A.3 Gascoigne . . . . .	185
A.4 Shakespeare . . . . .	188
A.5 Classical text edition . . . . .	191
A.6 Nijmegen . . . . .	196
A.7 Irish verse . . . . .	201
<b>References</b>	<b>206</b>
<b>Index</b>	<b>206</b>
<b>Change History</b>	<b>223</b>

## List of Figures

1	Output from <code>ledeasy.tex</code> . . . . .	174
2	Output from <code>ledfeat.tex</code> . . . . .	175
3	Output from <code>ledioc.tex</code> . . . . .	176
4	Output from <code>ledarden.tex</code> . . . . .	177
5	Output from <code>ledmixed.tex</code> . . . . .	178
6	Output from <code>ledekker.tex</code> . . . . .	179
7	Output from <code>ledbraonain.tex</code> . . . . .	180

# 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](mailto: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 A); 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 A, 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.<sup>1</sup> A description by John and Dominik of this version of EDMAC was published as ‘An overview of EDMAC: a PLAIN T<sub>E</sub>X 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<sub>E</sub>X 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*,<sup>2</sup> an edition of the letters of Nicolaus Copernicus,<sup>3</sup> Simon Bredon’s *Arithmetica*,<sup>4</sup> a Latin translation by Plato of Tivoli of an

<sup>1</sup>This 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).

<sup>2</sup>Gerhard 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).

<sup>3</sup>Being prepared at the German Copernicus Research Institute, Munich.

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

Arabic astrolabe text,<sup>5</sup> a Latin translation of part II of the Arabic *Algebra* by Abū Kāmil Shujā' b. Aslam,<sup>6</sup> the Latin *Rithmachia* of Werinher von Tegernsee,<sup>7</sup> a middle-Dutch romance epic on the Crusades,<sup>8</sup> a seventeenth-century Hungarian politico-philosophical tract,<sup>9</sup> an anonymous Latin compilation from Hungary entitled *Sermones Compilati in Studio Generali Quinqueecclesiensi in Regno Ungarie*,<sup>10</sup> the collected letters and papers of Leibniz,<sup>11</sup> Theodosius's *Spherics*, the German *Algorismus* of Sacrobosco, the Sanskrit text of the *Kāśikāvṛtti* of Vāmana and Jayāditya,<sup>12</sup> and the English texts of Thomas Middleton's collected works, as well as the editions illustrated in Appendix A.

### 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.

## 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 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.

<sup>5</sup>Richard Lorch, Gerhard Brey *et al.*, at the same Institute.

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

<sup>7</sup>Menso Folkerts, 'Die *Rithmachia* des Werinher von Tegernsee', *ibid.*

<sup>8</sup>Geert H. M. Claassens, *De Middelnederlandse Kruisvaartromans*, (Amsterdam: Schiphower en Brinkman, 1993).

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

<sup>10</sup>Being produced, as was the previous book, by Gyula Mayer in Budapest.

<sup>11</sup>Leibniz, *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>)

<sup>12</sup>Being prepared at Poona and Lausanne Universities.



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  
`\pend` 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
\pstart
This paragraph too has its
lines automatically numbered.
\pend
The lines of this paragraph are
not numbered.
\pstart
And here the numbering begins
again.
\pend
\endnumbering

```

1 This is a sample paragraph  
2 with lines numbered  
3 automatically.  
4 This paragraph too  
5 has its lines automatically  
6 numbered.  
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.
Another paragraph of numbered
text.
\endnumbering
\endgroup

```

1 A paragraph of numbered  
2 text.  
3 Another paragraph of  
4 numbered text.

`\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.<sup>13</sup>

`\firstlinenum` By default, `ledmac` numbers every 5th line. There are two counters, `firstlinenum`  
`\linenumincrement` 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:

```

\firstsublinenum
\sublinenumincrement

```

There are similar commands, `\firstsublinenum{<num>}` and `\sublinenumincrement{<num>}`

<sup>13</sup>For 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.

```

\pausenumbering
\resumenumbering
    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 \pausenumbering and \resumenumbering which
    are just like \endnumbering ... \beginnumbering, except that they arrange for
    your line numbering to continue across the break. Use \pausenumbering only
    between numbered paragraphs:
    \beginnumbering
    \pstart
    Paragraph of text.
    \pend
    \pausenumbering
    \resumenumbering
    \pstart
    Another paragraph.
    \pend
    \endnumbering

```

```

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`.

```

\numberpstarttrue
\numberpstartfalse
  \thepstart

```

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

### 3.1 Lineation commands

```

\numberlinefalse
\numberlinetrue
  \lineation

```

Line numbering can be disabled with `\numberlinefalse`. It can be enabled again with `\numberlinetrue`. Lines can be numbered either by page, by `pstart` or by section; you specify this using the `\lineation{<arg>}` macro, where `<arg>` is either `page`, `pstart` or `section`. You may only use this command at places where numbering is not in effect; you can't change the lineation system within a section. You can change it between sections: they don't all have to use the same lineation

system. The package's standard setting is `\lineation{section}`. If the lineation is by `pstart`, the `pstart` number will be printed before the line number in the notes.

`\linenummargin` The command `\linenummargin{location}` specifies the margin where the line numbers will be printed. The permissible value for `location` is one out of the list `left`, `right`, `inner`, or `outer`, for example `\linenummargin{inner}`. The package's default setting is `\linenummargin{left}` to typeset the numbers in the left hand margin. You can change this whenever you're not in the middle of making a paragraph.

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

`\firstlinenum` 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 `\firstlinenum{num}`, etc. `\firstlinenum` specifies the number of the first line in a section to number, and `\linenumincrement` is the increment between numbered lines. `\firstsublinenum` and `\sublinenumincrement` do the same for sub-lines. Initially, all these are set to 5 (e.g., `\firstlinenum{5}`).

`\linenumberlist` You can define `\linenumberlist` to specify a non-uniform distribution of printed line numbers. For example:  
`\def\linenumberlist{1,2,3,5,7,11,13,17,19,23,29}`  
to have numbers printed on prime-numbered lines only. There must be no spaces within the definition which consists of comma-separated 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 `\linenumberlist` or following the vacuous definition `\def\linenumberlist{}` the standard numbering sequence is applied. The standard sequence is that specified by the combination of the `firstlinenum`, `linenumincrement`, `firstsublinenum` and `linenumincrement` counter values.

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

### 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.

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

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

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

`Alph` Uppercase letters (A...Z).

`alph` Lowercase letters (a...z).

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

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

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

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

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

`\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 \edtext{Smith}{          1 I saw my friend
\Afootnote{Jones C, D.}}                2 Smith on Tuesday.
on Tuesday.                             2 Smith] Jones C, D.
```

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

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

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

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

this will be adequate for the most demanding editions. But it is not hard to add further layers of notes should they be required.

`\Aendnote`    The package also maintains five separate series of endnotes. Like footnotes  
`\Bendnote`    each macro takes a single argument like `\Aendnote{<text>}`. Normally, none of  
`\Cendnote`    them is printed: you must use the `\doendnotes` macro described below (p. 23) to  
`\Dendnote`    call for their output at the appropriate point in your document.

`\Eendnote`    Sometimes you want to change the lemma that gets passed to the notes.  
`\lemma`    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:

```

\edtext{I saw my friend
  \edtext{Smith}{\Afootnote{Jones
    C, D.}} on Tuesday.}
{\lemma{I \dots\ Tuesday.}
  \Bfootnote{The date was
    July 16, 1954.}
}

```

1	I saw my friend
2	Smith on Tuesday.
<u>2</u>	<u>Smith</u> ] Jones C, D.
<u>1-2</u>	<u>I ... Tuesday.]</u>
	The date was July 16, 1954.

`\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. 23) 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.176 and 178);
- `\foottwocol` formats them as separate paragraphs, but in two columns (see bottom notes in fig. 4, p.177);
- `\footthreecol`, in three columns (see second layer of notes in fig.2, p.175).

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.105):

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



You should set up the page layout parameters, and in particular the `\baselineskip` of the footnotes (this is done for you if you use the standard `\notefontsetup`), 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.<sup>14</sup>

## 4.2 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}{\@tag}{#1}}\to\inserts@list
    \global\advance\insert@count by \@ne
  \else
    \vGfootnote{G}{\{0|0|0|0|0|0|0\}}{#1}%
  \fi\ignorespaces}
\newinsert\Gfootins

\newcommand*\mpGfootnote}[1]{%
  \ifnumberedpar@
    \xright@appenditem{\noexpand\mpvGfootnote{G}%
                      {\l@d@nums}{\@tag}{#1}}\to\inserts@list
    \global\advance\insert@count by \@ne
  \else
    \mpvGfootnote{G}{\{0|0|0|0|0|0|0\}}{#1}%
  \fi\ignorespaces}
\newinsert\mpGfootins

\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.

---

<sup>14</sup>There 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 103 explains why this restriction is necessary.

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<sub>E</sub>X 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 `\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. 95)—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 Sullivan<sup>15</sup> 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.

<code>\stanza</code>	Use <code>\stanza</code> at the start of a stanza. Each line in a stanza is ended by an ampersand (&), and the stanza itself is ended by putting <code>\&amp;</code> at the end of the last line.
<code>\stanzaindentbase</code>	Lines within a stanza may be indented. The indents are integer multiples of the length <code>\stanzaindentbase</code> , whose default value is 20pt.
<code>\setstanzaindents</code>	In order to use the stanza macros, one must set the indentation values. First the value of <code>\stanzaindentbase</code> 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 <code>\setstanzaindents{3,1,2,1,2}</code> .

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<sub>E</sub>X 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, 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 `stanzaindentsrepetition` counter at  $n$ . For example :

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

is like

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

---

<sup>15</sup>Department of Mathematics, University College, Dublin 4, Ireland

If you don't use the `stanzaindentsrepetition` counter, make sure you have at least one more numerical entry in `\setstanzavalues` than the number of lines in the stanza. The macros make no restriction on the number of lines in a stanza. Stanza indentation values (and penalty values) obey T<sub>E</sub>X's grouping conventions, so if one stanza among several has a different structure, its indentations (penalties) may be set within a group; the prior values will be restored when the group ends.

`\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<sub>E</sub>X, which is the default. Values between 0 and 10000 are penalty values; values between 10001 and 20000 have 10000 subtracted and the result is given as a negative penalty. The mechanism used for indentations and penalties requires unsigned values less than 32768. No penalty is placed after the last line, so the final ,0 in then example above could be omitted. 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...

```

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

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

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

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

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

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

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

## 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@lemmafnt`, and `\notefontsetup` to select fonts, just as the footnote macros do (see p. 18 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.<sup>16</sup>

`\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`  
`\sublineref` These commands will produce, respectively, the page, line and sub-line on which the `\edlabel{⟨lab⟩}` command occurred.

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

---

<sup>16</sup>More precisely, you should stick to characters in the T<sub>E</sub>X categories of 'letter' and 'other'.

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`      text, and operate in the familiar fashion. As an example, here is one way of numbering  
`\pageref`    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<sup>3,4</sup> 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 package also provides three series of additional footnotes called `\footnoteA` through `\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., B). `\footnormalX` is the typical footnote format. With `\footparagraphX` the series is typeset a one paragraph, with `\foottwocolX` the notes are in two columns, and `\footthreecolX` 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.

There are six environments; the `edarray*` environments are for math and

```

edarrayl
edarrayc
edarrayr
edtabularl
edtabularc
edtabularr

```

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

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

1	2	3
a	bb	ccc
AAA	BB	C

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

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

For example:

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

produces the following parallel pair of verses.

1	wish I was a little bug	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.

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

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

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

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

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

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

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

You can also define your own 'fill'. For example:

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

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

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

1	2	3	4
a			d
A	B	C	D

`\edatleft` `\edatleft[<math>]{<symbol>}{<halfheight>}` typesets the math `<symbol>` as `\edatright` `\left<symbol>` with the optional `<math>` centered before it. The `<symbol>` is twice

$\langle halfheight \rangle$  tall. The `\edatright` macro is similar and it typesets `\right<symbol>` with  $\langle math \rangle$  centered after it.

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

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

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

For example:

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

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

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

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

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

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

## 15 Miscellaneous

<code>\extensionchars</code>	When the package assembles the name of the auxiliary file for a section, it prefixes <code>\extensionchars</code> to the section number. This is initially defined to be empty, but you can add some characters to help distinguish these files if you like; what you use is likely to be system-dependent. If, for example, you said <code>\renewcommand{\extensionchars}{!}</code> , then you would get temporary files called <code>jobname.!1</code> , <code>jobname.!2</code> , etc.
<code>\ifledfinal</code>	The package can take options. The option 'final', which is the default is for final typesetting; this sets <code>\ifledfinal</code> to TRUE. The other option, 'draft', may be useful during earlier stages and sets <code>\ifledfinal</code> to FALSE.
<code>\showlemma</code>	The lemma within the text is printed via <code>\showlemma{lemma}</code> . Normally, or with the 'final' option, the definition of <code>\showlemma</code> is: <code>\newcommand*{\showlemma}[1]{#1}</code> so it just produces its argument. With the 'draft' option it is defined as <code>\newcommand*{\showlemma}[1]{\textit{#1}}</code> 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: <pre> \ifledfinal\else   \renewcommand{\showlemma}[1]{\textbf{#1}}% or simply ...[1]{#1} \fi </pre>
<code>\ledplinenumtrue</code> <code>\ledplinenumfalse</code> <code>\symplinenum</code>	Following the declaration <code>\ledplinenumtrue</code> critical footnotes will be marked with their line number. After <code>\ledplinenumfalse</code> the footnotes will be marked by <code>\symplinenum</code> , whose default definition is <code>\newcommand*{\symplinenum}{}</code>

### 15.1 Hints

By doing a little work it is possible, for example, to set things up so that a particular footnote series only prints the linenumber for the first footnote on a line.<sup>17</sup> You may wish to skip the following but if not read it in conjunction with the code definitions from section 22.3. Suppose that we only want this to apply to the B series of normal footnotes. To accomplish this goal we have to modify the definition of `\normalvfootnote` as follows:

```
\makeatletter
```

---

<sup>17</sup>This was requested by Dirk-Jan Dekker ([djdekker@let.ru.nl](mailto:djdekker@let.ru.nl)).

```

\newcommand*\previous@B@number{-1}
\newcommand*\previous@page{-1}
\renewcommand*\normalvfootnote}[2]{
  \insert\csname #1footins\endcsname\bgroup
  \notefontsetup
  \footsplitskips
  \spaceskip=\z@skip \xspaceskip=\z@skip
  \l@dparsefootspec #2\ledplinenumtrue%           % NEW FROM HERE
  \ifnum\@nameuse{previous@#1@number} = \l@dparsedstartline\relax
    \ledplinenumfalse
  \fi
  \ifnum\previous@page=\l@dparsedstartpage\relax
  \else \ledplinenumtrue \fi
  \ifnum\l@dparsedstartline=\l@dparsedendline\relax
  \else \ledplinenumtrue \fi
  \expandafter\xdef\csname previous@#1@number\endcsname{\l@dparsedstartline}%
  \xdef\previous@page{\l@dparsedstartpage}%           % TO HERE
  \csname #1footfmt\endcsname #2\egroup}
\footnormal{B}
\makeatother

```

The additional code uses `\l@dparsefootspec` to get the footnote's line number as `\l@dparsedstartline` and the page number as `\l@dparsedstartpage`. It then sets `\ledplinenum` according to whether or not `\l@dparsedstartline` is the same as the previous (`\previous@B@number`) number. If the page number has changed then the line number must be printed. If the starting line number is not the same as the ending line number then the line number must be printed. After `\ledplinenum` has been set the two previous values are updated to the current line and page numbers.

After the redefinition of `\normalvfootnote` the B series has to be respecified as normal for the changes to take effect. The A series will still be in the traditional style of printing every line number. To eliminate duplicate printing from the normal A series, you simply need to define `\previous@A@number` and respecify the series.

Similar techniques can be used for the other footnote styles.

Dirk-Jan Dekker felt that there was too much empty space if the starting line number was omitted in a footnote. He proposed<sup>18</sup> this solution, here applied to a paragraphed footnote.

```

\renewcommand*\Bparafootfmt}[3]{%
  \ledsetnormalparstuff
  \scriptsize
  \notenumfont\printlines#1%           % NEW FROM HERE
  \ifledplinenum
    \enspace
  \else
    {\hskip 0em plus 0em minus .4em}%
  \fi%           % TO HERE

```

---

<sup>18</sup>Posted to `comp.text.tex` on 24 January 2004.



```
{\select@lemmafонт#1|#2}\rbracket\enskip
#3\penalty-10}
```

Another question has been how to control the printing, or not, of line numbers in the footnote from the `\edtext` command. Here is an awful hack to do this. The example is an extension of the code just above.

```
\newcounter{killnum}
  \setcounter{killnum}{0}
\newcommand*\killnumbers{\setcounter{killnum}{-1}}
\newcommand*\restorenumbers{\setcounter{killnum}{0}}
\renewcommand*\Bparafootfmt[3]{%
  \ledsetnormalparstuff
  \scriptsize
  \ifnum\c@killnum<\z@\ledplinenumfalse\fi%% NEW
  \notenumfont\printlines#1|%
  \ifledplinenum
    \enspace
  \else
    {\hskip 0em plus 0em minus .4em}%
  \fi%
  {\select@lemmafонт#1|#2}\rbracket\enskip
#3\penalty-10}
```

In the text it is used like:

```
...
\edtext{text}{\Bfootnote{TEXT\killnumbers}}% later B line numbers not printed
...
\edtext{textual}{\Bfootnote{TEXTUAL\restorenumbers}}% later B numbers printed
...
```

That is, `\killnumbers` and `\restorenumbers` only take effect for the next and later `\edtexts`, not the one they are in. You have to kill/restore numbers in the note *before* you want the change.

Dirk-Jan Dekker suggested<sup>19</sup> the following `\killnumber` macro if you want to occasionally kill a number.

```
\newcommand*\killnumber{\linenum{-1||-1||}}
```

Then insert

```
\ifnum#2=-1 \ledplinenumfalse\fi
```

near the start of the definition of `\printlines` so it reads

```
\def\printlines#1|#2|#3|#4|#5|#6|#7|{\begingroup
  \ifnum#2=-1 \ledplinenumfalse\fi%% NEW
  \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
  ...
```

---

<sup>19</sup>Private communication, 17 February 2004.

It is used like this:

```
\edtext{critical}{\killnumber\Afootnote{criticism}}
```

The `\killnumber` command will kill the line number for the one note, unlike `\killnumbers` which kills numbers for subsequent notes.

Perhaps, though, you just want a footnote series with no numbers at all (and maybe no lemma either).

```
\footparagraph{A}
\makeatletter
\def\zparafootfmt#1#2#3{%
  \ledsetnormalparstuff
  \notetextfont #3\penalty-10 }
\makeatother
\let\Afootfmt=\zparafootfmt
...
\beginnumbering
\edtext{}{\Afootnote{numberless and lemmaless}}
...
```

At least 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\prefootskip % the big initial skip
\prefootskip=3.3em plus .6em minus .6em
\newif\ifskipped \skippedfalse
\renewcommand*{\normalfootstart}[1]{%
  \ifskipped
    \vskip\skip\csname #1footins\endcsname% normal skip
  \else
    \skip\prefootskip%      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.

Another user has had a wider ranging set of requirements:

- Number paragraphs and use the number in the notes for that paragraph;

- Duplicate a paragraph number later in the document and use it for that paragraph's notes;
- In any series of notes only use the paragraph number for the first in the paragraph
- Have some series use line numbers in the notes and in other series have neither lemmas nor line numbers in the notes.
- Perhaps eliminate all paragraph numbers in the notes.

Here is some code that enables these requirements to be met. This should be in an environment where @ is treated as a letter. First, here is a version of `\ref` that returns a number even if the corresponding `\label` has not been defined.

```
\newcommand*\saferef}[1]{%
  \expandafter\ifx\csname r@#1\endcsname\relax 0\else
  \ref{#1}\fi}
```

Now for some code for the paragraph numbering. Use `\newpara` at the start of a numbered paragraph and `\oldpara{<lab>}` at the start of a 're-numbered' one, where `\label{<lab>}` has been used in the original numbered one.

```
\newcounter{para}\setcounter{para}{0}
\newcounter{thispara}\setcounter{thispara}{0}
\newcommand*\newpara{%
  \refstepcounter{para}%
  \setcounter{thispara}{\value{para}}%
  \noindent\textbf{\thepara. }}
\newcommand*\oldpara[1]{%
  \noindent\setcounter{thispara}{\saferef{#1}}\textbf{\saferef{#1}. }}
```

Set up the A note series for lemmas, line numbers and non-repeated paragraph numbers, assuming paragraphed notes.

```
\newif\ifparnumfoot
  \parnumfoottrue% false to eliminate paragraph numbers in notes
\newcommand*\previous@Aparnum}{-1}
\def\printlinesA#1|#2|#3|#4|#5|#6|#7|{\begingroup
  \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
  \ifnum\previous@Aparnum=\the\c@thispara% not a new paragraph
  \else% new paragraph, print, and update the check
    \ifparnumfoot \textbf{\thethispara.}\fi
    \xdef\previous@Aparnum{\the\c@thispara}%
  \fi
  \ifledplinenum \linenumr@p{#2}\else \symplinenum\fi
  \ifl@d@ssub \fullstop \sublinenumr@p{#3}\fi
  \ifl@d@dash \endashchar\fi
  \ifl@d@pnum #4\fullstop\fi
  \ifl@d@elin \linenumr@p{#5}\fi}
```

```

\ifl@d@esl \ifl@d@elin \fullstop\fi \sublinenumr@p{#6}\fi
\endgroup}
\renewcommand*{\Afootfmt}[3]{%
\ledsetnormalparstuff
{\notenumfont\printlinesA#1|}\enspace
{\select@lemmafонт#1|#2}\rbracket\enskip
#3\penalty-10 }

```

Set up the B series notes for no line numbers or lemmas, just non-repeated paragraph numbers, assuming normal notes.

```

\newcommand*{\previous@Bparnum}{-1}
\def\printlinesB#1|#2|#3|#4|#5|#6|#7|{\begingroup
\setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
\ifnum\previous@Bparnumm=\the\c@thispara% not a new paragraph
\else% new paragraph, print, and update the check
\ifparnumfoot \textbf{\thethispara.}\fi
\xdef\previous@Aparnum{\the\c@thispara}%
\fi
\endgroup}
\renewcommand*{\Bfootfmt}[3]{%
\ledsetnormalparstuff
{\notenumfont\printlinesB#1|}%\enspace
{\select@lemmafонт#1|#2}%\enskip
#3\strut\par}

```

You can use the above like:

```

...
\newpara\label{fpara} A numbered\edtext{\Bfootnote{lemma-less
and linenummer-less}} \edtext{paragraph}{\Afootnote{chunk}} ...
...
\oldpara{fpara} \edtext{Repeated}{\Afootnote{Again}}
paragraph\edtext{\Bfootnote{Just a comment}} ...
...

```

## 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 T<sub>E</sub>X 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 T<sub>E</sub>X, 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 14, p. 17, and described in more detail on p. 103, 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 T<sub>E</sub>X 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 this<sup>20</sup> 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

---

<sup>20</sup>Reported by Dirk-Jan Dekker in the CTT thread ‘Incompatibility of “color” package’ on 2003/08/28.

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
  \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}{\cleartooddpage}
\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}` and `\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 cooperates 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 changed<sup>21</sup> to `\critext`; use the `\edtext` macro instead. However, if you do use `\critext` (the new name for `\text`), the following is a reminder.

`\critext` Within numbered paragraphs, footnotes and endnotes are generated by forms of the `\critext` macro:

```
\critext{⟨lemma⟩}⟨commands⟩/
```

The `⟨lemma⟩` argument is the lemma in the main text: `\critext` both prints this as part of the text, and makes it available to the `⟨commands⟩` you specify to generate notes. The `/` at the end terminates the command; it is part of the macro's definition so that spaces after the macro will be treated as significant.

For example:

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

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

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

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

However, `\critext` cannot handle overlapping but unnnested 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

<sup>21</sup>A 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.



`\lemma` and `\linenum` commands may be used to generate overlapping notes if necessary.)

The second argument of the `\critext` macro,  $\langle commands \rangle$ , 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
```

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  $\text{\TeX}$ . 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  $\text{\TeX}$  (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}[2015/07/19 v0.19.2a 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 \newif\ifnoeledmac
7 \DeclareOption{final}{\ledfinaltrue}
8 \DeclareOption{draft}{\ledfinalfalse}
9 \DeclareOption{noeledmac}{\noeledmactrue}
10 \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.

```

11 \ProcessOptions*\relax
12
13 % \end{macrocode
14 % \end{macro}
15 %
16 % \begin{macro}{\showlemma}
17 % \verb?\showlemma?\marg{lemma} typesets the lemma text in the body.
18 % It depends on the option.
19 % \changes{v0.4}{2004/02/29}{Added \cs{showlemma}}
20 % \begin{macrocode}
21 \ifledfinal
22 \newcommand*{\showlemma}[1]{#1}
23 \else
24 \newcommand*{\showlemma}[1]{\textit{#1}}
25 \fi
26

```

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

```

Initialize it as \empty
27 \let\linenumberlist=\empty
28

```

`\@l@tempcnta` In imitation of L<sup>A</sup>T<sub>E</sub>X, 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).

```

29 \newcount\@l@tempcnta \newcount\@l@tempcntb

```

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

```

30 \newif\ifl@dmemoir
31 \ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
32

```

## 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.
33 \newcommand{\ledmac@warning}[1]{\PackageWarning{ledmac}{#1}}

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

35 \ifnoeledmac
36 \else
37 \ledmac@error{Using package 'ledmac' is deprecated. We suggest\MessageBreak using 'reledmac'}
38 \fi

\led@err@NumberingStarted
\led@err@NumberingNotStarted 39 \newcommand*{\led@err@NumberingStarted}{%
\led@err@NumberingShouldHaveStarted 40 \ledmac@error{Numbering has already been started}{\@ehc}}
41 \newcommand*{\led@err@NumberingNotStarted}{%
42 \ledmac@error{Numbering was not started}{\@ehc}}
43 \newcommand*{\led@err@NumberingShouldHaveStarted}{%
44 \ledmac@error{Numbering should already have been started}{\@ehc}}

\led@mess@NotesChanged
45 \newcommand*{\led@mess@NotesChanged}{%
46 \typeout{ledmac reminder: }%
47 \typeout{ The number of footnotes in this section
48 has changed since the last run.}%
49 \typeout{ You will need to run LaTeX two more times
50 before the footnote placement}%
51 \typeout{ and line numbering in this section are
52 correct.}}

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

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

\led@warn@BadLineation
\led@warn@BadLinenummargin 58 \newcommand*{\led@warn@BadLineation}{%
\led@warn@BadLockdisp 59 \ledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadSublockdisp 60 \newcommand*{\led@warn@BadLinenummargin}{%
61 \ledmac@warning{Bad \string\linenummargin\space argument}}
62 \newcommand*{\led@warn@BadLockdisp}{%

```

```

63 \ledmac@warning{Bad \string\lockdisp\space argument}}
64 \newcommand*{\led@warn@BadSublockdisp}{%
65 \ledmac@warning{Bad \string\sublockdisp\space argument}}

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

arn@BadAdvancelineSubline
d@warn@BadAdvancelineLine 68 \newcommand*{\led@warn@BadAdvancelineSubline}{%
69 \ledmac@warning{\string\advanceline\space produced a sub-line
70 number less than zero.}}
71 \newcommand*{\led@warn@BadAdvancelineLine}{%
72 \ledmac@warning{\string\advanceline\space produced a line
73 number less than zero.}}

\led@warn@BadSetline
\led@warn@BadSetlinenum 74 \newcommand*{\led@warn@BadSetline}{%
75 \ledmac@warning{Bad \string\setline\space argument}}
76 \newcommand*{\led@warn@BadSetlinenum}{%
77 \ledmac@warning{Bad \string\setlinenum\space argument}}

led@err@PstartNotNumbered
\led@err@PstartInPstart 78 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PendNotNumbered 79 \ledmac@error{\string\pstart\space must be used within a
\led@err@PendNoPstart 80 numbered section}{\@ehc}}
ed@err@AutoparNotNumbered 81 \newcommand*{\led@err@PstartInPstart}{%
82 \ledmac@error{\string\pstart\space encountered while another
83 \string\pstart\space was in effect}{\@ehc}}
84 \newcommand*{\led@err@PendNotNumbered}{%
85 \ledmac@error{\string\pend\space must be used within a
86 numbered section}{\@ehc}}
87 \newcommand*{\led@err@PendNoPstart}{%
88 \ledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
89 \newcommand*{\led@err@AutoparNotNumbered}{%
90 \ledmac@error{\string\autopar\space must be used within a
91 numbered section}{\@ehc}}

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

\led@warn@DuplicateLabel
\led@warn@RefUndefined 94 \newcommand*{\led@warn@DuplicateLabel}[1]{%
95 \ledmac@warning{Duplicate definition of label '#1' on page \the\pageno.}}
96 \newcommand*{\led@warn@RefUndefined}[1]{%
97 \ledmac@warning{Reference '#1' on page \the\pageno\space undefined.
98 Using '000'.}}

```

```

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

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

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

\led@err@TooManyColumns
\led@err@UnequalColumns 105 \newcommand*{\led@err@TooManyColumns}{%
\led@err@LowStartColumn 106 \ledmac@error{Too many columns}{\@ehc}}
\led@err@HighEndColumn 107 \newcommand*{\led@err@UnequalColumns}{%
\led@err@ReverseColumns 108 \ledmac@error{Number of columns is not equal to the number
109 in the previous row (or \protect\ \space forgotten?)}{\@ehc}}
110 \newcommand*{\led@err@LowStartColumn}{%
111 \ledmac@error{Start column is too low}{\@ehc}}
112 \newcommand*{\led@err@HighEndColumn}{%
113 \ledmac@error{End column is too high}{\@ehc}}
114 \newcommand*{\led@err@ReverseColumns}{%
115 \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 `\resumnumbering` 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.

```

116 \newcount\section@num
117 \section@num=0
118 \let\extensionchars=\empty

```

```

\ifnumbering The \ifnumbering flag is set to true if we're within a numbered section (that is,
\numberingtrue between \beginnumbering and \endnumbering). You can use \ifnumbering in
\numberingfalse your own code to check whether you're in a numbered section, but don't change
the flag's value.
119 \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 120 \newif\ifl@dpairing
\l@dpairingfalse 121 \l@dpairingfalse
\ifpst@rtedL 122 \newif\ifpst@rtedL
\pst@rtedLtrue 123 \pst@rtedLfalse
\pst@rtedLfalse 124 \newcount\l@dnumpstartsL
\l@dnumpstartsL 125 \newif\ifledRcol
\ifledRcol The \ifnumberingR flag is set to true if we're within a right text numbered
section.
126 \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.

127 \newcommand*\beginnumbering}{%
128 \ifnumbering
129 \led@err@NumberingStarted
130 \endnumbering
131 \fi
132 \global\numberingtrue
133 \global\advance\section@num \@ne
134 \initnumbering@reg
135 \message{Section \the\section@num }%
136 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
137 \l@ddend@stuff
138 \setcounter{pstart}{1}
139 }
140 \newcommand*\initnumbering@reg}{%
141 \global\pst@rtedLfalse
142 \global\l@dnumpstartsL \z@
143 \global\absline@num \z@
144 \global\line@num \z@

```

```

145 \global\subline@num \z@
146 \global\@lock \z@
147 \global\sub@lock \z@
148 \global\sublines@false
149 \global\let\next@page@num=\relax
150 \global\let\sub@change=\relax}
151

```

`\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.

```

152 \def\endnumbering{%
153   \ifnumbering
154     \global\numberingfalse
155     \normal@pars
156     \ifl@dpairing
157       \global\pst@rtedLfalse
158     \else
159       \ifx\insertlines@list\empty\else
160         \global\noteschanged@true
161       \fi
162       \ifx\line@list\empty\else
163         \global\noteschanged@true
164       \fi
165     \fi
166     \ifnoteschanged@
167       \led@mess@NotesChanged
168     \fi
169   \else
170     \led@err@NumberingNotStarted
171   \fi
172 \autoparfalse}

```

`\pausenumbering` The `\pausenumbering` macro is just the same as `\endnumbering`, but with the `\resumenumbering` `\ifnumbering` flag set to true, to show that numbering continues across the gap.<sup>22</sup>

```

173 \newcommand{\pausenumbering}{%
174   \endnumbering\global\numberingtrue}

```

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

```

175 \newcommand*{\resumenumbering}{%
176   \ifnumbering
177     \global\pst@rtedLtrue
178     \global\advance\section@num \@ne
179     \led@mess@SectionContinued{\the\section@num}%
180     \line@list@stuff{\jobname.\extensionchars\the\section@num}%

```

<sup>22</sup>Our thanks to Wayne Sullivan, who suggested the idea behind these macros.



```

181     \l@dend@stuff
182   \else
183     \led@err@NumberingShouldHaveStarted
184     \endnumbering
185     \beginnumbering
186   \fi}
187

```

## 19 Line counting

### 19.1 Choosing the system of lineation

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

```

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

```

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

`ledmac` will use the line-of-section system unless instructed otherwise.

```

188 \newif\ifbypage@
189 \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.

```

```

190 \newcommand*{\lineation}[1]{%
191   \ifnumbering
192     \led@err@LineationInNumbered
193   \else
194     \def\@tempa{#1}\def\@tempb{page}%
195     \ifx\@tempa\@tempb
196       \global\bypage@true
197       \global\bypstart@false
198     \else
199       \def\@tempb{pstart}%
200       \ifx\@tempa\@tempb
201         \global\bypage@false
202         \global\bypstart@true
203     \else
204       \def\@tempb{section}
205       \ifx\@tempa\@tempb
206         \global\bypage@false

```

```

207         \global\bystart@false
208     \else
209         \led@warn@BadLineation
210     \fi
211 \fi
212 \fi
213 \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.

```

214 \newcount\line@margin
215 \newcommand*{\linenummargin}[1]{%
216   \l@dgetline@margin{#1}%
217   \ifnum\@l@dttempcntb>\m@ne
218     \global\line@margin=\@l@dttempcntb
219   \fi}}
220 \newcommand*{\l@dgetline@margin}[1]{%
221   \def\@tempa{#1}\def\@tempb{left}%
222   \ifx\@tempa\@tempb
223     \@l@dttempcntb \z@
224   \else
225     \def\@tempb{right}%
226     \ifx\@tempa\@tempb
227       \@l@dttempcntb \@ne
228     \else
229       \def\@tempb{outer}%
230       \ifx\@tempa\@tempb
231         \@l@dttempcntb \tw@
232       \else
233         \def\@tempb{inner}%
234         \ifx\@tempa\@tempb
235           \@l@dttempcntb \thr@@
236         \else
237           \led@warn@BadLinenummargin
238           \@l@dttempcntb \m@ne
239         \fi
240       \fi
241     \fi
242   \fi}
243

```

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

```
244 \newcounter{firstlinenum}
245 \setcounter{firstlinenum}{5}
246 \newcounter{linenumincrement}
247 \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.

```
248 \newcounter{firstsublinenum}
249 \setcounter{firstsublinenum}{5}
250 \newcounter{sublinenumincrement}
251 \setcounter{sublinenumincrement}{5}
252
```

`\firstlinenum`    These macros can be used to set the corresponding counters.  
`\linenumincrement`    253 `\newcommand*\firstlinenum}[1]{\setcounter{firstlinenum}{#1}}`  
`\firstsublinenum`    254 `\newcommand*\linenumincrement}[1]{\setcounter{linenumincrement}{#1}}`  
`\sublinenumincrement`    255 `\newcommand*\firstsublinenum}[1]{\setcounter{firstsublinenum}{#1}}`  
                           256 `\newcommand*\sublinenumincrement}[1]{\setcounter{sublinenumincrement}{#1}}`  
                           257

`\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.

```
258 \newcount\lock@disp
259 \newcommand{\lockdisp}[1]{%
260 \l@dgetlock@disp{#1}%
261 \ifnum\@l@tempcntb>\m@ne
262 \global\lock@disp=\@l@tempcntb
263 \else
264 \led@warn@BadLockdisp
265 \fi}
266 \newcommand*\l@dgetlock@disp}[1]{
267 \def\@tempa{#1}\def\@tempb{first}%
268 \ifx\@tempa\@tempb
269 \@l@tempcntb \z@
270 \else
271 \def\@tempb{last}%
272 \ifx\@tempa\@tempb
273 \@l@tempcntb \@ne
274 \else
275 \def\@tempb{all}%
```

```

276     \ifx\@tempa\@tempb
277         \@l@dttempcntb \tw@
278     \else
279         \@l@dttempcntb \m@ne
280     \fi
281 \fi
282 \fi}
283

```

`\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.

```

284 \newcount\sublock@disp
285 \newcommand{\sublockdisp}[1]{%
286     \l@dgetlock@disp{#1}%
287     \ifnum\@l@dttempcntb>\m@ne
288         \global\sublock@disp=\@l@dttempcntb
289     \else
290         \led@warn@BadSublockdisp
291     \fi}}
292

```

`\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.

```

293 \newcommand*{\linenumberstyle}[1]{%
294     \def\linenumrep##1{\@nameuse{##1}{##1}}
295 \newcommand*{\sublinenumberstyle}[1]{%
296     \def\sublinenumrep##1{\@nameuse{##1}{##1}}
297
298     Initialise the number styles to arabic.
297 \linenumberstyle{arabic}
298 \let\linenumr@p\linenumrep
299 \sublinenumberstyle{arabic}
300 \let\sublinenumr@p\sublinenumrep
301

```

`\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.

```

302 \newlength{\linenumsep}
303 \setlength{\linenumsep}{1pc}
304 \newcommand*{\numlabfont}{\normalfont\scriptsize}
305 \newcommand*{\ledlinenum}{%
306   \numlabfont\linenumrep{\line@num}%
307   \ifsublines@
308     \ifnum\subline@num>0\relax
309       \unskip\fullstop\sublinenumrep{\subline@num}%
310     \fi
311   \fi}
312 \newcommand*{\leftlinenum}{%
313   \ledlinenum
314   \kern\linenumsep}
315 \newcommand*{\rightlinenum}{%
316   \kern\linenumsep
317   \ledlinenum}
318
```

## 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.

```
319 \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`.

```
320 \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

`\@toksa`  
`\@toksb`

current value of a counter. It creates global control sequences, like `\xdef`, and uses two temporary token-list registers, `\@toksa` and `\@toksb`.

```
321 \newtoks\@toksa \newtoks\@toksb
322 \global\@toksa={\}
323 \long\def\xright@appenditem#1\to#2{%
324   \global\@toksb=\expandafter{#2}%
325   \xdef#2{\the\@toksb\the\@toksa\expandafter{#1}}%
326   \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`.

```
327 \long\def\xleft@appenditem#1\to#2{%
328   \global\@toksb=\expandafter{#2}%
329   \xdef#2{\the\@toksa\expandafter{#1}\the\@toksb}%
330   \global\@toksb={}}
```

`\gl@p` The `\gl@p` macro removes the leftmost item from a list and places it in a control sequence. You say `\gl@p\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 `\gl@p` are all global.

```
331 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
332 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
333
```

### 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`.

```
334 \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.

```
335 \newcount\subline@num
```

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

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

```
336 \newif\ifsublines@
```

`\absline@num` The count `\absline@num` 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 `\line@num` will often be equal to it. It is used internally to keep track of where notes are to appear and where new pages start: using this value rather than `\line@num` is a lot simpler, because it doesn't depend on the lineation system in use.

```
337 \newcount\absline@num
```

We'll be calling `\absline@num` numbers 'absolute' numbers, and `\line@num` and `\subline@num` numbers 'visible' numbers.

`\@lock` The counts `\@lock` and `\sub@lock` tell us the state of line-number and sub-line-  
`\sub@lock` 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.

```
338 \newcount\@lock
```

```
339 \newcount\sub@lock
```

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

`\actionlines@list`

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

`\actions@list`

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

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

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

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

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

```

340     \list@create{\line@list}
341     \list@create{\insertlines@list}
342     \list@create{\actionlines@list}
343     \list@create{\actions@list}
344

```

```

\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
345 \newcount\page@num
346 \newcount\endpage@num
347 \newcount\endline@num
348 \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.

```
349 \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.

```
350 \newread\@inputcheck
351 \newcommand*{\read@linelist}[1]{%
352   \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.

```
353   \get@linelistfile{#1}%
354   \endgroup
355
```

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.

```
356   \global\page@num=\m@ne
357   \ifx\actionlines@list\empty
358     \gdef\next@actionline{1000000}%
359   \else
360     \gl@p\actionlines@list\to\next@actionline
361     \gl@p\actions@list\to\next@action
```

```

362 \fi}
363
\list@clearing@reg Clears the lists for \read@linelist
364 \newcommand*\list@clearing@reg}{%
365 \list@clear{\line@list}%
366 \list@clear{\insertlines@list}%
367 \list@clear{\actionlines@list}%
368 \list@clear{\actions@list}}

\get@linelistfile ledmac can take advantage of the LaTeX ‘safe file input’ macros to get the line-list
file.
369 \newcommand*\get@linelistfile}[1]{%
370 \InputIfFileExists{#1}{%
371 \global\noteschanged@false
372 \begingroup
373 \catcode'\ [=1 \catcode'\ ]=2
374 \makeatletter \catcode'\ ^^M=9}{%
375 \led@warn@NoLineFile{#1}%
376 \global\noteschanged@true
377 \begingroup}%
378 }
379

```

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, `\@1`, 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`.

`\@1` `\@1` does everything related to the start of a new line of numbered text.  
`\@1@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:

```
\@1{<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.

```
380 \newcommand{\@1}[2]{%
381   \fix@page{#1}%
382   \@1@reg}
383 \newcommand*{\@1@reg}{%
384   \ifx\l@dchset@num\relax \else
385     \advance\absline@num \@ne
386     \set@line@action
387     \let\l@dchset@num=\relax
388     \advance\absline@num \m@ne
389     \advance\line@num \m@ne
390   \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.

```
391 \advance\absline@num \@ne
392   \ifx\next@page@num\relax \else
393     \page@action
394     \let\next@page@num=\relax
395   \fi
396   \ifx\sub@change\relax \else
397     \ifnum\sub@change>\z@
398       \sublines@true
399     \else
400       \sublines@false
401     \fi
402     \sub@action
403     \let\sub@change=\relax
404   \fi
```

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

```
405   \ifcase\@lock
```

```

406         \or
407         \@lock \tw@
408         \or \or
409         \@lock \z@
410     \fi
411     \ifcase\sub@lock
412         \or
413         \sub@lock \tw@
414         \or \or
415         \sub@lock \z@
416     \fi
    Now advance the visible line number, unless it's been locked.
417     \ifsublines@
418         \ifnum\sub@lock<\tw@
419             \advance\subline@num \@ne
420         \fi
421     \else
422         \ifnum\@lock<\tw@
423             \advance\line@num \@ne \subline@num \z@
424         \fi
425     \fi}
426

```

`\@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.

```

427 \newcommand*\@page}[1]{%
428   \ifbypage@
429     \line@num \z@ \subline@num \z@
430   \fi
431   \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.

```

432   \def\next@page@num{#1}}
433

```

`\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`.

```

434 \newcount\last@page@num
435 \last@page@num=-10000
436 \newcommand*\fix@page}[1]{%
437   \ifnum #1=\last@page@num
438   \else
439     \ifbypage@
440       \line@num=\z@ \subline@num=\z@
441     \fi
442   \page@num=#1\relax

```

```

443   \last@page@num=#1\relax
444   \def\next@page@num{#1}%
445   \fi}
446

```

\@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 447 \newcommand*\@pend}[1]{
448 \newcommand*\@pendR}[1]{
449 \newcommand*\@lopL}[1]{
450 \newcommand*\@lopR}[1]{
451

```

\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.

```

452 \newcommand*\sub@on{\ifsublines@
453   \let\sub@change=\relax
454   \else
455   \def\sub@change{1}%
456   \fi}
457 \newcommand*\sub@off{\ifsublines@
458   \def\sub@change{-1}%
459   \else
460   \let\sub@change=\relax
461   \fi}
462

```

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

```

463 \newcommand*\@adv}[1]{\ifsublines@
464   \advance\subline@num by #1\relax
465   \ifnum\subline@num<\z@
466     \led@warn@BadAdvancelineSubline
467     \subline@num \z@
468   \fi
469   \else
470     \advance\line@num by #1\relax
471     \ifnum\line@num<\z@
472       \led@warn@BadAdvancelineLine
473       \line@num \z@
474     \fi
475   \fi
476   \set@line@action}
477

```

\@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.

```

478 \newcommand*{\@set}[1]{\ifsublines@
479   \subline@num=#1\relax
480 \else
481   \line@num=#1\relax
482 \fi
483 \set@line@action}
484

```

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

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

```

485 \newcommand*{\l@d@set}[1]{%
486 \line@num=#1\relax
487 \advance\line@num \@ne
488 \def\l@dchset@num{#1}}
489 \let\l@dchset@num\relax
490

```

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

```

491 \newcommand*{\page@action}{%
492 \xright@appenditem{\the\absline@num}\to\actionlines@list
493 \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.

```

494 \newcommand*{\set@line@action}{%
495 \xright@appenditem{\the\absline@num}\to\actionlines@list
496 \ifsublines@
497   \l@dtempcnta=-\subline@num
498 \else
499   \l@dtempcnta=-\line@num
500 \fi
501 \advance\l@dtempcnta by -5000
502 \xright@appenditem{\the\l@dtempcnta}\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.

```

503 \newcommand*{\sub@action}{%
504 \xright@appenditem{\the\absline@num}\to\actionlines@list
505 \ifsublines@
506   \xright@appenditem{-1001}\to\actions@list
507 \else
508   \xright@appenditem{-1002}\to\actions@list
509 \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 numbers or sub-line numbers.

`\do@lockonL`

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.

```

510 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
511 \newcommand*{\do@lockon}{%
512   \ifx\next\lock@off
513     \global\let\lock@off=\skip@lockoff
514   \else
515     \do@lockonL
516   \fi}
517 \newcommand*{\do@lockonL}{%
518   \xright@appenditem{\the\absline@num}\to\actionlines@list
519   \ifsublines@
520     \xright@appenditem{-1005}\to\actions@list
521     \ifnum\sub@lock=\z@
522       \sub@lock \@ne
523     \else
524       \ifnum\sub@lock=\thr@@
525         \sub@lock \@ne
526       \fi
527     \fi
528   \else
529     \xright@appenditem{-1003}\to\actions@list
530     \ifnum\@lock=\z@
531       \@lock \@ne
532     \else
533       \ifnum\@lock=\thr@@
534         \@lock \@ne
535       \fi
536     \fi
537   \fi}
538

```

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

```

\do@lockoff 539 \newcommand*{\do@lockoffL}{%
\do@lockoffL 540   \xright@appenditem{\the\absline@num}\to\actionlines@list
\skip@lockoff 541   \ifsublines@
542     \xright@appenditem{-1006}\to\actions@list
543     \ifnum\sub@lock=\tw@
544       \sub@lock \thr@@
545     \else
546       \sub@lock \z@
547     \fi
548   \else
549     \xright@appenditem{-1004}\to\actions@list
550     \ifnum\@lock=\tw@
551       \@lock \thr@@
552     \else

```



```

553     \@lock \z@
554     \fi
555     \fi}
556 \newcommand*{\do@lockoff}{\do@lockoffL}
557 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
558 \global\let\lock@off=\do@lockoff
559

```

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

```

560 \newcommand*{\n@num}{\n@num@reg}
561 \newcommand*{\n@num@reg}{%
562   \xright@appenditem{\the\absline@num}\to\actionlines@list
563   \xright@appenditem{-1007}\to\actions@list}
564

```

`\@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`.

```

565     \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.

```

566 \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.

```

567 \newcommand*{\@ref}[2]{%
568   \@ref@reg{#1}{#2}}
569 \newcommand*{\@ref@reg}[2]{%
570   \global\insert@count=#1\relax
571   \loop\ifnum\insert@count>\z@
572     \xright@appenditem{\the\absline@num}\to\insertlines@list
573     \global\advance\insert@count \m@ne
574   \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.

```

575 \begingroup
576   \let\@ref=\dummy@ref
577   \let\page@action=\relax
578   \let\sub@action=\relax
579   \let\set@line@action=\relax
580   \let\@lab=\relax
581   #2
582   \global\endpage@num=\page@num
583   \global\endline@num=\line@num
584   \global\endsubline@num=\subline@num
585 \endgroup

```

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

```

586   \xright@appenditem%
587     {\the\page@num|\the\line@num|%
588     \ifsublines@ \the\subline@num \else 0\fi}%
589     \the\endpage@num|\the\endline@num|%
590     \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.

```

591 #2}
592

```

## 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.
593 \newwrite\linenum@out

```

```

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

```

Once any file is opened on this stream, we keep it open forever, or else switch to another file that we keep open. The reason is that we want the output routine to write the page number for every page to this file; otherwise we'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`.

```

594 \newif\iffirst@linenum@out@
595   \first@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.

```
596 \newcommand*{\line@list@stuff}[1]{%
    First, use the commands of the previous section to interpret the line-list file
    from the last run.
597 \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.
598 \iffirst@linenum@out@
599 \immediate\closeout\linenum@out
600 \global\first@linenum@out@false
601 \immediate\openout\linenum@out=#1\relax
602 \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.
603 \closeout\linenum@out
604 \openout\linenum@out=#1\relax
605 \fi}
606
```

`\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.

```
607 \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 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.

```
608 \newcommand*{\flag@start}{%
609 \edef\next{\write\linenum@out{%
610 \string\@ref[\the\insert@count] []}}%
611 \next}
612 \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.

```
613 \newcommand*{\page@start}{%
614
```

`\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.

```
615 \newcommand*{\startsub}{\dimen0\lastskip
616   \ifdim\dimen0>0pt \unskip \fi
617   \write\linenum@out{\string\sub@on}%
618   \ifdim\dimen0>0pt \hskip\dimen0 \fi}
619 \def\endsub{\dimen0\lastskip
620   \ifdim\dimen0>0pt \unskip \fi
621   \write\linenum@out{\string\sub@off}%
622   \ifdim\dimen0>0pt \hskip\dimen0 \fi}
623
```

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

```
624 \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.

```
625 \newcommand*{\setline}[1]{%
626   \ifnum#1<\z@
627     \led@warn@BadSetline
628   \else
629     \write\linenum@out{\string\@set[#1]}%
630   \fi}
631
```

`\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.

```
632 \newcommand*{\setlinenum}[1]{%
633   \ifnum#1<\z@
634     \led@warn@BadSetlinenum
635   \else
636     \write\linenum@out{\string\l@d@set[#1]}%
637   \fi}
638
```

```

\startlock  You can use \startlock or \endlock in running text to start or end line number
\endlock    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.

639 \newcommand*{\startlock}{\write\linenum@out{\string\lock@on}}
640 \def\endlock{\write\linenum@out{\string\lock@off}}
641

\ifl@dskipnumber  In numbered text \skipnumbering will suspend the numbering for that particular
\l@dskipnumbertrue line.
\l@dskipnumberfalse
\skipnumbering 642 \newif\ifl@dskipnumber
\skipnumbering@reg 643 \l@dskipnumberfalse
\skipnumbering@reg 644 \newcommand*{\skipnumbering}{\skipnumbering@reg}
645 \newcommand*{\skipnumbering@reg}{%
646 \write\linenum@out{\string\n@num}%
647 \advanceline{-1}}
648

```

## 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.

```
649 \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.

```
650 \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.

```
651 \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` `\morenoexpands` We need to turn off macro expansion for certain sorts of macros we're likely to see within the lemma and within the notes.

The first class is font-changing macros. We suppress expansion for them by letting them become equal to zero.<sup>23</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

---

<sup>23</sup>Since 'control sequences equivalent to characters are not expandable'—*The TeXbook*, answer to Exercise 20.14.

read but it works just the same. The ones that cause problems are: those that use alignments— $\TeX$  seems to get confused about the difference between alignment parameters and macro parameters; those that use temporary control sequences; and those that look carefully at what the current font is.

(The `\copyright` macro defined in PLAIN  $\TeX$  has this sort of problem as well, but 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.)

```

652 \newcommand*\noexpands{\let\rm=0\let\it=0\let\sl=0\let\bf=0\let\tt=0%
653   \let\b=0\let\c=0\let\d=0\let\t=0%
654   \let\select@lemmafont=0%
655   \def\protect{\noexpand\protect\noexpand}%
656   \let\startsub=\relax \let\endsub=\relax
657   \let\startlock=\relax \let\endlock=\relax
658   \let\edlabel=\@gobble
659 % \let\edpageref=\@gobble
660 % \let\lineref=\@gobble
661 % \let\sublineref=\@gobble
662   \let\setline=\@gobble \let\advanceline=\@gobble
663   \let\critext=\dummy@text
664   \let\edtext=\dummy@edtext
665   \l@dtabnoexpands
666   \morenoexpands}
667 \let\morenoexpands=\relax
668

```

`\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.

```
669 \long\def\critext#1#2/{\leavevmode
```

`\@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.

```
670 \begingroup
671 \no@expands
672 \xdef\@tag{\protect#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`.

```
673 \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`.

```
674 \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.

```
675 \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.

```
676 \flag@start
677 \endgroup
678 \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.

```

679 \ifx\end@lemmas\empty \else
680   \glp\end@lemmas\to\x@lemma
681   \x@lemma
682   \global\let\x@lemma=\relax
683 \fi
684 \flag@end}

```

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

```

\edtext
685 \newcommand{\edtext}[2]{\leavevmode
686   \begingroup
687   \no@expands
688   \xdef\@tag{\protect#1}%
689   \set@line
690   \global\insert@count=0
691   \ignorespaces #2\relax
692   \flag@start
693 \endgroup
694 \showlemma{#1}%
695 \ifx\end@lemmas\empty \else
696   \glp\end@lemmas\to\x@lemma
697   \x@lemma
698   \global\let\x@lemma=\relax
699 \fi
700 \flag@end}
701

```

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

```

702 \newif\ifnumberline
703 \numberlinetrue
704 % \end{macrocode}
705 % \end{macro}
706 % \begin{macro}{\set@line}
707 % The \cs{set@line} macro
708 % is called by \cs{critext} to put the line-reference field and
709 % font specifier for the current block of text into \cs{l@d@nums}.
710 %
711 % One instance of \cs{critext} may generate several notes, or it
712 % may generate none---it's legitimate for argument \verb"#2" to \cs{critext} to
713 % be empty. But \cs{flag@start} and \cs{flag@end} induce the generation of
714 % a single entry in \cs{line@list} during the next run, and it's vital
715 % to also remove one and only one \cs{line@list} entry here.
716 %   \begin{macrocode}
717 \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.

```

718 \ifx\line@list\empty

```

```

719   \global\noteschanged>true
720   \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
721   \else
722     \gl@p\line@list\to\@tempb
723     \xdef\l@d@nums{\@tempb|\edfont@info}%
724     \global\let\@tempb=\undefined
725   \fi}
726

```

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

```

727 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
728

```

## 20.2 Substitute lemma

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

```

729 \newcommand*{\lemma}[1]{\xdef\@tag{\protect#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.55): 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.

```

730 \newcommand*{\linenum}[1]{%
731   \xdef\@tempa{#1|||||\noexpand\\ \l@d@nums}%
732   \global\let\l@d@nums=\empty
733   \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.

```

734 \def\line@set#1|#2\|#3|#4\\{%
735   \gdef\@tempb{#1}%
736   \ifx\@tempb\empty
737     \l@d@add{#3}%
738   \else
739     \l@d@add{#1}%

```

```

740 \fi
741 \gdef\@tempb{#4}%
742 \ifx\@tempb\empty\else
743     \l@d@add{|\line@set#2\|#4\}%
744 \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`.

```

745 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
746

```

## 21 Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than straight onto the vertical list, and when the paragraph ends we slice the paragraph into its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

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

`\raw@text` Here are numbers and flags that are used internally in the course of the paragraph decomposition.

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

```

747 \newbox\raw@text
748 \newif\ifnumberedpar@
749 \newcount\num@lines
750 \newbox\one@line
751 \newcount\par@line

```

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

Beware: everything that occurs between `\pstart` and `\pend` is happening within a group; definitions must be global if you want them to survive past the end of the paragraph.

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

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

```
752
753 \newcounter{pstart}
754 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
755 \newif\ifnumberpstart
756 \numberpstartfalse
757 \newcommand*{\pstart}{
758 \if@nobreak
759 \let\@oldnobreak\@nobreaktrue
760 \else
761 \let\@oldnobreak\@nobreakfalse
762 \fi
763 \@nobreaktrue
764 \ifnumbering \else
765   \led@err@PstartNotNumbered
766   \beginnumbering
767   \fi
768   \ifnumberedpar@
769   \led@err@PstartInPstart
770   \pend
771   \fi
772   \list@clear{\inserts@list}%
773   \global\let\next@insert=\empty
774   \begingroup\normal@pars
775   \global\setbox\raw@text=\vbox\bgroup\ifautopar\else\ifnumberpstart\ifinstanza\else\ifsidepstartnum
776   \numberedpar@true}
```

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

```
777 \newcommand*{\pend}{\ifnumbering \else
778   \led@err@PendNotNumbered
779   \fi
780   \ifnumberedpar@ \else
781   \led@err@PendNoPstart
782   \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.

```
783 \l@dzeropenalties
784 \endgraf\global\num@lines=\prevgraf\egroup
```

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`.

```

785 \newcount\pstartline%
786 \pstartline=0\relax%
787 \loop\ifvbox\raw@text%
788   \advance\pstartline 1%
789   \do@line%
790   \ifbypstart@%
791     \ifnum\pstartline=1%
792       \setline{1}%
793     \fi%
794   \fi%
795 \repeat

```

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

```

796 \flush@notes
797 \endgroup
798 \ignorespaces
799 \ifnumberpstart
800 \pstartnumtrue
801 \fi
802 \@oldnobreak
803 \addtocounter{pstart}{1}
804

```

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

```

805 \newcommand*\l@dzzeropenalties{%
806   \brokenpenalty \z@ \clubpenalty \z@
807   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
808   \postdisplaypenalty \z@ \widowpenalty \z@}
809

```

`\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.

```

810 \newif\ifautopar
811 \autoparfalse
812 \newcommand*\autopar>{
813   \ifledRcol
814     \ifnumberingR \else
815       \led@err@AutoparNotNumbered
816       \beginnumberingR
817       \fi
818     \else
819       \ifnumbering \else
820       \led@err@AutoparNotNumbered
821       \beginnumbering
822       \fi
823     \fi
824   \autopartrue
825   \everypar={\setbox0=\lastbox
826     \endgraf \vskip-\parskip
827     \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\fi\fi
828     \let\par=\pend}%
829   \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.

```

830 \newcommand*\normal@pars){\everypar={}\let\par\endgraf}
831

```

## 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.

```

832 \newcommand*\l@dunhbox@line}[1]{\unhbox #1}
833 \newcommand*\do@line){%
834   {\vbadness=10000
835     \splittopskip=\z@
836     \do@linehook
837   \l@emptyd@ta
838   \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%

```

```

839 \unvbox\one@line \global\setbox\one@line=\lastbox
840 \getline@num
841 \ifnum\@lock>\@ne
842   \inserthangingsymboltrue
843 \else
844   \inserthangingsymbolfalse
845 \fi
846 \affixline@num
847 \affixstart@num
848 \hb@xt@ \linewidth{\inserthangingsymbol\l@dld@ta\add@inserts\affixside@note
849   \l@dlsn@te
850   {\ledllfill\hb@xt@ \wd\one@line{\new@line\l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta
851   \l@drsn@te
852   }}%
853 \add@penalties%
854 }%

\do@linehook A hook into \do@line.
855 \newcommand*\do@linehook{}

\l@emptyd@ta Nulls the \. . .d@ta, which may later hold line numbers. Similarly for \l@dcsnotetext
\l@dld@ta for the text of a sidenote.
\l@drd@ta 856 \newcommand*\l@emptyd@ta{}%
\l@dcsnotetext 857 \gdef\l@dld@ta{}%
858 \gdef\l@drd@ta{}%
859 \gdef\l@dcsnotetext{}%
860

\l@dlsn@te Zero width boxes of the left and right side notes, together with their kerns.
\l@drsn@te 861 \newcommand*\l@dlsn@te{}%
862 \hb@xt@ \z@{\hss\box\l@dldp@rbox\kern\ledlsnotesep}}
863 \newcommand*\l@drsn@te{}%
864 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
865

\ledllfill These macros are called at the left (\ledllfill) and the right (\ledrlfill) of each
\ledrlfill numbered line. The initial definitions correspond to the original code for \do@line.
866 \newcommand*\ledllfill{\hfil}
867 \newcommand*\ledrlfill{}
868

```

### 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.
869 \newcommand*\getline@num{}%
870 \ifnumberline
871   \global\advance\absline@num \@ne

```



```

872 \fi
873 \do@actions
874 \do@ballast
875 \ifnumberline
876 \ifsublines@
877   \ifnum\sub@lock<\tw@
878     \global\advance\subline@num \@ne
879   \fi
880 \else
881   \ifnum\@lock<\tw@
882     \global\advance\line@num \@ne
883     \global\subline@num \z@
884   \fi
885 \fi
886 \fi
887 }

```

`\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. 89).

`\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}{<some figure>}` in your document.

```

888 \newcount\ballast@count
889 \newcounter{ballast}
890 \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.

```

891 \newcommand*{\do@ballast}{\global\ballast@count \z@
892   \begingroup
893     \advance\absline@num \@ne
894     \ifnum\next@actionline=\absline@num
895       \ifnum\next@action>-1001\relax
896         \global\advance\ballast@count by -\c@ballast
897       \fi
898     \fi
899   \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`.

```
900 \newcommand*{\do@actions}{%
901   \global\let\do@actions@next=\relax
902   \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.

```
903   \ifnum\next@action>-1001
904     \global\page@num=\next@action
905     \ifbypage@
906       \global\line@num=\z@ \global\subline@num=\z@
907     \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`.)

```
908   \else
909     \ifnum\next@action<-4999
910       \@l@tempcnta=-\next@action
911       \advance\@l@tempcnta by -5001
912       \ifsublines@
913         \global\subline@num=\@l@tempcnta
914       \else
915         \global\line@num=\@l@tempcnta
916     \fi
```

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

```
917   \else
918     \@l@tempcnta=-\next@action
919     \advance\@l@tempcnta by -1000
920     \do@actions@fixedcode
921   \fi
922 \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.

```
923   \ifx\actionlines@list\empty
924     \gdef\next@actionline{1000000}%
925   \else
926     \glp\actionlines@list\to\next@actionline
927     \glp\actions@list\to\next@action
928     \global\let\do@actions@next=\do@actions
929   \fi
930 \fi
```

Make the recursive call, if necessary.

```
931 \do@actions@next}
932
```

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

```
933 \newcommand*{\do@actions@fixedcode}{%
934   \ifcase\l@dttempcnta
935   \or%           % 1001
936     \global\sublines@true
937   \or%           % 1002
938     \global\sublines@false
939   \or%           % 1003
940     \global\@lock=\@ne
941   \or%           % 1004
942     \ifnum\@lock=\tw@
943       \global\@lock=\thr@@
944     \else
945       \global\@lock=\z@
946     \fi
947   \or%           % 1005
948     \global\sub@lock=\@ne
949   \or%           % 1006
950     \ifnum\sub@lock=\tw@
951       \global\sub@lock=\thr@@
952     \else
953       \global\sub@lock=\z@
954     \fi
955   \or%           % 1007
956     \l@dskipnumbertrue
957   \else
958     \led@warn@BadAction
959   \fi}
960
961
```

## 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  $\backslash\text{line@num} \leq \backslash\text{firstlinenum}$ , we compare the two directly instead of making these calculations.

We compute, in the scratch counter  $\backslash\text{@l@tempcnta}$ , the number of the next line that should be printed with a number (*m* in the above discussion), and move the current line number into the counter  $\backslash\text{@l@tempcntb}$  for comparison.

Remember that some counts are now counters!

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

```
962 \newcommand*{\affixline@num}{%
```

No number is attached if  $\backslash\text{ifl@dskipnumber}$  is TRUE (and then it is set to its normal FALSE value). No number is attached if  $\backslash\text{ifnumberline}$  is FALSE (the normal value is TRUE).

```
963 \ifnumberline
964 \ifl@dskipnumber
965   \global\l@dskipnumberfalse
966 \else
967   \ifsublines@
968     \l@tempcntb=\subline@num
969     \ifnum\subline@num>\c@firstsublinenum
970       \l@tempcnta=\subline@num
971       \advance\l@tempcnta by-\c@firstsublinenum
972       \divide\l@tempcnta by\c@sublinenumincrement
973       \multiply\l@tempcnta by\c@sublinenumincrement
974       \advance\l@tempcnta by\c@firstsublinenum
975     \else
976       \l@tempcnta=\c@firstsublinenum
977   \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.

```
978   \ch@cksub@l@ck
```

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

```
979   \else
980     \l@tempcntb=\line@num
```

Check on the  $\backslash\text{linenumberlist}$  If it's  $\backslash\text{empty}$  use the standard algorithm.

```
981   \ifx\linenumberlist\empty
982     \ifnum\line@num>\c@firstlinenum
983       \l@tempcnta=\line@num
984       \advance\l@tempcnta by-\c@firstlinenum
985       \divide\l@tempcnta by\c@linenumincrement
986       \multiply\l@tempcnta by\c@linenumincrement
987       \advance\l@tempcnta by\c@firstlinenum
988     \else
```

```

989     \@l@tempcnta=\c@firstlinenum
990     \fi
991     \else

```

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

```

992     \@l@tempcnta=\line@num
993     \edef\rem@inder{\linenumberlist,\number\line@num,}%
994     \edef\sc@n@list{\def\noexpand\sc@n@list
995     ###1,\number\@l@tempcnta,###2|{\def\noexpand\rem@inder{###2}}}%
996     \sc@n@list\expandafter\sc@n@list\rem@inder|%
997     \ifx\rem@inder\empty\advance\@l@tempcnta\@ne\fi
998     \fi

```

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

```

999     \ch@ck@l@ck
1000    \fi

```

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

```

1001    \ifnum\@l@tempcnta=\@l@tempcntb

```

If we got here, we're going to print a line number; so now we need to calculate a number that will tell us which side of the page will get the line number. We start from `\line@margin`, which asks for one side always if it's less than 2; and then if the side does depend on the page number, we simply add the page number to this side code—because the values of `\line@margin` have been devised so that this produces a number that's even for left-margin numbers and odd for right-margin numbers.

For LaTeX we have to consider two column documents as well. In this case I think we need to put the numbers at the outside of the column — the left of the first column and the right of the second. Do the twocolumn stuff before going on with the original code.

`\l@dld@ta` A left line number is stored in `\l@dld@ta` and a right one in `\l@drd@ta`.

```

\l@drd@ta 1002 \if@twocolumn
1003     \if@firstcolumn
1004         \gdef\l@dld@ta{\llap{\leftlinenum}}%
1005     \else
1006         \gdef\l@drd@ta{\rlap{\rightlinenum}}%
1007     \fi
1008 \else

```

Continuing the original code ...

```

1009     \@l@tempcntb=\line@margin
1010     \ifnum\@l@tempcntb>\@ne
1011         \advance\@l@tempcntb \page@num
1012     \fi

```

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

```

1013     \ifodd\@l@tempcntb
1014         \gdef\l@drd@ta{\rlap{\rightlinenum}}%

```

```

1015 \else
1016 \gdef\l@dld@ta{\llap{\leftlinenum}}%
1017 \fi
1018 \fi
1019 \else

```

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

```

1020 %% #1%
1021 \fi

```

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

```

1022 \f@x@l@cks
1023 \fi
1024 \fi
1025 }
1026

```

`\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.

```

1027 \newcommand*\ch@cksub@l@ck}{%
1028 \ifcase\sub@lock
1029 \or
1030 \ifnum\sublock@disp=\@ne
1031 \l@dtempcntb=\z@ \l@dtempcnta=\@ne
1032 \fi
1033 \or
1034 \ifnum\sublock@disp=\tw@ \else
1035 \l@dtempcntb=\z@ \l@dtempcnta=\@ne
1036 \fi
1037 \or
1038 \ifnum\sublock@disp=\z@
1039 \l@dtempcntb=\z@ \l@dtempcnta=\@ne
1040 \fi
1041 \fi}

```

Similarly for line numbers.

```

1042 \newcommand*\ch@ck@l@ck}{%
1043 \ifcase\@lock
1044 \or
1045 \ifnum\lock@disp=\@ne
1046 \l@dtempcntb=\z@ \l@dtempcnta=\@ne
1047 \fi
1048 \or
1049 \ifnum\lock@disp=\tw@ \else
1050 \l@dtempcntb=\z@ \l@dtempcnta=\@ne
1051 \fi
1052 \or
1053 \ifnum\lock@disp=\z@

```

```

1054         \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1055         \fi
1056     \fi}

1057 \newcommand*{\f@x@l@cks}{%
1058     \ifcase\@lock
1059     \or
1060     \global\@lock=\tw@
1061     \or \or
1062     \global\@lock=\z@
1063     \fi
1064     \ifcase\sub@lock
1065     \or
1066     \global\sub@lock=\tw@
1067     \or \or
1068     \global\sub@lock=\z@
1069     \fi}
1070

```

`\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.

```

1071 \newcommand{\pageparbreak}{\pend\newpage\pstart\noindent}
1072

```

## 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 1073
\ifsidepstartnum 1074 \newif\ifsidepstartnum
1075 \newcommand*{\affixpstart@num}{%
1076     \ifsidepstartnum
1077     \if@twocolumn

```

```

1078     \if@firstcolumn
1079         \gdef\l@dld@ta{\llap{\leftstartnum}}}%
1080     \else
1081         \gdef\l@drd@ta{\rlap{\rightstartnum}}}%
1082     \fi
1083 \else
1084     \l@tempcntb=\line@margin
1085     \ifnum\l@tempcntb>\@ne
1086         \advance\l@tempcntb \page@num
1087     \fi
1088     \ifodd\l@tempcntb
1089         \gdef\l@drd@ta{\rlap{\rightstartnum}}}%
1090     \else
1091         \gdef\l@dld@ta{\llap{\leftstartnum}}}%
1092     \fi
1093 \fi
1094 \fi
1095
1096 }
1097 %
1098
1099 \newif\ifpstartnum
1100 \pstartnumtrue
1101 \newcommand*{\leftstartnum}{
1102     \ifpstartnum\thepstart
1103     \kern\linenumsep\fi
1104     \global\pstartnumfalse
1105 }
1106 \newcommand*{\rightstartnum}{
1107     \ifpstartnum
1108     \kern\linenumsep
1109     \thepstart
1110     \fi
1111     \global\pstartnumfalse
1112 }

```

## 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.

```
1113 \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`.

```
1114 \newcommand*{\add@inserts}{%
1115   \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.

```
1116   \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.

```
1117   \ifx\next@insert\empty
1118     \ifx\insertlines@list\empty
1119       \global\noteschanged@true
1120       \gdef\next@insert{100000}%
1121     \else
1122       \gl@p\insertlines@list\to\next@insert
1123     \fi
1124   \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.

```
1125   \ifnum\next@insert=\absline@num
1126     \gl@p\inserts@list\to\@insert
1127     \@insert
1128     \global\let\@insert=\undefined
1129     \global\let\next@insert=\empty
1130     \global\let\add@inserts@next=\add@inserts
1131   \fi
1132 \fi
```

Make the recursive call, if necessary.

```
1133 \add@inserts@next}
1134
```

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

```

1135 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
1136   \ifnum\num@lines>\@ne
1137     \global\advance\par@line \@ne
1138     \ifnum\par@line=\@ne
1139       \advance\@l@tempcnta \clubpenalty
1140     \fi
1141     \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
1142     \ifnum\@l@tempcntb=\num@lines
1143       \advance\@l@tempcnta \widowpenalty
1144     \fi
1145     \ifnum\par@line<\num@lines
1146       \advance\@l@tempcnta \interlinepenalty
1147     \fi
1148   \fi
1149   \ifnum\@l@tempcnta=\z@
1150     \relax
1151   \else
1152     \ifnum\@l@tempcnta>-10000
1153       \penalty\@l@tempcnta
1154     \else
1155       \penalty -10000
1156     \fi
1157   \fi}
1158

```

## 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.

```

1159 \newcommand*{\flush@notes}{%
1160   \@xloop
1161     \ifx\inserts@list\empty \else
1162       \gl@p\inserts@list\to\@insert
1163       \@insert
1164       \global\let\@insert=\undefined
1165   \repeat}
1166

```

`\@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  $\text{\TeX}$   $\backslash\text{loop}$  is used, too, so we could just call it  $\backslash\text{loop}$ ; but it seems preferable not to change the definitions of any of the standard macros.)

This variant of  $\backslash\text{loop}$  was introduced by Alois Kabelschacht in *TUGboat* 8 (1987), pp. 184–5.

```
1167 \def\@xloop#1\repeat{%
1168   \def\body{#1\expandafter\body\fi}%
1169   \body}
1170
```

## 22 Footnotes

The footnote macros are adapted from those in PLAIN  $\text{\TeX}$ , 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.

$\backslash\text{notefontsetup}$  The font setup defined in  $\backslash\text{notefontsetup}$  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  $\backslash\text{notefontsetup}$  effectively maps to LaTeX  $\backslash\text{footnotesize}$  for a 10pt document.

```
1171 \newcommand*{\notefontsetup}{\footnotesize}
```

$\backslash\text{notenumfont}$  The line numbers will be printed using the font selected by executing  $\backslash\text{notenumfont}$ .

The original font for  $\backslash\text{notenumfont}$  maps to LaTeX  $\backslash\text{scriptsize}$  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  $\backslash\text{notefontsetup}$ ).

```
1172 \newcommand*{\notenumfont}{\normalfont}
```

$\backslash\text{select@lemmafnt}$   $\backslash\text{select@lemmafnt}$  is provided to set the right font for the lemma in a note.

$\backslash\text{select@@lemmafnt}$  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.

```

1173 \def\select@lemmafонт#1|#2|#3|#4|#5|#6|#7|{\select@lemmafонт#7|}
1174 \def\select@lemmafонт#1/#2/#3/#4|{%
1175   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
1176   \selectfont}
1177

```

## 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.

```

1178 \newcommand*{\Afootnote}[1]{%
1179   \ifnumberedpar@
1180     \xright@appenditem{\noexpand\vAfootnote{A}%
1181       {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1182     \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`.

```

1183   \else
1184     \vAfootnote{A}{0|0|0|0|0|0|0}{#1}%
1185   \fi\ignorespaces}

```

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

```

\Cfootnote 1186 \newcommand*{\Bfootnote}[1]{%
\Dfootnote 1187   \ifnumberedpar@
\Efootnote 1188     \xright@appenditem{\noexpand\vBfootnote{B}%
1189       {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1190     \global\advance\insert@count \@ne
1191   \else
1192     \vBfootnote{B}{0|0|0|0|0|0|0}{#1}%
1193   \fi\ignorespaces}

1194 \newcommand*{\Cfootnote}[1]{%
1195   \ifnumberedpar@
1196     \xright@appenditem{\noexpand\vCfootnote{C}%
1197       {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1198     \global\advance\insert@count \@ne
1199   \else
1200     \vCfootnote{C}{0|0|0|0|0|0|0}{#1}%
1201   \fi\ignorespaces}

1202 \newcommand*{\Dfootnote}[1]{%
1203   \ifnumberedpar@

```

```

1204 \xright@appenditem{\noexpand\vDfootnote{D}%
1205     {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1206 \global\advance\insert@count \@ne
1207 \else
1208 \vDfootnote{D}{0|0|0|0|0|0|0|0}{#1}%
1209 \fi\ignorespaces}

1210 \newcommand*\Efootnote}[1]{%
1211 \ifnumberedpar@
1212 \xright@appenditem{\noexpand\vEfootnote{E}%
1213     {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1214 \global\advance\insert@count \@ne
1215 \else
1216 \vEfootnote{E}{0|0|0|0|0|0|0|0}{#1}%
1217 \fi\ignorespaces}
1218

```

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

```

\mpBfootins 1219 \newinsert\mpAfootins
\mpCfootins 1220 \newinsert\mpBfootins
\mpDfootins 1221 \newinsert\mpCfootins
\mpEfootins 1222 \newinsert\mpDfootins
1223 \newinsert\mpEfootins
1224

```

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

```

\mpBfootnote 1225 \newcommand*\mpAfootnote}[1]{%
\mpCfootnote 1226 \ifnumberedpar@
\mpDfootnote 1227 \xright@appenditem{\noexpand\mpvAfootnote{A}%
\mpEfootnote 1228     {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1229 \global\advance\insert@count \@ne
1230 \else
1231 \mpvAfootnote{A}{0|0|0|0|0|0|0|0}{#1}%
1232 \fi\ignorespaces}

1233 \newcommand*\mpBfootnote}[1]{%
1234 \ifnumberedpar@
1235 \xright@appenditem{\noexpand\mpvBfootnote{B}%
1236     {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1237 \global\advance\insert@count \@ne
1238 \else
1239 \mpvBfootnote{B}{0|0|0|0|0|0|0|0}{#1}%
1240 \fi\ignorespaces}

1241 \newcommand*\mpCfootnote}[1]{%
1242 \ifnumberedpar@
1243 \xright@appenditem{\noexpand\mpvCfootnote{C}%
1244     {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1245 \global\advance\insert@count \@ne
1246 \else
1247 \mpvCfootnote{C}{0|0|0|0|0|0|0|0}{#1}%
1248 \fi\ignorespaces}

```

```

1249 \newcommand*\mpDfootnote}[1]{%
1250   \ifnumberedpar@
1251     \xright@appenditem{\noexpand\mpvDfootnote{D}%
1252       {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1253     \global\advance\insert@count \@ne
1254   \else
1255     \mpvDfootnote{D}{\l@d@nums}{\@tag}{#1}%
1256   \fi\ignorespaces}

1257 \newcommand*\mpEfootnote}[1]{%
1258   \ifnumberedpar@
1259     \xright@appenditem{\noexpand\mpvEfootnote{E}%
1260       {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1261     \global\advance\insert@count \@ne
1262   \else
1263     \mpvEfootnote{E}{\l@d@nums}{\@tag}{#1}%
1264   \fi\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 T<sub>E</sub>X, 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.

```

1265 \newcommand*\normalvfootnote}[2]{%
1266   \insert\csname #1footins\endcsname\bgroup
1267   \notefontsetup
1268   \footsplitskips
1269   \spaceskip=\z@skip \xspaceskip=\z@skip
1270   \csname #1footfmt\endcsname #2\egroup}

```

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

```

1271 \newcommand*{\footsplitskips}{%
1272   \interlinepenalty=\interfootnotelinepenalty
1273   \floatingpenalty=\@MM
1274   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
1275   \leftskip=\z@skip \rightskip=\z@skip}
1276

```

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

```

1277 \newcommand*{\mpnormalvfootnote}[2]{%
1278   \global\setbox\@nameuse{mp#1footins}\vbox{%
1279     \unvbox\@nameuse{mp#1footins}
1280     \notefontsetup
1281     \hsize\columnwidth
1282     \@parboxrestore
1283     \color@begingroup
1284     \csname #1footfmt\endcsname #2\color@endgroup}}
1285

```

`\ledsetnormalparstuff` `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see p. 55), 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).

```

1286 \newcommand*{\ledsetnormalparstuff}{%
1287   \normal@pars
1288   \parindent \z@ \parfillskip \z@ \@plus 1fil}
1289 \newcommand*{\normalfootfmt}[3]{%
1290   \ledsetnormalparstuff
1291   {\notenumfont\printlines#1|}\strut\enspace
1292   {\select@lemmafnt#1|#2}\rbracket\enskip#3\strut\par}
1293

```

`\endashchar` The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals does not contain an en-dash or square brackets, and its period and comma are in odd locations. To allow use of the standard footnote macros with such fonts, we use the following macros for certain characters.

The `\endashchar` macro is simply an en-dash from the normal font and is immune to changes in the surrounding font. The same goes for the full stop. These two are used in `\printlines`. The right bracket macro is the same again; it crops up in `\normalfootfmt` and the other footnote macros for controlling the format of footnotes.

```

1294 \def\endashchar{\textnormal{--}}
1295 \newcommand*\fullstop{\textnormal{.}}
1296 \newcommand*\rbracket{\textnormal{\thinspace]]}}
1297

```

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 55: the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma.

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

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

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

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

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

```

\ifl@d@pnum
\ifl@d@ssub 1298 \newif\ifl@d@pnum
\ifl@d@elin 1299 \l@d@pnumfalse
\ifl@d@esl 1300 \newif\ifl@d@ssub
\ifl@d@dash 1301 \l@d@ssubfalse
1302 \newif\ifl@d@elin
1303 \l@d@elinfalse
1304 \newif\ifl@d@esl
1305 \l@d@eslfalse
1306 \newif\ifl@d@dash
1307 \l@d@dashfalse

```

```

\ifledplinenum Sometimes it could be useful not to print the line number, or give it a symbolic value
\symplinenum (perhaps if there are several notes from the same line).
1308 \newif\ifledplinenum
1309 \ledplinenumtrue
1310 \newcommand*\symplinenum{}
1311

```

```

\l@d@parsefootspec \l@d@parsefootspec{<spec>}{<lemma>}{<text>} parses a footnote specification.
\l@d@p@rsefootspec <lemma> and <text> are the lemma and text respectively. <spec> is the line and
\l@d@p@rsestartpage page number and lemma font specifier in \l@d@nums style format. The real work
\l@d@p@rsestartline is done by \l@d@p@rsefootspec which defines macros holding the numeric values.
\l@d@p@rsestartsub
\l@d@p@rseendpage
\l@d@p@rseendline
\l@d@p@rseendsub

```



```

1312 \newcommand*\l@dp@rsefootsec}[3]{\l@dp@rsefootsec#1|}
1313 \def\l@dp@rsefootsec#1|#2|#3|#4|#5|#6|#7|{%
1314   \gdef\l@dparsedstartpage{#1}%
1315   \gdef\l@dparsedstartline{#2}%
1316   \gdef\l@dparsedstartsub{#3}%
1317   \gdef\l@dparsedendpage{#4}%
1318   \gdef\l@dparsedendline{#5}%
1319   \gdef\l@dparsedendsub{#6}%
1320 }

```

Initialise the several number value macros.

```

1321 \def\l@dparsedstartpage{0}%
1322 \def\l@dparsedstartline{0}%
1323 \def\l@dparsedstartsub{0}%
1324 \def\l@dparsedendpage{0}%
1325 \def\l@dparsedendline{0}%
1326 \def\l@dparsedendsub{0}%
1327

```

`\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`.

```

1328 \newcommand*\setprintlines}[6]{%
1329   \l@d@pnumfalse \l@d@dashfalse
1330   \ifbypage@
1331     \ifnum#4=#1 \else
1332       \l@d@pnumtrue
1333       \l@d@dashtrue
1334     \fi
1335   \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.

```

1336   \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
1337   \ifnum#2=#5 \else
1338     \l@d@elintrue
1339     \l@d@dashtrue
1340   \fi

```

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

```

1341   \l@d@ssubfalse
1342   \ifnum#3=0 \else
1343     \l@d@ssubtrue
1344   \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.

```

1345 \l@d@eslfalse
1346 \ifnum#6=0 \else
1347     \ifnum#6=#3
1348         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
1349     \else
1350         \l@d@esltrue
1351         \l@d@dashtrue
1352     \fi
1353 \fi}

```

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

```

1354 \def\printlines#1|#2|#3|#4|#5|#6|#7|{\begingroup
1355     \ifbypstart%
1356     \ifl@dpairing%
1357         \ifledRcol%
1358             \thepstartR%
1359         \else%
1360             \thepstartL%
1361         \fi%
1362     \else%
1363         \thepstart%
1364     \fi%
1365 \fi%
1366 \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).

```

1367 \ifl@d@pnum #1\fullstop\fi

```

The other thing is whether to print the real starting line number or a symbolic value.

```

1368 \ifledplinenum \linenumrep{#2}\else \symplinenum\fi
1369 \ifl@d@ssub \fullstop \sublinenumrep{#3}\fi
1370 \ifl@d@dash \endashchar\fi
1371 \ifl@d@pnum #4\fullstop\fi
1372 \ifl@d@elin \linenumrep{#5}\fi
1373 \ifl@d@esl \ifl@d@elin \fullstop\fi \sublinenumrep{#6}\fi
1374 \endgroup}
1375

```

`\normalfootstart` `\normalfootstart` is a standard footnote-starting macro, called in the output routine whenever there are footnotes of this series to be printed: it skips a bit and then draws a rule.

Any `footstart` macro must put onto the page something that takes up space exactly equal to the `\skip\footins` value for the associated series of notes. `TEX` makes page computations based on that `\skip` value, and the output pages will suffer from spacing problems if what you add takes up a different amount of space.

The `\leftskip` and `\rightskip` values are both zeroed here. Similarly, these skips are cancelled in the `vfootnote` macros for the various types of notes. Strictly

speaking, this is necessary only if you are using paragraphed footnotes, but we have put it here and in the other `vfootnote` macros too so that the behavior of `ledmac` in this respect is general across all footnote types (you can change this). What this means is that any `\leftskip` and `\rightskip` you specify applies to the main text, but not the footnotes. The footnotes continue to be of width `\hsize`.

```
1376 \newcommand*{\normalfootstart}[1]{%
1377   \vskip\skip\csname #1footins\endcsname
1378   \leftskipOpt \rightskipOpt
1379   \csname #1footnoterule\endcsname}
```

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

```
1380 \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.

```
1381 \newcommand*{\normalfootgroup}[1]{\unvbox\csname #1footins\endcsname}
1382
```

`\mpnormalfootgroup` A somewhat different version for minipages.

```
1383 \newcommand*{\mpnormalfootgroup}[1]{%
1384   \vskip\skip@nameuse{mp#1footins}
1385   \normalcolor
1386   \@nameuse{#1footnoterule}
1387   \unvbox\csname mp#1footins\endcsname}}
1388
```

## 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 `\baselineskip`, 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\vsizel
\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`

```
1389 \newcommand*{\ledfootinsdim}{0.8\vsizel}
1390
```

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

```
1391 \newinsert\Afootins \newinsert\Bfootins
1392 \newinsert\Cfootins \newinsert\Dfootins
1393 \newinsert\Efootins
```

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

```
1394 \newcommand*{\footnormal}[1]{%
1395   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
1396   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
1397   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
1398   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
1399   \expandafter\let\csname #1footnoterule\endcsname=%
1400                                     \normalfootnoterule
1401   \count\csname #1footins\endcsname=1000
1402   \dimen\csname #1footins\endcsname=\ledfootinsdim
1403   \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).

```
1404   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
1405   \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
1406   \count\csname mp#1footins\endcsname=1000
1407   \dimen\csname mp#1footins\endcsname=\ledfootinsdim
1408   \skip\csname mp#1footins\endcsname=1.2em \@plus .6em \@minus .6em
1409 }
1410
```

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.

```
1411 \footnormal{A}
1412 \footnormal{B}
1413 \footnormal{C}
1414 \footnormal{D}
1415 \footnormal{E}
1416
```

## 22.5 Paraphrased footnotes

The paraphrased-footnote option reformats all the footnotes of one series for a page into a single paragraph; this is especially appropriate when the notes are numerous and brief. The code is based on *The TeXbook*, pp.398–400, with alterations for our environment. This algorithm uses a considerable amount of save-stack space: a T<sub>E</sub>X of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

`\footparagraph` The `\footparagraph` macro sets up everything for one series of footnotes so that they'll be paraphrased; it takes the series letter as argument. We include the setting of `\count\footins` to 1000 for the footnote series just in case you are switching to paraphrased footnotes after having columnar ones, since they change this value (see below).

It is important to call `\footparagraph` only after `\hsize` has been set for the pages that use this series of notes; otherwise T<sub>E</sub>X will try to put too many or too few of these notes on each page. If you need to change the `\hsize` within the document, call `\footparagraph` again afterwards to take account of the new value. The argument of `\footparagraph` is the letter (A–E) denoting the series of notes to be paraphrased.

```
1417 \newcommand*{\footparagraph}[1]{%
1418   \expandafter\let\csname #1footstart\endcsname=\parafootstart
1419   \expandafter\let\csname v#1footnote\endcsname=\para@vfootnote
1420   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
1421   \expandafter\let\csname #1footgroup\endcsname=\para@footgroup
1422   \count\csname #1footins\endcsname=1000
1423   \para@footsetup{#1}
```

And the extra setup for minipages.

```
1424   \expandafter\let\csname mpv#1footnote\endcsname=\mppara@vfootnote
1425   \expandafter\let\csname mp#1footgroup\endcsname=\mppara@footgroup
1426   \count\csname mp#1footins\endcsname=1000
1427 }
1428
```

You can redefine the `\parafootftmsep` command to print a separator between each paraphrased footnote (on the same page). A usual separator is a double pipe (`()`). To add double-pipe separators:

```
\renewcommand{\parafootftmsep}{\thinspace$||$\enspace}
```

`\footfudgefiddle` For paraphrased footnotes T<sub>E</sub>X 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.

```
1429 \providecommand{\footfudgefiddle}{64}
```

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

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

```
1430 \newcommand*{\para@footsetup}[1]{\notefontsetup
1431   \dimen0=\baselineskip
1432   \multiply\dimen0 by 1024
1433   \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
1434   \expandafter
1435   \xdef\csname #1footfudgefactor\endcsname{%
1436     \expandafter\strip@pt\dimen0 }}
1437
```

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.

```
1438 \newcommand*{\parafootstart}[1]{%
1439   \rightskip=0pt \leftskip=0pt \parindent=0pt
1440   \vskip\skip\csname #1footins\endcsname
1441   \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>24</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>25</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>26</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>27</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. 98 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.

```

1442 \newcommand*{\para@vfootnote}[2]{%
1443   \insert\csname #1footins\endcsname
1444   \bgroup
1445     \notefontsetup
1446     \footplitskips
1447     \setbox0=\vbox{\hsize=\maxdimen
1448       \noindent\csname #1footfmt\endcsname#2}%
1449     \setbox0=\hbox{\unvxh0}%
1450     \dp0=0pt

```

<sup>24</sup>Michael Downes, 'Line Breaking in `\unboxed` Text', *TUGboat* 11 (1990), pp. 605–612.

<sup>25</sup>See *The TeXbook*, p. 455 (editions after January 1990).

<sup>26</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>27</sup>'Line Breaking', p. 610.

```
1451 \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.

```
1452 \box0
1453 \penalty0
1454 \egroup}
1455
```

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 TeXbook*, 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.

```
1456 \newcommand*{\mppara@vfootnote}[2]{%
1457 \global\setbox\@nameuse{mp#1footins}\vbox{%
1458 \unvbox\@nameuse{mp#1footins}%
1459 \notefontsetup
1460 \footsplitskips
1461 \setbox0=\vbox{\hsize=\maxdimen
1462 \noindent\color@begingroup\csname #1footfmt\endcsname #2\color@endgroup}%
1463 \setbox0=\hbox{\unvxh0}%
1464 \dp0=\z@
1465 \ht0=\csname #1footfudgefactor\endcsname\wd0
1466 \box0
1467 \penalty0
1468 }}
1469
```

`\unvxh` Here is 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 TeXbook*, pp.99-100). `\unvxh` cancels these unwanted paragraph-final items using `\unskip` and `\unpenalty`.

```
1470 \newcommand*{\unvxh}[1]{%
1471 \setbox0=\vbox{\unvbox#1%
1472 \global\setbox1=\lastbox}%
1473 \unhbox1
1474 \unskip % remove \rightskip,
1475 \unskip % remove \parfillskip,
```



```

1476 \unpenalty          % remove \penalty of 10000,
1477 \hskip\ipn@skip}    % but add the glue to go between the notes
1478

```

`\interparanoteglue` Close observers will notice that we snuck some glue called `\ipn@skip` onto the end of the hbox produced by `\unvxh` in the above macro.

`\ipn@skip`

We want to be able to have some glue between our paragraphed footnotes. But since we are initially setting our notes in internal vertical mode, as little paragraphs, any paragraph-final glue will get discarded. Since `\unvxh` is already busy fiddling with glue and penalties at the end of these paragraphs, we take advantage of the opportunity to provide our inter-note spacing.

We collect the value of the inter-parafootnote glue value as the parameter of a macro called—wait for it—`\interparanoteglue`. We put this value into the value of a glue register `\ipn@skip` (inter-para-note-skip) making sure first to set the current font to the value normally used in footnotes so that the value of an `em` will be taken from the right font.

```

1479 \newskip\ipn@skip
1480 \newcommand*{\interparanoteglue}[1]{%
1481     {\notefontsetup\global\ipn@skip=#1 \relax}}
1482 \interparanoteglue{1em plus.4em minus.4em}
1483

```

There is a point to be careful about regarding the `\interparanoteglue`. 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 `\interparanoteglue`, but not its stretch or shrink components, since at this point the note has no need to stretch or shrink. Later, when the paragraph is actually composed by `\parafootgroup` in the output routine, `TEX` will almost certainly do some stretching and shrinking of this glue in order to make the paragraph look nice. Probably the stretching and shrinking over the whole paragraph will cancel each other out. But if not, the actual vertical size of the paragraph may not match the size the output routine had been told to expect, and you may get an overfull/underfull `\vbox` message from the output routine. To minimize the risk of this, you can do two things: keep the `plus` and `minus` components of `\interparanoteglue` small compared with its natural glue, and keep them the same as each other. As a general precaution, keep the size and flexibility of the `\skip\footins` glue on the high side too: because the reckoning is approximate, footnote blocks may be up to a line bigger or smaller than the output routine allows for, so keep some flexible space between the text and the notes.

`\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, and the third is the text of the footnote.

```

1484 \newcommand*{\parafootfmt}[3]{%
1485   \insertparafootftmsep%
1486   \ledsetnormalparstuff
1487   {\notenumfont\printlines#1}\enspace
1488   {\select@lemfont#1|#2}\rbracket\enskip
1489   #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 `\insertparafootftmsep` command is used to insert the `\parafootftmsep` 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.

```

1490 \newcommand*{\para@footgroup}[1]{%
1491   \unvbox\csname #1footins\endcsname
1492   \makehboxofhboxes
1493   \setbox0=\hbox{\unhbox0 \removehboxes}%
1494   \notefontsetup
1495   \noindent\unhbox0\par}
1496

```

`\mppara@footgroup` The minipage version.

```

1497 \newcommand*{\mppara@footgroup}[1]{%
1498   \vskip\skip\@nameuse{mp#1footins}
1499   \normalcolor
1500   \@nameuse{#1footnoterule}%
1501   \unvbox\csname mp#1footins\endcsname
1502   \makehboxofhboxes
1503   \setbox0=\hbox{\unhbox0 \removehboxes}%
1504   \notefontsetup
1505   \noindent\unhbox0\par}}
1506

```

`\makehboxofhboxes`

```

\removehboxes 1507 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
1508   \loop
1509     \unpenalty
1510     \setbox2=\lastbox
1511     \ifhbox2
1512       \setbox0=\hbox{\box2\unhbox0}%
1513     \repeat}
1514
1515 \newcommand*{\removehboxes}{\setbox0=\lastbox

```

```
1516 \ifhbox0{\removehboxes}\unhbox0 \fi}
1517
```

### 22.5.1 Insertion of footnotes separator

`\parafootftmsep` The `\parafootftmsep` macro is inserted between each paragraphed footnote. The default value is empty, but the user can redefine it via `\renewcommand`.

```
1518 \newcommand{\parafootftmsep}{}
```

The command `\insertparafootftmsep` must be called at the beginning of `\parafootftm` (and like commands). `\insertparafootftmsep` checks to see if the page number has changed since the previous note. If not, `\insertparafootftmsep` calls `\parafootftmsep`.

```
\prevpage@num
\insertparafootftmsep 1519 \newcount\prevpage@num
1520 \newcommand{\insertparafootftmsep}{%
1521   \ifnum\prevpage@num=\page@num%
1522     \parafootftmsep%
1523   \fi%
1524   \global\prevpage@num=\page@num%
1525 }
```

## 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`.

```
1526 \newcount\@k \newdimen\@h
1527 \newcommand*{\rigidbalance}[3]{\setbox0=\box#1 \@k=#2 \@h=#3
1528   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
1529   \valign{##\vfil\cr\dosplits}}}}
1530
1531 \newcommand*{\dosplits}{\ifnum\@k>0 \noalign{\hfil}\splitoff
1532   \global\advance\@k-1\cr\dosplits\fi}
1533
1534 \newcommand*{\splitoff}{\dimen0=\ht0
1535   \divide\dimen0 by\@k \advance\dimen0 by\@h
1536   \setbox2 \vsplit0 to \dimen0
1537   \unvbox2 }
1538
```

**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.

```
1539 \newcommand*\footthreecol}[1]{%
1540   \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
1541   \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
1542   \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
1543   \threecolfootsetup{#1}
```

The additional setup for minipages.

```
1544   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
1545   \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
1546   \mpthreecolfootsetup{#1}
1547 }
1548
```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (p. 98 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.

```
1549 \newcommand*\threecolfootsetup}[1]{%
1550   \count\csname #1footins\endcsname 333
1551   \multiply\dimen\csname #1footins\endcsname \thr@@}
```

`\mpthreecolfootsetup` The setup for minipages.

```
1552 \newcommand*\mpthreecolfootsetup}[1]{%
1553   \count\csname mp#1footins\endcsname 333
1554   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
1555
```

`\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).

```
1556 \newcommand*{\threecolvfootnote}[2]{%
1557   \insert\csname #1footins\endcsname\bgroup
1558   \notefontsetup
1559   \footsplitskips
1560   \csname #1footfmt\endcsname #2\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.

```
1561 \newcommand*{\threecolfootfmt}[3]{%
1562   \normal@pars
1563   \hsize .3\hsize
1564   \parindent=0pt
1565   \tolerance=5000
1566   \raggedright
1567   \leavevmode
1568   \strut{\notenumfont\printlines#1}\enspace
1569   {\select@lemmafnt#1|#2}\rbracket\enskip
1570   #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.

```
1571 \newcommand*{\threecolfootgroup}[1]{\notefontsetup
1572   \splittopskip=\ht\strutbox
1573   \expandafter
1574   \rigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}}
```

`\mpthreecolfootgroup` The setup for minipages.

```
1575 \newcommand*{\mpthreecolfootgroup}[1]{%
1576   \vskip\skip\@nameuse{mp#1footins}
1577   \normalcolor
1578   \@nameuse{#1footnoterule}
1579   \splittopskip=\ht\strutbox
1580   \expandafter}
```

```
1581 \rigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
1582
```

### 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.

```
1583 \newcommand*{\foottwocol}[1]{%
1584 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
1585 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
1586 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
1587 \twocolfootsetup{#1}
```

The additional setup for minipages.

```
1588 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
1589 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup
1590 \mptwocolfootsetup{#1}
1591 }
1592
```

`\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`.

```
1593 \newcommand*{\twocolfootsetup}[1]{%
1594 \count\csname #1footins\endcsname 500
1595 \multiply\dimen\csname #1footins\endcsname \tw@}

1596 \newcommand*{\twocolvfootnote}[2]{\insert\csname #1footins\endcsname\bgroup
1597 \notefontsetup
1598 \footsplitskips
1599 \csname #1footfmt\endcsname #2\egroup}

1600 \newcommand*{\twocolfootfmt}[3]{%
1601 \normal@pars
1602 \hsize .45\hsize
1603 \parindent=0pt
1604 \tolerance=5000
1605 \raggedright
1606 \leavevmode
1607 \strut{\notenumfont\printlines#1|}\enspace
1608 {\select@lemmafnt#1|#2}\rbracket\enskip
1609 #3\strut\par\allowbreak}

1610 \newcommand*{\twocolfootgroup}[1]{\notefontsetup
1611 \splittopskip=\ht\strutbox
1612 \expandafter
1613 \rigidbalance\csname #1footins\endcsname \tw@ \splittopskip}}
1614
```

```

\mptwocolfootsetup The versions for minipages.
\mptwocolfootgroup 1615 \newcommand*{\mptwocolfootsetup}[1]{%
1616   \count\csname mp#1footins\endcsname 500
1617   \multiply\dimen\csname mp#1footins\endcsname \tw@}
1618 \newcommand*{\mptwocolfootgroup}[1]{%
1619   \vskip\skip\@nameuse{mp#1footins}
1620   \normalcolor
1621   \@nameuse{#1footnoterule}
1622   \splittopskip=\ht\strutbox
1623   \expandafter
1624   \rigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
1625

```

## 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.
1626 \countdef\pageno=0 \pageno=1
1627 \newcommand*{\advancepageno}{\ifnum\pageno<\z@ \global\advance\pageno\m@ne
1628   \else\global\advance\pageno\@ne\fi}
1629

```

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\@cclv \unvbox\@cclv % 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 T<sub>E</sub>X'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 1630 \providecommand{\m@m@makecolfloats}{%
1631   \edef\@freelist{\@freelist\@midlist}%
1632   \global \let \@midlist \@empty
1633   \@combinefloats}
1634 \providecommand{\m@m@makecoltext}{%
1635   \ifvbox\@kludgeins
1636     \@makespecialcolbox
1637   \else
1638     \setbox\@outputbox \vbox to\@colht {%
1639       \@texttop
1640       \dimen@ \dp\@outputbox
1641       \unvbox\@outputbox
1642       \vskip -\dimen@
1643       \@textbottom}%
1644   \fi}
1645 \providecommand{\m@m@makecolintro}{}
1646

```

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

```

1647 \gdef\l@d@makecol{%
1648   \l@ddofootinsert
1649   \m@m@makecolfloats

```

```

1650 \m@m@makecoltext
1651 \global \maxdepth \@maxdepth}
1652

```

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

```

1653 \newcommand*{\l@ddofootinsert}{%
1654 %%% \page@start
1655 \ifvoid\footins
1656 \setbox\@outputbox \box\@cclv
1657 \else
1658 \setbox\@outputbox \vbox {%
1659 \boxmaxdepth \@maxdepth
1660 \@tempdima\dp\@cclv
1661 \unvbox \@cclv
1662 \vskip \skip\footins
1663 \color@begingroup
1664 \normalcolor
1665 \footnoterule
1666 \unvbox \footins
1667 \color@endgroup
1668 }%
1669 \fi

```

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

```

1670 \l@ddoxtrafeet
1671 }
1672

```

`\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).

```

1673 \newcommand*{\l@ddoxtrafeet}{%
1674 \doxtrafeeti
1675 \doxtrafeetii}
1676

```

`\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'.

```

1677 \newcommand*{\doxtrafeetii}{%
1678 \setbox\@outputbox \vbox{%
1679 \unvbox\@outputbox
1680 \@opxtrafeetii}}

```

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

```

1681 \newcommand*{\@opxtrafeetii}{%
1682 \ifvoid\Afootins\else\Afootstart{A}\Afootgroup{A}\fi
1683 \ifvoid\Bfootins\else\Bfootstart{B}\Bfootgroup{B}\fi

```

```

1684 \ifvoid\Cfootins\else\Cfootstart{C}\Cfootgroup{C}\fi
1685 \ifvoid\Dfootins\else\Dfootstart{D}\Dfootgroup{D}\fi
1686 \ifvoid\Efootins\else\Efootstart{E}\Efootgroup{E}\fi
1687

```

`\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`.

```

1688 \newcommand*{\l@ddodoreintrafeet}{%
1689 \doreintrafeeti
1690 \doreintrafeetii}
1691

```

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

```

1692 \newcommand*{\doreintrafeetii}{%
1693 \ifvoid\Afootins\else\insert\Afootins{\unvbox\Afootins}\fi
1694 \ifvoid\Bfootins\else\insert\Bfootins{\unvbox\Bfootins}\fi
1695 \ifvoid\Cfootins\else\insert\Cfootins{\unvbox\Cfootins}\fi
1696 \ifvoid\Dfootins\else\insert\Dfootins{\unvbox\Dfootins}\fi
1697 \ifvoid\Efootins\else\insert\Efootins{\unvbox\Efootins}\fi
1698 }
1699

```

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

```

1700 \gdef \l@d@reinserts{%
1701 \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
1702 \l@ddodoreintrafeet
1703 \ifvbox\@kludgeins\insert\@kludgeins{\unvbox\@kludgeins}\fi
1704 }
1705

```

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.)

```

1706 \@ifclassloaded{memoir}{%
  memoir is loaded so we use memoir’s built in hooks.
1707 \g@addto@macro{\m@mmdoextrafeet}{\l@ddoxtrafeet}%
1708 \g@addto@macro{\m@mmdodoreintrafeet}{\l@ddodoreintrafeet}%
1709 }{%
  memoir has not been loaded, so redefine @makecol and @reinserts.
1710 \gdef\@makecol{\l@d@makecol}%
1711 \gdef\@reinserts{\l@d@reinserts}%
1712 }
1713

```

`\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.

```

1714 \newcommand*{\addfootins}[1]{%
1715   \footnormal{#1}
      Add it to the output.
1716   \g@addto@macro{\@opxtrafeetii}{%
1717     \ifvoid\@nameuse{#1footins}\else
1718       \@nameuse{#1footstart{#1}}\@nameuse{#1footgroup}{#1}\fi}
      Add it to the reinsertions.
1719   \g@addto@macro{\doreinextrafeetii}{%
1720     \ifvoid\@nameuse{#1footins}\else
1721       \insert\@nameuse{#1footins}{\unvbox\@nameuse{#1footins}}\fi}
      Add it to minipages.
1722   \g@addto@macro{\l@dedbeginmini}{%
1723     \expandafter\let\csname #1footnote\endcsname = \@nameuse{mp#1footnote}}
      And at the end of a minipage.
1724   \g@addto@macro{\l@dedendmini}{%
1725     \ifvoid\@nameuse{mp#1footins}\else\@nameuse{mpfootgroup#1{#1}}\fi}
1726 }
1727
```

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.

```

1728 \newif\if@led@nofoot
1729 \newcommand*{\@led@extranofeet}{}
1730
1731 \ifclassloaded{memoir}{%
```

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

`\@mem@extranofeet`

```

1732 \g@addto@macro{\@mem@extranofeet}{%
1733   \ifvoid\Afootins\else\@mem@nofootfalse\fi
1734   \ifvoid\Bfootins\else\@mem@nofootfalse\fi
1735   \ifvoid\Cfootins\else\@mem@nofootfalse\fi
1736   \ifvoid\Dfootins\else\@mem@nofootfalse\fi
1737   \ifvoid\Efootins\else\@mem@nofootfalse\fi
1738   \ifvoid\footinsA\else\@mem@nofootfalse\fi
1739   \ifvoid\footinsB\else\@mem@nofootfalse\fi
1740   \ifvoid\footinsC\else\@mem@nofootfalse\fi
1741   \@led@extranofeet}
1742 }{%
```

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

```

\@led@testifnofoot
\@doclearpage 1743 \newcommand*{\@led@testifnofoot}{%
1744 \@led@nofoottrue
1745 \ifvoid\footins\else\@led@nofootfalse\fi
1746 \ifvoid\Afootins\else\@led@nofootfalse\fi
1747 \ifvoid\Bfootins\else\@led@nofootfalse\fi
1748 \ifvoid\Cfootins\else\@led@nofootfalse\fi
1749 \ifvoid\Dfootins\else\@led@nofootfalse\fi
1750 \ifvoid\Efootins\else\@led@nofootfalse\fi
1751 \ifvoid\footinsA\else\@led@nofootfalse\fi
1752 \ifvoid\footinsB\else\@led@nofootfalse\fi
1753 \ifvoid\footinsC\else\@led@nofootfalse\fi
1754 \@led@extranofeet}
1755
1756 \renewcommand{\@doclearpage}{%
1757 \@led@testifnofoot
1758 \if@led@nofoot
1759 \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
1760 \setbox\@tempboxa\box\@cclv
1761 \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
1762 \global \let \@toplist \@empty
1763 \global \let \@botlist \@empty
1764 \global \@colroom \@colht
1765 \ifx \@currlist\@empty
1766 \else
1767 \latexerr{Float(s) lost}\@ehb
1768 \global \let \@currlist \@empty
1769 \fi
1770 \@makefcolumn\@deferlist
1771 \@whilesw\if@fcolmade \fi{\@opcol\@makefcolumn\@deferlist}%
1772 \if@twocolumn
1773 \if@firstcolumn
1774 \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%
1775 \global \let \@dbltoplist \@empty
1776 \global \@colht \textheight
1777 \beginngroup
1778 \dblfloatplacement
1779 \@makefcolumn\@dbldeferlist
1780 \@whilesw\if@fcolmade \fi{\@outputpage
1781 \makefcolumn\@dbldeferlist}%
1782 \endgroup
1783 \else
1784 \vbox{}\clearpage
1785 \fi
1786 \fi
1787 \else
1788 \setbox\@cclv\vbox{\box\@cclv\vfil}%
1789 \l@makecol\@opcol
1790 \clearpage

```

```

1791 \fi}
1792 }
1793

```

## 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.

```

1794 \list@create{\labelref@list}

```

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

```

1795 %% \newcommand*\zz@@@{000|000|000} % set three counters to zero in one go
1796 \newcommand*\zz@@@{000|000} % set two counters to zero in one go
1797

```

`\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>28</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>29</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.

```

1798 \newcommand*\edlabel}[1]{\@bsphack
1799 \write\linenum@out{\string\@lab}%
1800 \ifx\labelref@list\empty

```

<sup>28</sup>The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

<sup>29</sup>(jdb43@cam.ac.uk) via the ctt thread 'ledmac cross referencing', 25 August 2003.

```

1801   \xdef\label@refs{\zz@@@}%
1802   \else
1803     \gl@p\labelref@list\to\label@refs
1804   \ifvmode
1805     \advance\label@refs
1806   \fi
1807   \fi
1808 %   \edef\next{\write\@aux{\string\l@dmake@labels\label@refs|{#1}}}%
1809 %   \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).

```

1810   \protected@write\@auxout{}%
1811     {\string\l@dmake@labels\space\thepage|\label@refs|{#1}}%
1812   \@esphack}
1813

```

`\advance\label@refs`

```

\labelrefsparseline 1814 %In cases where \cs{edlabel} is the first element in a paragraph, we have a problem with line counts,
\labelrefsparsesubline 1815 %Hence, we need to test \cs{edlabel} if it occurs at the start of a paragraph. To do so, we use \cs{
1816 %\changes{v0.19}{2012/09/08}{Debug \cs{advance\label@refs} to prevent some troubles.}
1817 %We do so using \cs{advance\label@refs} command.
1818 \newcommand{\advance\label@refs}{%
1819   \newcounter{line}%
1820   \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
1821   \stepcounter{line}%
1822   \ifsublines%
1823     \newcounter{subline}%
1824     \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
1825     \stepcounter{subline}{1}%
1826     \def\label@refs{\theline|\thesubline}%
1827   \else%
1828     \def\label@refs{\theline|0}%
1829 \fi%
1830 }
1831 \def\labelrefsparseline#1|#2{#1}
1832 \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).

```

1833 \newcommand*{\l@dmake@labels}{%
1834 \def\l@dmake@labels#1|#2|#3|#4{%
1835   \expandafter\ifx\csname the@label#4\endcsname \relax\else
1836     \led@warn@DuplicateLabel{#4}%

```

```

1837 \fi
1838 \expandafter\gdef\csname the@label#4\endcsname{#1|#2|#3}%
1839 \ignorespaces}
1840

```

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.

```

1841 \AtBeginDocument{%
1842   \def\l@dmake@labels#1|#2|#3|#4{%
1843 }
1844

```

`\@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`.

```

1845 \newcommand*{\@lab}{\xright@appenditem
1846   {\linenumrep{\line@num}}|%
1847   \ifsublines@ \sublinenumrep{\subline@num}\else 0\fi}\to\labelref@list}
1848

```

`\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`.

```

1849 \newcommand*{\edpageref}[1]{\l@dref@undefined{#1}\l@dgetref@num{1}{#1}}
1850 \newcommand*{\xpageref}[1]{\l@dgetref@num{1}{#1}}
1851

```

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

```

\lineref 1852 \newcommand*{\lineref}[1]{\l@dref@undefined{#1}\l@dgetref@num{2}{#1}}
\lineref 1853 \newcommand*{\xlineref}[1]{\l@dgetref@num{2}{#1}}
1854

```

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

```

\sublineref 1855 \newcommand*{\sublineref}[1]{\l@dref@undefined{#1}\l@dgetref@num{3}{#1}}
\sublineref 1856 \newcommand*{\xsublineref}[1]{\l@dgetref@num{3}{#1}}
1857

```



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.

```
1858 \newcommand*\l@dref@undefined}[1]{%
1859   \expandafter\ifx\csname the@label#1\endcsname\relax
1860     \led@warn@RefUndefined{#1}%
1861   \fi}
1862
```

`\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.

```
1863 \newcommand*\l@dgetref@num}[2]{%
1864   \expandafter
1865   \ifx\csname the@label#2\endcsname \relax
1866     000%
1867   \else
1868     \expandafter\expandafter\expandafter
1869     \l@dlabel@parse\csname the@label#2\endcsname|#1%
1870   \fi}
1871
```

`\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`.)

```
1872 \newcommand*\l@dlabel@parse}{%
1873 \def\l@dlabel@parse#1|#2|#3|#4{%
1874   \ifcase #4\relax
1875   \or #1%
1876   \or #2%
1877   \or #3%
1878   \fi}
1879
```

`\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.

```
1880 \newcommand*\xxref}[2]{%
1881   {\expandafter\ifx\csname the@label#1\endcsname
1882     \relax \expandafter\let\csname the@label#1\endcsname\zz@@@fi
1883     \expandafter\ifx\csname the@label#2\endcsname \relax
1884     \expandafter\let\csname the@label#2\endcsname\zz@@@fi
1885     \linenum{\csname the@label#1\endcsname|%
1886       \csname the@label#2\endcsname}}
1887
```

`\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`.

```
1888 \newcommand*\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\endcsname{#2}}
1889
```

(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. 75 and 55), 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 1890 \newwrite\l@d@end
1891 \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.

1892 \newcommand{\l@dend@open}[1]{\global\l@dend@true\immediate\openout\l@d@end=#1\relax}
1893 \newcommand{\l@dend@close}{\global\l@dend@false\immediate\closeout\l@d@end}
1894
```

`\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.

```
1895 \newcommand{\l@dend@stuff}{%
1896   \ifl@dend@relax\else
1897     \l@dend@open{\jobname.end}%
1898   \fi
1899   \immediate\write\l@d@end{\string\l@d@section{\the\section@num}}
1900 }
```

`\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.

```
1901 \newcommand*\Aendnote}[1]{\{\newlinechar='40
1902   \immediate\write\l@d@end{\string\Aend%
1903     {\ifnumberedpar@\l@d@nums\fi}%
1904     {\ifnumberedpar@\@tag\fi}{#1}}\ignorespaces}
1905 \newcommand*\Bendnote}[1]{\{\newlinechar='40
1906   \immediate\write\l@d@end{\string\Bend%
1907     {\ifnumberedpar@\l@d@nums\fi}%
1908     {\ifnumberedpar@\@tag\fi}{#1}}\ignorespaces}
1909 \newcommand*\Cendnote}[1]{\{\newlinechar='40
1910   \immediate\write\l@d@end{\string\Cend%
1911     {\ifnumberedpar@\l@d@nums\fi}%
1912     {\ifnumberedpar@\@tag\fi}{#1}}\ignorespaces}
1913 \newcommand*\Dendnote}[1]{\{\newlinechar='40
1914   \immediate\write\l@d@end{\string\Dend%
1915     {\ifnumberedpar@\l@d@nums\fi}%
1916     {\ifnumberedpar@\@tag\fi}{#1}}\ignorespaces}
1917 \newcommand*\Eendnote}[1]{\{\newlinechar='40
1918   \immediate\write\l@d@end{\string\Eend%
1919     {\ifnumberedpar@\l@d@nums\fi}%
1920     {\ifnumberedpar@\@tag\fi}{#1}}\ignorespaces}
1921 }
```

`\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>30</sup> The `\endprint` here is nearly identical in  
`\@gobblethree` its functioning to `\normalfootfmt`.

`\l@d@section` <sup>30</sup>Christophe Hebeisen ([christophe.hebeisen@a3.epfl.ch](mailto:christophe.hebeisen@a3.epfl.ch)) emailed on 2003/11/05 to say he had found that `\@gobblethree` was also defined in the `amsfonts` package.

The endnote file also contains `\l@d@section` commands, which supply the section numbers from the main text; standard `ledmac` does nothing with this information, but it's there if you want to write custom macros to do something with it.

```

1922 \def\endprint#1#2#3{{\notefontsetup{\notenumfont\printendlines#1}}%
1923     \enspace{\select@lemmafонт#1|#2}\enskip#3\par}}
1924 \providecommand*{\@gobblethree}[3]{}
1925 \let\Aend=\@gobblethree
1926 \let\Bend=\@gobblethree
1927 \let\Cend=\@gobblethree
1928 \let\Dend=\@gobblethree
1929 \let\Eend=\@gobblethree
1930 \let\l@d@section=\@gobble
1931

```

`\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.

```

1932 \newcommand*{\setprintendlines}[6]{%
1933     \l@d@pnumfalse \l@d@dashfalse
1934     \ifnum#4=#1 \else
1935         \l@d@pnumtrue
1936         \l@d@dashtrue
1937     \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.

```

1938     \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
1939     \ifnum#2=#5 \else
1940         \l@d@elintrue
1941         \l@d@dashtrue
1942     \fi

```

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

```

1943     \l@d@ssubfalse
1944     \ifnum#3=0 \else
1945         \l@d@ssubtrue
1946     \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.

```

1947     \l@d@eslfalse
1948     \ifnum#6=0 \else

```

```

1949     \ifnum#6=#3
1950         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
1951     \else
1952         \l@d@esltrue
1953         \l@d@dashtrue
1954     \fi
1955 \fi}

```

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

```

1956 \def\printendlines#1|#2|#3|#4|#5|#6|#7|{\begingroup
1957   \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).

```

1958   \printnpnum{#1} \linenumrep{#2}%
1959   \ifl@d@ssub \fullstop \sublinenumrep{#3}\fi
1960   \ifl@d@dash \endashchar\fi
1961   \ifl@d@pnum \printnpnum{#4}\fi
1962   \ifl@d@elin \linenumrep{#5}\fi
1963   \ifl@d@esl \ifl@d@elin \fullstop\fi \sublinenumrep{#6}\fi
1964 \endgroup}
1965

```

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

```

1966 \newcommand*\printnpnum[1]{p.#1) }
1967

```

`\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.

```

1968 \newcommand*\doendnotes[1]{\l@d@d@close
1969   \begingroup
1970     \makeatletter
1971     \expandafter\let\csname #1end\endcsname=\endprint
1972     \input\jobname.end
1973   \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.

```

1974 \newcommand*\noendnotes{\global\let\l@d@d@stuff=\relax
1975   \global\chardef\l@d@end=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 `\marginpars` in numbered text do  
`\@xympar` not disturb the flow.

```
1976 \let\l@dold@xympar\@xympar
1977 \renewcommand{\@xympar}{%
1978   \ifnumberedpar@
1979     \led@warn@NoMarginpars
1980     \@esphack
1981   \else
1982     \l@dold@xympar
1983   \fi}
1984
```

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).

```
1985 \newcount\sidenote@margin
1986 \newcommand*{\sidenotemargin}[1]{%
1987   \l@dgetsidenote@margin{#1}%
1988   \ifnum\@l@dttempcntb>\m@ne
1989     \global\sidenote@margin=\@l@dttempcntb
1990   \fi}}
1991 \newcommand*{\l@dgetsidenote@margin}[1]{%
1992   \def\@tempa{#1}\def\@tempb{left}%
1993   \ifx\@tempa\@tempb
1994     \@l@dttempcntb \z@
1995   \else
1996     \def\@tempb{right}%
1997     \ifx\@tempa\@tempb
1998       \@l@dttempcntb \@ne
1999     \else
2000       \def\@tempb{outer}%
2001       \ifx\@tempa\@tempb
2002         \@l@dttempcntb \tw@
2003       \else
2004         \def\@tempb{inner}%
2005         \ifx\@tempa\@tempb
2006           \@l@dttempcntb \thr@@
2007         \else
2008           \led@warn@BadSidenotemargin
2009           \@l@dttempcntb \m@ne
2010         \fi
2011       \fi
2012     \fi
2013   \fi}
2014 \sidenotemargin{right}
2015
```

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

`\l@drp@rbox`

```

2016 \newbox\l@dlp@rbox
2017 \newbox\l@drp@rbox
2018

```

`\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 2019 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep 2020 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup 2021 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup 2022 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
2023 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
2024 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
2025

```

`\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 2026 \newcommand*\ledleftnote[1]{\edtext{\l@dlsnote{#1}}}
2027 \newcommand*\ledrightnote[1]{\edtext{\l@drsnote{#1}}}
2028 \newcommand*\ledsidenote[1]{\edtext{\l@dcsnote{#1}}}
2029
2030

```

`\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 2031 \newif\ifrighnoteup
2032 \righnoteuptrue
2033 \newcommand*\l@dlsnote[1]{%
2034 \ifnumberedpar@
2035 \xright@appenditem{\noexpand\l@dlsnote{#1}}%
2036 \to\inserts@list
2037 \global\advance\insert@count \@ne
2038 \fi\ignorespaces}
2039 \newcommand*\l@drsnote[1]{%
2040 \ifnumberedpar@
2041 \xright@appenditem{\noexpand\l@drsnote{#1}}%
2042 \to\inserts@list
2043 \global\advance\insert@count \@ne
2044 \fi\ignorespaces}
2045 \newcommand*\l@dcsnote[1]{%
2046 \ifnumberedpar@
2047 \xright@appenditem{\noexpand\l@dcsnote{#1}}%
2048 \to\inserts@list
2049 \global\advance\insert@count \@ne
2050 \fi\ignorespaces}
2051

```

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

```

\vl@drsnote 2052 \newcommand*\vl@dlsnote[1]{\setl@dlp@rbox{#1}}
\vl@dcsnote 2053 \newcommand*\vl@drsnote[1]{\setl@drp@rbox{#1}}

```

```
2054 \newcommand*{\l@dcsnote}[1]{\gdef\l@dcsnotetext{#1}}
2055
```

`\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.

```
2056 \newcommand*{\setl@dlp@rbox}[1]{%
2057   {\parindent\z@\hsize=\ledl@notewidth\ledl@notefontsetup
2058     \global\setbox\l@dlp@rbox
2059     \ifleftnoteup
2060       =\vbox to\z@{\vss #1}%
2061     \else
2062       =\vbox to 0.70\baselineskip{\strut#1\vss}%
2063     \fi}}
2064 %% \global\setbox\l@dlp@rbox=\vbox to\z@{\#3\vss}}% aligns on top line
2065 \newcommand*{\setl@drp@rbox}[1]{%
2066   {\parindent\z@\hsize=\ledr@notewidth\ledr@notefontsetup
2067     \global\setbox\l@drp@rbox
2068     \ifrightnoteup
2069       =\vbox to\z@{\vss#1}%
2070     \else
2071       =\vbox to0.7\baselineskip{\strut#1\vss}%
2072     \fi}}
2073 \newif\ifleftnoteup
2074 \leftnoteuptrue
```

`\save@dcsnote` Save the moveable note text in `\l@dcsnotetext`.

```
\l@dcsnotetext 2075 \newcommand*{\save@dcsnote}[3]{%
2076   \gdef\l@dcsnotetext{#3}}
2077
```

`\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`.

```
2078 \newcommand*{\affixside@note}{%
2079   \gdef\@templ@d{}%
2080   \ifx\@templ@d\l@dcsnotetext \else
2081     \iftwocolumn
2082       \if@firstcolumn
2083         \setl@dlp@rbox{\l@dcsnotetext}%
2084       \else
2085         \setl@drp@rbox{\l@dcsnotetext}%
2086       \fi
2087     \else
2088       \@l@tempcntb=\sidenote@margin
2089       \ifnum\@l@tempcntb>\@ne
2090         \advance\@l@tempcntb by\page@num
2091       \fi
2092     \ifodd\@l@tempcntb
```



```

2093     \setl@drp@rbox{\l@dcsnotetext}%
2094     \else
2095     \setl@dlp@rbox{\l@dcsnotetext}%
2096     \fi
2097     \fi
2098 \fi}
2099

```

## 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.

```

2100 \providecommand*\multiplefootnotemarker}{3sp}
2101 \providecommand*\multfootsep}{\textsuperscript{\normalfont,}}
2102

```

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

```

2103 \providecommand*\m@mmf@prepare}{%
2104 \kern-\multiplefootnotemarker
2105 \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.

```

2106 \providecommand*\m@mmf@check}{%
2107 \ifdim\lastkern=\multiplefootnotemarker\relax
2108 \edef\x@sf{\the\spacefactor}%
2109 \unkern
2110 \multfootsep
2111 \spacefactor\x@sf\relax
2112 \fi}
2113

```

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

```

2114 \@ifclassloaded{memoir}{}{%

```

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

```

2115 \let\l@dold@footnotetext\@footnotetext
2116 \renewcommand{\@footnotetext}[1]{%
2117 \l@dold@footnotetext{#1}%
2118 \m@mmf@prepare}

```

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

```
2119 \renewcommand*{\@footnotemark}{%
2120   \leavevmode
2121   \ifhmode
2122     \edef\x@sf{\the\spacefactor}%
2123     \m@mmf@check
2124     \nobreak
2125     \fi
2126     \@makefnmark
2127     \m@mmf@prepare
2128     \ifhmode\spacefactor\x@sf\fi
2129     \relax}
```

Finished the modifications for the non-memoir case.

```
2130 }
2131
```

`\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.

```
2132 \let\l@doldold@footnotetext\@footnotetext
2133 \renewcommand{\@footnotetext}[1]{%
2134   \ifnumberedpar@
2135     \edtext{-}{\l@dbfnote{#1}}%
2136   \else
2137     \l@doldold@footnotetext{#1}%
2138   \fi}
```

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

```
2139 \newcommand{\l@dbfnote}[1]{%
2140   \ifnumberedpar@
2141     \xright@appenditem{\noexpand\v1@dbfnote{#1}}{\@thefnmark}}%
2142     \to\inserts@list
2143   \global\advance\insert@count \@ne
2144   \fi\ignorespaces}
2145 \newcommand{\v1@dbfnote}[2]{%
2146   \def\@thefnmark{#2}%
2147   \l@doldold@footnotetext{#1}}
2148
```

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.

```
2149 \newcommand{\footnoteA}[1]{%
2150   \stepcounter{footnoteA}%
2151   \protected@xdef\@thefnmarkA{\thefootnoteA}%
2152   \@footnotemarkA
2153   \vfootnoteA{A}{#1}\m@mmf@prepare}
2154
```

`\footinsA` The insert for the A series.

```
2155 \newinsert\footinsA
```

`\c@footnoteA` The A series counter.

```
\thefootnoteA 2156 \newcounter{footnoteA}
2157   \renewcommand{\thefootnoteA}{\arabic{footnoteA}}
2158
```

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

```
2159 \newcommand*\footfootmarkA{\textsuperscript{\thefootnoteA}}
2160
```

`\mpfootnoteA` The extras for minipages.

```
\mpfootinsA 2161 \newcommand{\mpfootnoteA}[1]{%
2162   \stepcounter{footnoteA}%
2163   \protected@xdef\@thefnmarkA{\thefootnoteA}%
2164   \@footnotemarkA
2165   \mpvfootnoteA{A}{#1}\m@mmf@prepare}
2166 \newinsert\mpfootinsA
2167
```

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 2168 \newcommand*\prebodyfootmark{%
2169   \leavevmode
2170   \ifhmode
2171     \edef\x@sf{\the\spacefactor}%
2172     \m@mmf@check
2173     \nobreak
2174   \fi}
```

```

2175 \newcommand{\postbodyfootmark}{%
2176   \m@mmf@prepare
2177   \ifhmode\spacefactor\@x@sf\fi\relax}
2178

```

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

```

2179 \newcommand*{\normal@footnotemarkX}[1]{%
2180   \prebodyfootmark
2181   \@nameuse{bodyfootmark#1}%
2182   \postbodyfootmark}
2183

```

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

```

2184 \newcommand*{\normalbodyfootmarkX}[1]{%
2185   \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>`.

```

2186 \newcommand*{\normalvfootnoteX}[2]{%
2187   \insert\@nameuse{footins#1}\bgroup
2188     \notefontsetup
2189     \footsplitskips
2190     \spaceskip=\z@skip \xspaceskip=\z@skip
2191     \@nameuse{footfmt#1}{#1}{#2}\egroup}
2192

```

`\mpnormalvfootnoteX` The minipage version.

```

2193 \newcommand*{\mpnormalvfootnoteX}[2]{%
2194   \global\setbox\@nameuse{mpfootins#1}\vbox{%
2195     \unvbox\@nameuse{mpfootins#1}
2196     \notefontsetup
2197     \hsize\columnwidth
2198     \@parboxrestore
2199     \color@begingroup
2200     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
2201

```

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

```

2202 \newcommand*{\normalfootfmtX}[2]{%
2203   \ledsetnormalparstuff
2204   {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2205     #2\strut\par}}
2206

```

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

```

2207 \newcommand*{\normalfootfootmarkX}[1]{%
2208   \textsuperscript{\@nameuse{@thefnmark#1}}}
2209

```

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

```

2210 \newcommand*{\normalfootstartX}[1]{%
2211   \vskip\skip\@nameuse{footins#1}%
2212   \leftskip=\z@
2213   \rightskip=\z@
2214   \@nameuse{footnoterule#1}}
2215

```

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

```

2216 \let\normalfootnoteruleX=\footnoterule
2217

```

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

```

2218 \newcommand*{\normalfootgroupX}[1]{%
2219   \unvbox\@nameuse{footins#1}}
2220

```

`\mpnormalfootgroupX` The minipage version.

```

2221 \newcommand*{\mpnormalfootgroupX}[1]{%
2222   \vskip\skip\@nameuse{mpfootins#1}
2223   \normalcolor
2224   \@nameuse{footnoterule#1}
2225   \unvbox\@nameuse{mpfootins#1}}
2226

```

`\normalbfnoteX`

```

2227 \newcommand{\normalbfnoteX}[2]{%
2228   \ifnumberedpar@
2229     \xright@appenditem{\noexpand\vbfnoteX{#1}{#2}{\@nameuse{thefootnote#1}}}%
2230                       \to\inserts@list
2231     \global\advance\insert@count \@ne
2232     \fi\ignorespaces}
2233

```

`\vbfnoteX`

```

2234 \newcommand{\vbfnoteX}[3]{%
2235   \@namedef{@thefnmark#1}{#3}%
2236   \@nameuse{regvfootnote#1}{#1}{#2}}
2237

```

`\vnumfootnoteX`

```

2238 \newcommand{\vnumfootnoteX}[2]{%
2239   \ifnumberedpar@

```

```

2240 \edtext{}{\normalbfnoteX{#1}{#2}}%
2241 \else
2242 \@nameuse{regvfootnote#1}{#1}{#2}%
2243 \fi}
2244

```

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

```

2245 \newcommand*{\footnormalX}[1]{%
2246 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
2247 \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
2248 \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
2249 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
2250 \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
2251 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
2252 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
2253 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
2254 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
2255 \count\csname footins#1\endcsname=1000
2256 \dimen\csname footins#1\endcsname=\ledfootinsdim
2257 \skip\csname footins#1\endcsname=1.2em \@plus .6em \@minus .6em

```

Aditions for minipages.

```

2258 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
2259 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
2260 \count\csname mpfootins#1\endcsname=1000
2261 % \dimen\csname mpfootins#1\endcsname=0.8\vsizer
2262 \dimen\csname mpfootins#1\endcsname=\ledfootinsdim
2263 \skip\csname mpfootins#1\endcsname=1.2em \@plus .6em \@minus .6em
2264 }
2265

```

### 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>}
2266 \newcommand*{\foottwocolX}[1]{%
2267 \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
2268 \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
2269 \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
2270 \twocolfootsetupX{#1}
2271 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
2272 \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
2273 \mptwocolfootsetupX{#1}}
2274

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX

```

```

2275 \newcommand*{\twocolfootsetupX}[1]{%
2276   \count\csname footins#1\endcsname 500
2277   \multiply\dimen\csname footins#1\endcsname by \tw@}
2278 \newcommand*{\mptwocolfootsetupX}[1]{%
2279   \count\csname mpfootins#1\endcsname 500
2280   \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
2281

\twocolvfootnoteX \twocolvfootnoteX{<series>}
2282 \newcommand*{\twocolvfootnoteX}[2]{%
2283   \insert\csname footins#1\endcsname\bgroup
2284   \notefontsetup
2285   \footsplitskips
2286   \spaceskip=\z@skip \xspaceskip=\z@skip
2287   \@nameuse{footfmt#1}{#1}{#2}\egroup}
2288

\twocolfootfmtX \twocolfootfmtX{<series>}
2289 \newcommand*{\twocolfootfmtX}[2]{%
2290   \normal@pars
2291   \hsize .45\hsize
2292   \parindent=\z@
2293   %% \parfillskip=0pt \@plus 1fil
2294   \tolerance=5000\relax
2295   \raggedright
2296   \leavevmode
2297   {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2298     #2\strut\par}\allowbreak}
2299

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX 2300 \newcommand*{\twocolfootgroupX}[1]{\notefontsetup
2301   \splittopskip=\ht\strutbox
2302   \expandafter
2303   \rigidbalance\csname footins#1\endcsname \tw@ \splittopskip}}
2304 \newcommand*{\mptwocolfootgroupX}[1]{%
2305   \vskip\skip\@nameuse{mpfootins#1}
2306   \normalcolor
2307   \@nameuse{footnoterule#1}
2308   \splittopskip=\ht\strutbox
2309   \expandafter
2310   \rigidbalance\csname mpfootins#1\endcsname \tw@ \splittopskip}}
2311

```

### 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>}
2312 \newcommand*{\footthreecolX}[1]{%
2313 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
2314 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
2315 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
2316 \threecolfootsetupX{#1}
2317 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
2318 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
2319 \mpthreecolfootsetupX{#1}}
2320

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX 2321 \newcommand*{\threecolfootsetupX}[1]{%
2322 \count\csname footins#1\endcsname 333
2323 \multiply\dimen\csname footins#1\endcsname by \thr@@}
2324 \newcommand*{\mpthreecolfootsetupX}[1]{%
2325 \count\csname mpfootins#1\endcsname 333
2326 \multiply\dimen\csname mpfootins#1\endcsname by \thr@@}
2327

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
2328 \newcommand*{\threecolvfootnoteX}[2]{%
2329 \insert\csname footins#1\endcsname\bgroup
2330 \notefontsetup
2331 \footsplitskips
2332 \@nameuse{footfmt#1-#1-#2}\egroup}
2333

\threecolfootfmtX \threecolfootfmtX{<series>}
2334 \newcommand*{\threecolfootfmtX}[2]{%
2335 \normal@pars
2336 \hsize .3\hsize
2337 \parindent=\z@
2338 %%% \parfillskip=0pt \@plus 1fil
2339 \tolerance=5000\relax
2340 \raggedright
2341 \leavevmode
2342 {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2343 #2\strut\par}\allowbreak}
2344

\threecolfootgroupX \threecolfootgroupX{<series>}
\mpthreecolfootgroupX 2345 \newcommand*{\threecolfootgroupX}[1]{\notefontsetup
2346 \splittopskip=\ht\strutbox
2347 \expandafter
2348 \rigidbalance\csname footins#1\endcsname \thr@@ \splittopskip}}
2349 \newcommand*{\mpthreecolfootgroupX}[1]{%
2350 \vskip\skip\@nameuse{mpfootins#1}
2351 \normalcolor

```



```

2352 \@nameuse{footnoterule#1}
2353 \splittopskip=\ht\strutbox
2354 \expandafter
2355 \rigidbalance\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
2356

```

### 27.2.3 Paragraphed footnotes

The following macros set footnotes as one paragraph.

```

\footparagraphX \footparagraphX{<series>}
2357 \newcommand*{\footparagraphX}[1]{%
2358 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
2359 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
2360 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
2361 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
2362 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
2363 \count\csname footins#1\endcsname=1000
2364 \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
2365 \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
2366 \count\csname mpfootins#1\endcsname=1000
2367 \para@footsetupX{#1}}
2368

```

```

\para@footsetupX \para@footsetupX{<series>}
2369 \newcommand*{\para@footsetupX}[1]{\notefontsetup
2370 \dimen0=\baselineskip
2371 \multiply\dimen0 by 1024
2372 \divide\dimen0 by \hsize \multiply\dimen0 by \footfudgefiddle\relax
2373 \expandafter
2374 \xdef\csname footfudgefactor#1\endcsname{%
2375 \expandafter\strip@pt\dimen0 }}
2376

```

```

\parafootstartX \parafootstartX{<series>}
2377 \newcommand*{\parafootstartX}[1]{%
2378 \vskip\skip\@nameuse{footins#1}%
2379 \leftskip=\z@
2380 \rightskip=\z@
2381 \parindent=\z@
2382 \vskip\skip\@nameuse{footins#1}%
2383 \@nameuse{footnoterule#1}}
2384

```

```

\para@vfootnoteX \para@vfootnoteX{<series>}{<text>}
\mppara@vfootnoteX 2385 \newcommand*{\para@vfootnoteX}[2]{%
2386 \insert\csname footins#1\endcsname
2387 \bgroup
2388 \notefontsetup

```

```

2389 \footsplitskips
2390 \setbox0=\vbox{\hsize=\maxdimen
2391 \noindent\@nameuse{footfmt#1}{#1}{#2}}%
2392 \setbox0=\hbox{\unvxh0}%
2393 \dp0=\z@
2394 \ht0=\csname footfudgefactor#1\endcsname\wd0
2395 \box0
2396 \penalty0
2397 \egroup}
2398 \newcommand*\mppara@vfootnoteX}[2]{%
2399 \global\setbox\@nameuse{mpfootins#1}\vbox{%
2400 \unvbox\@nameuse{mpfootins#1}
2401 \notefontsetup
2402 \footsplitskips
2403 \setbox0=\vbox{\hsize=\maxdimen
2404 \noindent\color@begingroup\@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
2405 \setbox0=\hbox{\unvxh0}%
2406 \dp0=\z@
2407 \ht0=\csname footfudgefactor#1\endcsname\wd0
2408 \box0
2409 \penalty0}}
2410

\parafootfmtX \parafootfmtX{<series>}
2411 \newcommand*\parafootfmtX}[2]{%
2412 \insertparafootftmsep
2413 \ledsetnormalparstuff
2414 {\notenumfont\@nameuse{footfootmark#1}\strut%\enspace
2415 #2\penalty-10}}
2416

\para@footgroupX \para@footgroupX{<series>}
\mppara@footgroupX 2417 \newcommand*\para@footgroupX}[1]{%
2418 \unvbox\csname footins#1\endcsname
2419 \makehboxofhboxes
2420 \setbox0=\hbox{\unhbox0 \removehboxes}%
2421 \notefontsetup
2422 \noindent\unhbox0\par}
2423 \newcommand*\mppara@footgroupX}[1]{%
2424 \vskip\skip\@nameuse{mpfootins#1}
2425 \normalcolor
2426 \@nameuse{footnoterule#1}
2427 \unvbox\csname mpfootins#1\endcsname
2428 \makehboxofhboxes
2429 \setbox0=\hbox{\unhbox0 \removehboxes}%
2430 \notefontsetup
2431 \noindent\unhbox0\par}}
2432

```

## 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.

```
2433 \newcommand{\footnoteB}[1]{%
2434   \stepcounter{footnoteB}%
2435   \protected@xdef\thefnmarkB{\thefootnoteB}%
2436   \@footnotemarkB
2437   \vfootnoteB{B}{#1}\m@mmf@prepare}
2438
```

`\c@footnoteB`

```
\thefootnoteB 2439 \newcounter{footnoteB}
2440   \renewcommand{\thefootnoteB}{\arabic{footnoteB}}
2441
```

`\footinsB`

```
2442 \newinsert\footinsB
2443
```

`\mpfootnoteB` The extras for minipages.

```
\mpfootinsB 2444 \newcommand{\mpfootnoteB}[1]{%
2445   \stepcounter{footnoteB}%
2446   \protected@xdef\thefnmarkB{\thefootnoteB}%
2447   \@footnotemarkB
2448   \mpvfootnoteB{B}{#1}\m@mmf@prepare}
2449 \newinsert\mpfootinsB
2450
```

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

```
2451 \newcommand{\footnoteC}[1]{%
2452   \stepcounter{footnoteC}%
2453   \protected@xdef\thefnmarkC{\thefootnoteC}%
2454   \@footnotemarkC
2455   \vfootnoteC{C}{#1}\m@mmf@prepare}
```

`\c@footnoteC`

```
\thefootnoteC 2456 \newcounter{footnoteC}
   \footinsC 2457   \renewcommand{\thefootnoteC}{\arabic{footnoteC}}
2458 \newinsert\footinsC
2459
```

`\mpfootnoteC` The extras for minipages.

```
\mpfootinsC 2460 \newcommand{\mpfootnoteC}[1]{%
2461   \stepcounter{footnoteC}%
2462   \protected@xdef\thefnmarkC{\thefootnoteC}%
2463   \@footnotemarkC
2464   \mpvfootnoteC{C}{#1}\m@mmf@prepare}
2465 \newinsert\mpfootinsC
2466
```

Don't forget to initialise the series.

```
2467 \footnormalX{A}
2468 \footnormalX{B}
2469 \footnormalX{C}
2470
```

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

```
2471 \newcommand*\doxtrafeeti}{%
2472   \setbox\@outputbox \vbox{%
2473     \unvbox\@outputbox
2474     \ifvoid\footinsA\else\footstartA{A}\footgroupA{A}\fi
2475     \ifvoid\footinsB\else\footstartB{B}\footgroupB{B}\fi
2476     \ifvoid\footinsC\else\footstartC{C}\footgroupC{C}\fi
2477   }}
2478
2479 \newcommand\doreintrafeeti}{%
2480   \ifvoid\footinsA\else\insert\footinsA{\unvbox\footinsA}\fi
2481   \ifvoid\footinsB\else\insert\footinsB{\unvbox\footinsB}\fi
2482   \ifvoid\footinsC\else\insert\footinsC{\unvbox\footinsC}\fi
2483 }
2484
```

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

```
2485 \newcommand*\addfootinsX}[1]{%
2486   \footnormalX{#1}%
2487   \g@addto@macro{\doxtrafeeti}{%
2488     \setbox\@outputbox \vbox{%
2489       \unvbox\@outputbox
2490       \ifvoid\@nameuse{footins#1}\else
2491         \@nameuse{footstart#1}{#1}\@nameuse{footgroup#1}{#1}\fi}%
2492   \g@addto@macro{\doreintrafeeti}{%
2493     \ifvoid\@nameuse{footins#1}\else
2494       \insert\@nameuse{footins#1}{\unvbox\@nameuse{footins#1}}\fi}%
2495   \g@addto@macro{\l@dfambeginmini}{%
2496     \expandafter\expandafter\expandafter\let\expandafter\expandafter
2497     \csname footnote#1\endcsname \csname mpfootnote#1\endcsname}%
2498   \g@addto@macro{\l@dfamendmini}{%
2499     \ifvoid\@nameuse{mpfootins#1}\else\@nameuse{mpfootgroup#1}{#1}\fi}%
2500 }
```

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

```
2501 \newcommand*{\l@dfheetbeginmini}{\l@dedbeginmini\l@dfambeginmini}
2502 \newcommand*{\l@dfheetendmini}{\l@dedendmini\l@dfamendmini}
2503
```

`\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.

```
2504 \newcommand*{\l@dedbeginmini}{%
2505   \let\Afootnote=\mpAfootnote%
2506   \let\Bfootnote=\mpBfootnote%
2507   \let\Cfootnote=\mpCfootnote%
2508   \let\Dfootnote=\mpDfootnote%
2509   \let\Efootnote=\mpEfootnote}
2510 \newcommand*{\l@dedendmini}{%
2511   \ifvoid\mpAfootins\else\mpAfootgroup{A}\fi%
2512   \ifvoid\mpBfootins\else\mpBfootgroup{B}\fi%
2513   \ifvoid\mpCfootins\else\mpCfootgroup{C}\fi%
2514   \ifvoid\mpDfootins\else\mpDfootgroup{D}\fi%
2515   \ifvoid\mpEfootins\else\mpEfootgroup{E}\fi}
2516
```

`\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.

```
2517 \newcommand*{\l@dfambeginmini}{%
2518   \let\footnoteA=\mpfootnoteA%
2519   \let\footnoteB=\mpfootnoteB%
2520   \let\footnoteC=\mpfootnoteC}
2521 \newcommand*{\l@dfamendmini}{%
2522   \ifvoid\mpfootinsA\else\mpfootgroupA{A}\fi%
2523   \ifvoid\mpfootinsB\else\mpfootgroupB{B}\fi%
2524   \ifvoid\mpfootinsC\else\mpfootgroupC{C}\fi}
2525
```

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

```
2526 \def\@iiiminipage#1#2[#3]#4{%
2527   \leavevmode
2528   \@pboxswfalse
2529   \setlength\@tempdima{#4}%
2530   \def\@mpargs{#1}#2[#3]#4}%
2531   \setbox\@tempboxa\vbox\bgroup
2532     \color@begingroup
2533     \hsize\@tempdima
2534     \textwidth\hsize \columnwidth\hsize
2535     \@parboxrestore
2536     \def\@mpfn{\mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
2537     \let\@footnotetext\@mpfootnotetext
```

The next line is our addition to the original.

```

2538 \l@dfeetbeginmini% added
2539 \let\@listdepth\@mplistdepth \@mplistdepth\z@
2540 \@minipagerestore
2541 \@setminipage}
2542

```

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

```

2543 \def\endminipage{%
2544 \par
2545 \unskip
2546 \ifvoid\@mpfootins\else
2547 \l@dunboxmpfoot
2548 \fi

```

The next line is our addition to the original.

```

2549 \l@dfeetendmini% added
2550 \@minipagefalse
2551 \color@endgroup
2552 \egroup
2553 \expandafter\@iiiparbox\@mpargs{\unvbox\@tempboxa}}
2554

```

`\l@dunboxmpfoot`

```

2555 \newcommand*\l@dunboxmpfoot}{%
2556 \vskip\skip\@mpfootins
2557 \normalcolor
2558 \footnoterule
2559 \unvbox\@mpfootins}
2560

```

`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.

```

2561 \newenvironment{ledgroup}{%
2562 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
2563 \let\@footnotetext\@mpfootnotetext
2564 \l@dfeetbeginmini%
2565 }{%
2566 \par
2567 \unskip
2568 \ifvoid\@mpfootins\else
2569 \l@dunboxmpfoot
2570 \fi
2571 \l@dfeetendmini%
2572 }
2573

```

`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.

```

2574 \newenvironment{ledgroupsized}[2][1]{%
  Set the various text measures.
2575 \hspace #2\relax
2576 %% \textwidth #2\relax
2577 %% \columnwidth #2\relax
  Initialize fills for centering.
2578 \let\ledllfill\hfil
2579 \let\ledrlfill\hfil
2580 \def\@tempa{#1}\def\@tempb{1}%
  Left adjusted numbered lines
2581 \ifx\@tempa\@tempb
2582 \let\ledllfill\relax
2583 \else
2584 \def\@tempb{r}%
2585 \ifx\@tempa\@tempb
  Right adjusted numbered lines
2586 \let\ledrlfill\relax
2587 \fi
2588 \fi
  Set up the footnoting.
2589 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
2590 \let\@footnotetext\@mpfootnotetext
2591 \l@dfteetbeginmini%
2592 }{%
2593 \par
2594 \unskip
2595 \ifvoid\@mpfootins\else
2596 \l@dunboxmpfoot
2597 \fi
2598 \l@dfteetendmini%
2599 }
2600

```

## 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 2601 \newcommand{\pagelinesep}{-}
2602 \newcommand{\edindexlab}{\&\&}
2603 \newcounter{labidx}
2604 \setcounter{labidx}{0}
2605

```

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

```
2606 \newcommand{\doedindexlabel}{\stepcounter{labidx}%
2607   \edlabel{\edindexlab\thelabidx}}
2608
```

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

```
2609 \newcommand{\thepage}{%
2610   \thepage\pagelinesep\lineref{\edindexlab\thelabidx}}
2611
```

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

```
2612 \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.

```
2613 \g@addto@macro{\makememindexhook}{%
2614   \def\edindex{\@bsphack%
2615     \ifnextchar [{\l@d@index}{\l@d@index[\jobname]}}
2616   \newcommand{\edindex}[2][\jobname]{\@bsphack\@esphack}
```

`\l@d@index` `\l@d@index[file]` is the first stage of `\edindex`, handling the `idx` file. This is virtually a verbatim copy of memoir's `\@index`, the change being calling `\l@d@wrindexm@m` instead of `\@wrindexm@m`.

```
2617 \def\l@d@index[#1]{%
2618   \@ifundefined{#1idxfile}%
2619   {\ifreportnoidxfile
2620     \led@warn@NoIndexFile{#1}%
2621     \fi
2622     \begingroup
2623     \@sanitize
2624     \@nowrindex}%
2625   {\def\@idxfile{#1}%
2626     \doedindexlabel
2627     \begingroup
2628     \@sanitize
2629     \l@d@wrindexm@m}}
```

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

```
2630 \newcommand{\l@d@wrindexm@m}[1]{\l@d@wrindexhyp#1||\}
2631 \def\l@d@wrindexhyp#1|#2|#3\{\%
2632   \ifshowindexmark\@showidx{#1}\fi
2633   \ifx\#2\%
```



```

2634     \protected@write\@auxout{}%
2635         {\string\@wrindexm@m{\@idxfile}#{1|hyperpage}{\thepage}}%
2636     \else
2637         \def\Hy@temp@A{#2}%
2638         \ifx\Hy@temp@A\HyInd@ParenLeft
2639             \protected@write\@auxout{}%
2640                 {\string\@wrindexm@m{\@idxfile}#{1|#2hyperpage}{\thepage}}%
2641         \else
2642             \protected@write\@auxout{}%
2643                 {\string\@wrindexm@m{\@idxfile}#{1|#2}{\thepage}}%
2644         \fi
2645     \fi
2646 \endgroup
2647 \esphack}

```

That finishes the memoir-specific code.

```
2648 }{%
```

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.

```

2649 \g@addto@macro{\makeindex}{%
2650     \def\edindex{\@bsphack
2651         \doedindexlabel
2652         \begingroup
2653         \@sanitize
2654         \wredindex}}
2655 \newcommand{\edindex}[1]{\@bsphack\esphack}

```

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

```

2656 \newcommand{\@wredindex}[1]{%
2657     \protected@write\@indexfile{}%
2658         {\string\indexentry{#1}{\thepage}}%
2659 \endgroup
2660 \esphack}

```

That finishes the non-memoir index code.

```
2661 }
```

```
2662
```

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

```

2663 \AtBeginDocument{\@ifpackageloaded{hyperref}{}{%
2664     \def\l@d@wrindexhyp#1||\{\%
2665         \ifshowindexmark\showidx{#1}\fi
2666         \protected@write\@auxout{}%
2667             {\string\@wrindexm@m{\@idxfile}#{1}{\thepage}}%
2668         \endgroup
2669         \esphack}}}
2670

```

### 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.
2671 \newtoks\@emptytoks
2672

The rest is from amsmath.

\l@denvbody A token register to contain the body.
2673 \newtoks\l@denvbody
2674

\addtol@denvbody \addtol@denvbody{arg} adds arg to the token register \l@denvbody.
2675 \newcommand{\addtol@denvbody}[1]{%
2676   \global\l@denvbody\expandafter{\the\l@denvbody#1}}
2677

\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.
2678 \newcommand{\l@dcollect@body}[1]{%
2679   \l@denvbody{\expandafter#1\expandafter{\the\l@denvbody}}%
2680   \edef\processl@denvbody{\the\l@denvbody\noexpand\end{\@currenvir}}%
2681   \l@denvbody\@emptytoks \def\l@dbegin@stack{b}%
2682   \begingroup
2683     \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
2684     \edef\processl@denvbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
2685     \processl@denvbody}
2686

\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.
2687 \def\l@dpush@begins#1\begin#2{%
2688   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
2689

\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.

```

2690 \def\l@dcollect@body#1\end#2{%
2691   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
2692     \expandafter\@gobble\l@dbegin@stack}%
2693   \ifx\@empty\l@dbegin@stack
2694     \endgroup
2695     \@checkend{#2}%
2696     \addtol@denvbody{#1}%
2697   \else
2698     \addtol@denvbody{#1\end{#2}}%
2699   \fi
2700   \processl@denvbody % A little tricky! Note the grouping
2701 }
2702

```

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

```

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

```

```

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

```

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

```

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

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

```

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

```

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

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

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

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

\makeatother

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

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

```

```

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

\end{document}

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

```

## 31 Verse

This is principally Wayne Sullivan's code and commentary from EDSTANZA [Sul92].

The macro `\hangingsymbol` is used to insert a symbol on each hanging of verses. For example, in french typographie the symbol is '['. We obtain it by the next code:

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

The `\ifinstanza` boolean is used to be sure that we are in a stanza part.

```

\hangingsymbol
\ifinstanza 2703 \newcommand*{\hangingsymbol}{,}
2704 \newif\ifinstanza
2705 \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.

```

```

2706 \newif\ifinserthangingsymbol
2707 \newcommand{\inserthangingsymbol}{%
2708 \ifinserthangingsymbol%
2709 \ifinstanza%
2710 \hfill\hangingsymbol%
2711 \fi%
2712 \fi%
2713 }

```

```

\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.

```

```

2714 \newcommand*{\ampersand}{\char'\&}
2715

```

`\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.

```
2716 \chardef\body=\catcode'\@
2717 \catcode'\@=11
2718 \chardef\next=\catcode'\&
2719 \catcode'\&=\active
2720
```

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.

```
2721 \newcount\stanza@count
2722 \newlength{\stanzaindentbase}
2723 \setlength{\stanzaindentbase}{20pt}
2724
```

`\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.

```
2725 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
2726 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,,|}%
2727     \stanza@count\z@
2728     \def\next{\expandafter\strip@szacnt\@tempa
2729         \ifx\@tempb\empty\let\next\relax\else
2730         \expandafter\mathchardef\csname #1@\number\stanza@count
2731         \@endcsname\@tempb\relax
2732         \advance\stanza@count\@ne\fi\next}%
2733     \next}
2734
```

`\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.

```
2735 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
2736 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
2737
2738 \newcounter{stanzaindentsrepetition}
2739 \newcount\stanza@modulo
2740
```

```

2741 \newcommand*{\managestanza@modulo}[0]{
2742     \advance\stanza@modulo\@ne
2743     \ifnum\stanza@modulo>\value{stanzaindentsrepetition}
2744         \stanza@modulo\@ne
2745     \fi
2746 }

```

`\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.

```

2747 \def\stanza@line{
2748     \ifnum\value{stanzaindentsrepetition}=0
2749         \parindent=\csname sza@\number\stanza@count
2750             @\endcsname\stanzaindentbase
2751     \else
2752         \managestanza@modulo
2753         \parindent=\csname sza@\number\stanza@modulo
2754             @\endcsname\stanzaindentbase
2755     \fi
2756     \pstart\stanza@hang\ignorespaces}
2757 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
2758     \hangindent\expandafter
2759     \noexpand\csname sza@0@\endcsname\stanzaindentbase
2760     \hangafter\@ne}
2761 \def\sza@penalty{\count@\csname szp@\number\stanza@count @\endcsname
2762     \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
2763     \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.

```

2764 \let\startstanzahook\relax
2765 \let\endstanzaextra\relax
2766 \xdef\stanza{\noexpand\instanzatrue\expandafter

```

```

2767 \begingroup\startstanzahook%
2768 \catcode'\&\active\global\stanza@count\@ne\stanza@modulo\@ne
2769 \noexpand\ifnum\expandafter\noexpand
2770 \csname sza@0@\endcsname=z@\let\noexpand\stanza@hang\relax
2771 \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
2772 \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
2773 \expandafter\noexpand\csname szp@0@\endcsname=z@
2774 \let\noexpand\sza@penalty\relax\noexpand\fi \def\noexpand&{%
2775 \noexpand\endlock\noexpand\pend\noexpand\sza@penalty\global
2776 \advance\stanza@count\@ne\noexpand\stanza@line}\def\noexpand
2777 &{\noexpand\endlock\noexpand\pend\endgroup\noexpand\instanzafalse\expandafter\er
2778 \noexpand\stanza@line}
2779

```

`\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`.

```

2780 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
2781 \hskip -#1\llap{#2}\hskip #1\ignorespaces}
2782

```

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.

```

2783 \catcode'\&=\next
2784 \catcode'\@=\body
2785 %% \let\ampersand=\&
2786 \setstanzavalues{szp}{0}
2787

```

## 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..

```

2788 \newcommand*\l@dtabnoexpands}{%
2789   \def\ss{\noexpand\ss}%
2790   \def\"##1{\noexpand\"##1}%
2791   \def\'##1{\noexpand\'##1}%
2792   \def\`##1{\noexpand\`##1}%
2793   \def\^##1{\noexpand\^##1}%
2794   \def\phantom##1{\noexpand\phantom{##1}}%
2795   \def\hphantom##1{\noexpand\hphantom{##1}}%
2796   \def\underbrace##1{\noexpand\underbrace{##1}}%
2797   \def\dots{\noexpand\dots}%
2798   \let\rtab=0%
2799   \let\ctab=0%
2800   \let\ltab=0%
2801   \let\rtabtext=0%
2802   \let\ltabtext=0%
2803   \let\ctabtext=0%
2804   \let\edbeforetab=0%
2805   \let\edaftertab=0%
2806   \let\edatab=0%
2807   \let\edatabell=0%
2808   \let\edatleft=0%
2809   \let\edatright=0%
2810   \let\edvertline=0%
2811   \let\edvertdots=0%
2812   \let\edrowfill=0%
2813 }
2814

```

`\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.

```

2815 \newcount\l@dampcount
2816 \l@dampcount=1\relax
2817 \newcount\l@dcolcount
2818 \l@dcolcount=0\relax
2819

```

```

\hilfsbox Some (temporary) helper items.
\hilfsskip 2820 \newbox\hilfsbox
\Hilfsbox 2821 \newskip\hilfsskip
\hilfscount 2822 \newbox\Hilfsbox
                2823 \newcount\hilfscount
                2824

```

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).

```

2825 \newdimen\dcoli
2826 \newdimen\dcolii
2827 \newdimen\dcoliii
2828 \newdimen\dcoliv
2829 \newdimen\dcolv
2830 \newdimen\dcolvi
2831 \newdimen\dcolvii
2832 \newdimen\dcolviii
2833 \newdimen\dcolix
2834 \newdimen\dcolx
2835 \newdimen\dcolxi
2836 \newdimen\dcolxii
2837 \newdimen\dcolxiii
2838 \newdimen\dcolxiv
2839 \newdimen\dcolxv
2840 \newdimen\dcolxvi
2841 \newdimen\dcolxvii
2842 \newdimen\dcolxviii
2843 \newdimen\dcolxix
2844 \newdimen\dcolxx
2845 \newdimen\dcolxxi
2846 \newdimen\dcolxxii
2847 \newdimen\dcolxxiii
2848 \newdimen\dcolxxiv
2849 \newdimen\dcolxxv
2850 \newdimen\dcolxxvi
2851 \newdimen\dcolxxvii
2852 \newdimen\dcolxxviii
2853 \newdimen\dcolxxix
2854 \newdimen\dcolxxx
2855 \newdimen\dcolerr % added for error handling
2856

```

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

```

2857 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???
2858 \or \dcoli \or \dcolii \or \dcoliii
2859 \or \dcoliv \or \dcolv \or \dcolvi
2860 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx

```

```

2861 \or \dcolxi \or \dcolxii \or \dcolxiii
2862 \or \dcolxiv \or \dcolxv \or \dcolxvi
2863 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
2864 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
2865 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
2866 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
2867 \else \dcolerr \fi}
2868

```

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

```

2869 \newcommand*{\step1@dcolcount}{\advance\l@dcolcount\@ne
2870 \ifnum\l@dcolcount>30\relax
2871 \led@err@TooManyColumns
2872 \fi}
2873

```

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

```

2874 \newcommand{\l@dsetmaxcolwidth}{%
2875 \ifdim\l@dcolwidth < \wd\hilfsbox
2876 \l@dcolwidth = \wd\hilfsbox
2877 \else \relax \fi}
2878

```

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

```

\CRITEXT 2879 \let\EDTEXT=\edtext
\xcritext 2880 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
2881 \let\CRITEXT=\critext
2882 \long\def\xcritext #1#2/{\CRITEXT{#1}{#2}/}

```

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

```

\xedlabel 2883 \let\EDLABEL=\edlabel
2884 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}

```

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

```

\xedindex 2885 \let\EDINDEX=\edindex
\nulledindex 2886 \ifl@dmemoir
2887 \newcommand{\xedindex}{\@bsphack%
2888 \@ifnextchar [{\l@d@index}{\l@d@index[\jobname]}}
2889 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
2890 \else
2891 \newcommand{\xedindex}{\@bsphack%
2892 \doedindexlabel
2893 \begingroup
2894 \@sanitize
2895 \wredindex}
2896 \newcommand{\nulledindex}[1]{\@bsphack\@esphack}
2897 \fi
2898

```

```

\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 2899 \let\A@@footnote=\Afootnote
\D@@footnote 2900 \let\B@@footnote=\Bfootnote
\E@@footnote 2901 \let\C@@footnote=\Cfootnote
                2902 \let\D@@footnote=\Dfootnote
                2903 \let\E@@footnote=\Efootnote

\@line@num Macro supporting restoration of \linenum.
                2904 \let\@line@num=\linenum

\l@gobbledarg \l@gobbledarg replaces its delineated argument by \relax (was \verschwinden).
\l@gobblearg \l@gobblearg{<arg>} replaces its argument by \relax.
                2905 \def\l@gobbledarg #1/{\relax}
                2906 \newcommand*\l@gobblearg}[1]{\relax}
                2907

\Relax
\NEXT 2908 \let\Relax=\relax
\@hilfs@count 2909 \let\NEXT=\next
                2910 \newcount\@hilfs@count
                2911

\measuremcell Measure (recursively) the width required for a math cell. (was \messen)
                2912 \def\measuremcell #1&{%
                2913     \ifx #1\ \ifnum\l@dc@lcount=0\let\NEXT\relax%
                2914         \else\l@dcheckcols%
                2915             \l@dc@lcount=0%
                2916             \let\NEXT\measuremcell%
                2917             \fi%
                2918     \else\setbox\hilfsbox=\hbox{${\displaystyle#1}$}%
                2919         \step\l@dc@lcount%
                2920         \l@dsetmaxcolwidth%
                2921         \let\NEXT\measuremcell%
                2922         \fi\NEXT}
                2923

\measuretcell Measure (recursively) the width required for a text cell. (was \messentext)
                2924 \def\measuretcell #1&{%
                2925     \ifx #1\ \ifnum\l@dc@lcount=0\let\NEXT\relax%
                2926         \else\l@dcheckcols%
                2927             \l@dc@lcount=0%
                2928             \let\NEXT\measuretcell%
                2929             \fi%
                2930     \else\setbox\hilfsbox=\hbox{#1}%
                2931         \step\l@dc@lcount%
                2932         \l@dsetmaxcolwidth%
                2933         \let\NEXT\measuretcell%
                2934         \fi\NEXT}
                2935

```

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

```
2936 \def\measuremrow #1\{\%
2937   \ifx #1&\let\NEXT\relax%
2938   \else\measuremcell #1&\&\&\&%
2939     \let\NEXT\measuremrow%
2940   \fi\NEXT}
```

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

```
2941 \def\measuretrow #1\{\%
2942   \ifx #1&\let\NEXT\relax%
2943   \else\measuretcell #1&\&\&\&%
2944     \let\NEXT\measuretrow%
2945   \fi\NEXT}
2946
```

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

```
2947 \newskip\edtabcolsep
2948 \global\edtabcolsep=10pt
2949
```

`\NEXT`

```
\Next 2950 \let\NEXT\relax
2951 \let\Next=\next
```

`\variab`

```
2952 \newcommand{\variab}{\relax}
2953
```

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

```
2954 \newcommand*{\l@dcheckcols}{\%
2955   \ifnum\l@dcolcount=1\relax
2956   \else
2957     \ifnum\l@dampcount=1\relax
2958     \else
2959       \ifnum\l@dcolcount=\l@dampcount\relax
2960       \else
2961         \l@d@err@UnequalColumns
2962       \fi
2963     \fi
2964     \l@dampcount=\l@dcolcount
2965   \fi}
2966
```

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

```
\l@drestoreforcritext 2967 \newcommand{\l@dmodforcritext}{\%
2968   \let\critext\relax%
2969   \let\Afootnote\l@dgobbledarg%
2970   \let\Bfootnote\l@dgobbledarg%
2971   \let\Cfootnote\l@dgobbledarg%
```

```

2972 \let\Dfootnote\l@dgobbledarg%
2973 \let\Efootnote\l@dgobbledarg%
2974 \let\edindex\nulledindex%
2975 \let\linenum@gobble}
2976 \newcommand{\l@drestoreforcritext}{%
2977 \def\Afootnote##1##2/{\A@footnote{##1}{##2}}%
2978 \def\Bfootnote##1##2/{\B@footnote{##1}{##2}}%
2979 \def\Cfootnote##1##2/{\C@footnote{##1}{##2}}%
2980 \def\Dfootnote##1##2/{\D@footnote{##1}{##2}}%
2981 \def\Efootnote##1##2/{\E@footnote{##1}{##2}}%
2982 \let\edindex\xedndex}
2983

```

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

```

\l@drestoreforedtext 2984 \newcommand{\l@dmodforedtext}{%
2985 \let\edtext\relax
2986 \let\Afootnote\l@dgobblearg
2987 \let\Bfootnote\l@dgobblearg
2988 \let\Cfootnote\l@dgobblearg
2989 \let\Dfootnote\l@dgobblearg
2990 \let\Efootnote\l@dgobblearg
2991 \let\edindex\nulledindex
2992 \let\linenum@gobble}
2993 \newcommand{\l@drestoreforedtext}{%
2994 \def\Afootnote##1{\A@footnote{##1}}%
2995 \def\Bfootnote##1{\B@footnote{##1}}%
2996 \def\Cfootnote##1{\C@footnote{##1}}%
2997 \def\Dfootnote##1{\D@footnote{##1}}%
2998 \def\Efootnote##1{\E@footnote{##1}}%
2999 \let\edindex\xedndex}
3000

```

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

```

\l@drestorefills 3001 \newcommand{\l@dnullfills}{%
3002 \def\edlabel##1{}}%
3003 \def\edrowfill##1##2##3{}}%
3004 }
3005 \newcommand{\l@drestorefills}{%
3006 \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
3007 }
3008

```

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.

```

3009 \newcommand{\letsforverteilen}{%
3010   \let\critext\xcritext
3011   \let\edtext\xedtext
3012   \let\edindex\xedindex
3013   \let\Afootnote\A@@footnote
3014   \let\Bfootnote\B@@footnote
3015   \let\Cfootnote\C@@footnote
3016   \let\Dfootnote\D@@footnote
3017   \let\Efootnote\E@@footnote
3018   \let\linenum\@line@num
3019   \hilfsskip=\l@dcolwidth%
3020   \advance\hilfsskip by -\wd\hilfsbox

```

```
3021 \def\edlabel##1{\xedlabel{##1}}
3022
```

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

```
3023 \def\setmcellright #1&{\def\edlabel##1{}}%
3024         \let\edindex\nulledindex
3025     \ifx #1\\ \ifnum\l@dcolcount=0\removelastskip
3026         \let\Next\relax%
3027     \else\l@dcolcount=0%
3028         \let\Next=\setmcellright%
3029     \fi%
3030 \else%
3031     \disablel@dtabfeet%
3032     \step1@dcolcount%
3033     \setbox\hilfsbox=\hbox{\displaystyle{#1}}%
3034     \letsforverteilen%
3035     \hskip\hilfsskip\displaystyle{#1}%
3036     \hskip\edtabcolsep%
3037     \let\Next=\setmcellright%
3038 \fi\Next}
3039
```

`\settcclright` Typeset (recursively) cells of text right justified. (was `\rverteilentext`)

```
3040 \def\settcclright #1&{\def\edlabel##1{}}%
3041         \let\edindex\nulledindex
3042     \ifx #1\\ \ifnum\l@dcolcount=0\removelastskip
3043         \let\Next\relax%
3044     \else\l@dcolcount=0%
3045         \let\Next=\settcclright%
3046     \fi%
3047 \else%
3048     \disablel@dtabfeet%
3049     \step1@dcolcount%
3050     \setbox\hilfsbox=\hbox{#1}%
3051     \letsforverteilen%
3052     \hskip\hilfsskip#1%
3053     \hskip\edtabcolsep%
3054     \let\Next=\settcclright%
3055 \fi\Next}
```

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

```
3056 \def\setmcellleft #1&{\def\edlabel##1{}}%
3057         \let\edindex\nulledindex
3058     \ifx #1\\ \ifnum\l@dcolcount=0 \let\Next\relax%
3059         \else\l@dcolcount=0%
3060         \let\Next=\setmcellleft%
3061     \fi%
3062 \else \disablel@dtabfeet%
3063     \step1@dcolcount%
```



```

3064         \setbox\hilfsbox=\hbox{\displaystyle{#1}}%
3065         \letsforverteilen
3066         $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
3067         \let\Next=\setmcellleft%
3068     \fi\Next}
3069

```

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

```

3070 \def\settcclleft #1{\def\edlabel##1{%
3071         \let\edindex\nulledindex
3072     \ifx #1\ \ifnum\l@dc@colcount=0 \let\Next\relax%
3073         \else\l@dc@colcount=0%
3074         \let\Next=\settcclleft%
3075         \fi%
3076     \else \disablel@dtabfeet%
3077         \stepl@dc@colcount%
3078         \setbox\hilfsbox=\hbox{#1}%
3079         \letsforverteilen
3080         #1\hskip\hilfsskip\hskip\edtabcolsep%
3081         \let\Next=\settcclleft%
3082     \fi\Next}

```

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

```

3083 \def\setmcellcenter #1{\def\edlabel##1{%
3084         \let\edindex\nulledindex
3085     \ifx #1\ \ifnum\l@dc@colcount=0\let\Next\relax%
3086         \else\l@dc@colcount=0%
3087         \let\Next=\setmcellcenter%
3088         \fi%
3089     \else \disablel@dtabfeet%
3090         \stepl@dc@colcount%
3091         \setbox\hilfsbox=\hbox{\displaystyle{#1}}%
3092         \letsforverteilen%
3093         \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
3094         \hskip\edtabcolsep%
3095         \let\Next=\setmcellcenter%
3096     \fi\Next}
3097

```

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

```

3098 \def\settcclcenter #1{\def\edlabel##1{%
3099         \let\edindex\nulledindex
3100     \ifx #1\ \ifnum\l@dc@colcount=0 \let\Next\relax%
3101         \else\l@dc@colcount=0%
3102         \let\Next=\settcclcenter%
3103         \fi%
3104     \else \disablel@dtabfeet%
3105         \stepl@dc@colcount%
3106         \setbox\hilfsbox=\hbox{#1}%

```

```

3107         \letsforverteilen%
3108         \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
3109         \hskip\edtabcolsep%
3110         \let\Next=\settcellcenter%
3111     \fi\Next}
3112

```

`\NEXT`

```

3113 \let\NEXT=\relax
3114

```

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

```

3115 \def\setmrowright #1\\{%
3116     \ifx #1& \let\NEXT\relax
3117     \else \centerline{\setmcellright #1&\\&\\&}
3118         \let\NEXT=\setmrowright
3119     \fi\NEXT}

```

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

```

3120 \def\settroright #1\\{%
3121     \ifx #1& \let\NEXT\relax
3122     \else \centerline{\settcellright #1&\\&\\&}
3123         \let\NEXT=\settroright
3124     \fi\NEXT}
3125

```

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

```

3126 \def\setmrowleft #1\\{%
3127     \ifx #1& \let\NEXT\relax
3128     \else \centerline{\setmcellleft #1&\\&\\&}
3129         \let\NEXT=\setmrowleft
3130     \fi\NEXT}

```

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

```

3131 \def\settrorleft #1\\{%
3132     \ifx #1& \let\NEXT\relax
3133     \else \centerline{\settcellleft #1&\\&\\&}
3134         \let\NEXT=\settrorleft
3135     \fi\NEXT}
3136

```

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

```

3137 \def\setmrowcenter #1\\{%
3138     \ifx #1& \let\NEXT\relax%
3139     \else \centerline{\setmcellcenter #1&\\&\\&}
3140         \let\NEXT=\setmrowcenter
3141     \fi\NEXT}

```

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

```

3142 \def\settrowcenter #1\{%
3143   \ifx #1& \let\NEXT\relax
3144   \else \centerline{\settcellcenter #1&\&\&}
3145         \let\NEXT=\settrowcenter
3146   \fi\NEXT}
3147

```

`\nullsetzen` (was `\nullsetzen`)

```

3148 \newcommand{\nullsetzen}{%
3149   \stepl@dcolcount%
3150   \l@dcolwidth=0pt%
3151   \ifnum\l@dcolcount=30\let\NEXT\relax%
3152         \l@dcolcount=0\relax
3153   \else\let\NEXT\nullsetzen%
3154   \fi\NEXT}
3155

```

`\edatleft` `\edatleft[ $\langle symbol \rangle$ ]{ $\langle len \rangle$ }` (combination and generalisation of original `\Seklam` and `\Seklamgl`). Left  $\langle symbol \rangle$ ,  $2\langle len \rangle$  high with prepended  $\langle math \rangle$  vertically centered.

```

3156 \newcommand{\edatleft}[3][\@empty]{%
3157   \ifx#1\@empty
3158     \vbox to 10pt{\vss\hbox{\left#2\vrule width0pt height #3
3159                               depth 0pt \right. $\hss}\vfil}
3160   \else
3161     \vbox to 4pt{\vss\hbox{\#1\left#2\vrule width0pt height #3
3162                               depth 0pt \right. $\}\vfil}
3163   \fi}

```

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

```

3164 \newcommand{\edatright}[3][\@empty]{%
3165   \ifx#1\@empty
3166     \vbox to 10pt{\vss\hbox{\left.\vrule width0pt height #3
3167                               depth 0pt \right#2 $\hss}\vfil}
3168   \else
3169     \vbox to 4pt{\vss\hbox{\left.\vrule width0pt height #3
3170                               depth 0pt \right#2 #1 $\}\vfil}
3171   \fi}
3172

```

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

```

3173 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
3174

```

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

```

3175 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
3176         {\cleaders\hbox{${\m@th\hbox{.}}\vbox to 0.5em{ }$\}\vfil}}}
3177

```

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`)

```

3178 \newdimen\edfilldimen
3179 \edfilldimen=0pt
3180

```

`\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...`

```

3181 \newcounter{addcolcount}
3182 \renewcommand{\theadcolcount}{\roman{addcolcount}}

```

`\l@dtabaddcols` `\l@dtabaddcols{<startcol>}{<endcol>}` adds the widths of the columns `<startcol>`  
through `<endcol>` to `\edfilldimen`. It is a LaTeX style reimplementaion of the  
original `\@add@`.

```

3183 \newcommand{\l@dtabaddcols}[2]{%

```

```

3184 \l@dcheckstartend{#1}{#2}%
3185 \ifl@dstartendok
3186 \setcounter{addcolcount}{#1}%
3187 \@whilenum \value{addcolcount}<#2\relax \do
3188 {\advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
3189 \advance\edfilldimen by \edtabcolsep
3190 \stepcounter{addcolcount}}%
3191 \advance\edfilldimen by \the \csname dcol\theadcolcount\endcsname
3192 \fi
3193 }
3194

```

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

```

3195 \newif\ifl@dstartendok
3196 \newcommand{\l@dcheckstartend}[2]{%
3197 \l@dstartendoktrue
3198 \ifnum #1<\@ne
3199 \l@dstartendokfalse
3200 \led@err@LowStartColumn
3201 \fi
3202 \ifnum #2>30\relax
3203 \l@dstartendokfalse
3204 \led@err@HighEndColumn
3205 \fi
3206 \ifnum #1>#2\relax
3207 \l@dstartendokfalse
3208 \led@err@ReverseColumns
3209 %%% \ledmac@error{Start column is greater than end column}{\@ehc}%
3210 \fi
3211 }
3212

```

`\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.

```

3213 \newcommand*{\edrowfill}[3]{%
3214 \l@dtabaddcols{#1}{#2}%
3215 \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
3216 \let\@edrowfill=\edrowfill
3217 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
3218

```

`\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)
3219 \newcommand{\leftltab}[1]{%
3220 \hb@xt@z@{\vbox{\edtabindent%
3221 \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
3222

\leftrtab \leftrtab{<text>}{<math>} for \edbeforetab in \rtab. (was \linksrtab)
3223 \newcommand{\leftrtab}[2]{%
3224 #2\hb@xt@z@{\vbox{\edtabindent%
3225 \advance\Hilfsskip by\dcoli%
3226 \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
3227

\leftctab \leftctab{<text>}{<math>} for \edbeforetab in \ctab. (was \linksztab)
3228 \newcommand{\leftctab}[2]{%
3229 \hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
3230 \advance\Hilfsskip by 0.5\dcoli%
3231 \setbox\hilfsbox=\hbox{\def\edlabel##1}{%
3232 \disablel@dtabfeet$\displaystyle{#2}$}%
3233 \advance\Hilfsskip by -0.5\wd\hilfsbox%
3234 \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
3235 #2}
3236

\rightctab \rightctab{<math>}{<text>} for \edaftertab in \ctab. (was \rechtsztab)
3237 \newcommand{\rightctab}[2]{%
3238 \setbox\hilfsbox=\hbox{\def\edlabel##1}{%
3239 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
3240 #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
3241 \advance\Hilfsskip by 0.5\l@dcolwidth%
3242 \advance\Hilfsskip by -\wd\hilfsbox%
3243 \setbox\hilfsbox=\hbox{\def\edlabel##1}{%
3244 \disablel@dtabfeet$\displaystyle{#1}$}%
3245 \advance\Hilfsskip by -0.5\wd\hilfsbox%
3246 \advance\Hilfsskip by \edtabcolsep%
3247 \moveright\Hilfsskip\hbox{ #2}}\hss}%
3248 }
3249

\rightltab \rightltab{<math>}{<text>} for \edaftertab in \ltab. (was \rechtsltab)
3250 \newcommand{\rightltab}[2]{%
3251 \setbox\hilfsbox=\hbox{\def\edlabel##1}{%
3252 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
3253 #1\hb@xt@z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
3254 \advance\Hilfsskip by\l@dcolwidth%
3255 \advance\Hilfsskip by-\wd\hilfsbox%
3256 \setbox\hilfsbox=\hbox{\def\edlabel##1}{%
3257 \disablel@dtabfeet$\displaystyle{#1}$}%
3258 \advance\Hilfsskip by-\wd\hilfsbox%

```

```

3259     \advance\Hilfsskip by\edtabcolsep%
3260     \moveright\Hilfsskip\hbox{ #2}\hss}%
3261     }
3262

```

`\righttab` `\righttab{ $$ }` for `\edaftertab` in `\rtab`. (was `\rechtsrtab`)

```

3263 \newcommand{\righttab}[2]{%
3264     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
3265     \disablel@dtabfeet#2}%
3266     #1\hb@xt@z@{\vbox{\edtabindent%
3267     \advance\Hilfsskip by-\wd\hilfsbox%
3268     \advance\Hilfsskip by\edtabcolsep%
3269     \moveright\Hilfsskip\hbox{ #2}\hss}%
3270     }
3271

```

`\rtab` `\rtab{body}` typesets *body* as an array with the entries right justified. (was `\edbeforetab` `\rtab`) (Here and elsewhere, `\edbeforetab` and `\edaftertab` were originally `\davor` and `\danach`) The original `\rtab` and friends included a fair bit of common code which I have extracted into macros.

The process is first to measure the *body* to get the column widths, and then in a second pass to typeset the body.

```

3272 \newcommand{\rtab}[1]{%
3273     \l@dnnullfills
3274     \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
3275     \def\edaftertab##1##2{\righttab{##1}{##2}}%
3276     \measuretbody{#1}%
3277     \l@drestorefills
3278     \variab
3279     \setmrowright #1\&\%
3280     \enablel@dtabfeet}
3281

```

`\measuretbody` `\measuretbody{body}` measures the array *body*.

```

3282 \newcommand{\measuretbody}[1]{%
3283     \disablel@dtabfeet%
3284     \l@dcolcount=0%
3285     \nullsetzen%
3286     \l@dcolcount=0
3287     \measuremrow #1\&\%
3288     \global\l@dampcount=1}
3289

```

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

```

3290 \newcommand{\rtabtext}[1]{%
3291     \l@dnnullfills
3292     \measuretbody{#1}%
3293     \l@drestorefills

```

```

3294 \variab
3295 \settroright #1\\&\\%
3296 \enablel@dtabfeet}
3297

```

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

```

3298 \newcommand{\measuretbody}[1]{%
3299 \disablel@dtabfeet%
3300 \l@dc colcount=0%
3301 \nullsetzen%
3302 \l@dc colcount=0
3303 \measuretbody #1\\&\\%
3304 \global\l@dampcount=1}
3305

```

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

```

\edbeforetab 3306 \newcommand{\ltab}[1]{%
\edaftertab 3307 \l@dn nullfills
3308 \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
3309 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
3310 \measuretbody{#1}%
3311 \l@drestorefills
3312 \variab
3313 \setmrowleft #1\\&\\%
3314 \enablel@dtabfeet}
3315

```

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

```

3316 \newcommand{\ltabtext}[1]{%
3317 \l@dn nullfills
3318 \measuretbody{#1}%
3319 \l@drestorefills
3320 \variab
3321 \setmrowleft #1\\&\\%
3322 \enablel@dtabfeet}
3323

```

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

```

\edbeforetab 3324 \newcommand{\ctab}[1]{%
\edaftertab 3325 \l@dn nullfills
3326 \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
3327 \def\edaftertab##1##2{\rightctab{##1}{##2}}%
3328 \measuretbody{#1}%
3329 \l@drestorefills
3330 \variab
3331 \setmrowcenter #1\\&\\%
3332 \enablel@dtabfeet}
3333

```



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

```
3334 \newcommand{\ctabtext}[1]{%
3335   \l@dnnullfills
3336   \measuretbody{#1}%
3337   \l@drestorefills
3338   \variab
3339   \settrrowcenter #1\&\%
3340   \enablel@dtabfeet}
3341
```

`\spreadtext` (was `\breitertext`)

```
3342 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
3343   \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
```

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

```
3344 \newcommand{\spreadmath}[1]{%
3345   \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
3346
```

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`)

```
3347 \def\tabellzwischen #1&{%
3348   \ifx #1\ \let\NEXT\relax \l@dcolcount=0
3349   \else \step1@dcolcount%
3350         \l@dcolwidth = #1 mm
3351         \let\NEXT=\tabellzwischen
3352   \fi \NEXT }
3353
```

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

```
3354 \def\edatabell #1\&{%
3355   \tabellzwischen #1&\&}
```

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

```
3356 \def\Setzen #1&{%
3357   \ifx #1\relax \let\NEXT=\relax
3358   \else \step1@dcolcount%
3359         \let\tabelskip=\l@dcolwidth
3360         \EDTAB #1|
3361         \let\NEXT=\Setzen
3362   \fi \NEXT}
3363
```

```

\EDATAB (was \ATAB)
3364 \def\EDATAB #1\{%
3365   \ifx #1\Relax \centerline{\Setzen #1\relax&}
3366           \let\Next\relax
3367   \else \centerline{\Setzen #1&\relax&}
3368           \let\Next=\EDATAB
3369   \fi\Next}

\edatab (was \atab)
3370 \newcommand{\edatab}[1]{%
3371   \variab%
3372   \EDATAB #1\\Relax\\}
3373

\HILFSskip More helpers.
\Hilfsskip 3374 \newskip\HILFSskip
3375 \newskip\Hilfsskip
3376

\EDTABINDEnt (was \TABINDEnt)
3377 \newcommand{\EDTABINDEnt}{%
3378   \ifnum\l@dc@lcount=30\let\NEXT\relax\l@dc@lcount=0%
3379   \else\step\l@dc@lcount%
3380     \advance\Hilfsskip by\l@dc@lwidth%
3381     \ifdim\l@dc@lwidth=0pt\advance\hilfsc@unt\@ne
3382     \else\advance\Hilfsskip by \the\hilfsc@unt\edtabcolsep%
3383     \hilfsc@unt=1\fi%
3384     \let\NEXT=\EDTABINDEnt%
3385   \fi\NEXT}%

\edtabindent (was \tabindent)
3386 \newcommand{\edtabindent}{%
3387   \l@dc@lcount=0\relax
3388   \Hilfsskip=0pt%
3389   \hilfsc@unt=1\relax
3390   \EDTABINDEnt%
3391   \hilfsskip=\hsiz%
3392   \advance\hilfsskip -\Hilfsskip%
3393   \Hilfsskip=0.5\hilfsskip%
3394   }%
3395

\EDTAB (was \TAB)
3396 \def\EDTAB #1|#2|{%
3397   \setbox\tabhilfbox=\hbox{\$displaystyle{#1}$}%
3398   \setbox\tabHilfbox=\hbox{\$displaystyle{#2}$}%
3399   \advance\tabelskip -\wd\tabhilfbox%
3400   \advance\tabelskip -\wd\tabHilfbox%
3401   \unhbox\tabhilfbox\hskip\tabelskip%

```

```

3402 \unhbox\tabHilfbox}%
3403

```

\EDTABtext (was \TABtext)

```

3404 \def\EDTABtext #1|#2|{%
3405 \setbox\tabhilfbox=\hbox{#1}%
3406 \setbox\tabHilfbox=\hbox{#2}%
3407 \advance\tabelskip -\wd\tabhilfbox%
3408 \advance\tabelskip -\wd\tabHilfbox%
3409 \unhbox\tabhilfbox\hskip\tabelskip%
3410 \unhbox\tabHilfbox}%

```

\tabhilfbox Further helpers.

```

\tabHilfbox 3411 \newbox\tabhilfbox
3412 \newbox\tabHilfbox
3413

```

```

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

```

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

```

edarrayc 3414 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{}
edarrayr 3415 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{}
3416 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{}
3417

```

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

```

edtabularc 3418 \newenvironment{edtabularl}{\l@dcollect@body\ltabtext}{}
edtabularr 3419 \newenvironment{edtabularc}{\l@dcollect@body\ctabtext}{}
3420 \newenvironment{edtabularr}{\l@dcollect@body\rtabtext}{}
3421

```

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 3422 \newcommand{\usingcritext}{%
\usingedtext 3423 \def\disablel@dtabfeet{\l@dmodforcritext}%
3424 \def\enablel@dtabfeet{\l@drestoreforcritext}}
3425 \newcommand{\usingedtext}{%
3426 \def\disablel@dtabfeet{\l@dmodforedtext}%
3427 \def\enablel@dtabfeet{\l@drestoreforedtext}}
3428
3429 \usingedtext
3430

```

### **33 The End**

i/code<sub>i</sub>

## A Examples

This section presents some sample documents.

The examples in sections A.2 through A.5, plus A.7, were originally written for TeX. I have done some limited conversions of these so that they look more like LaTeX code. In particular wherever possible I have replaced `\def` commands by either `\newcommand` or `\renewcommand` as appropriate. I have also replaced the original TeX font handling commands by the LaTeX font commands.

The other examples were written natively in LaTeX.

The figures are from processed versions of the files. Having latexed a file I used DVIPS to get Encapsulated PostScript, then the `epstopdf` script to get a PDF version as well, for example:

```
> latex ledeasy
> latex ledeasy
> latex ledeasy
> dvips -E -o ledeasy.eps ledeasy
> epstopdf ledeasy.eps % produces ledeasy.pdf
```

For those who aren't fascinated by LaTeX code, I show the all the typeset results first, then the code that produced them.

# Simple Example

Peter Wilson\*

## Contents

<b>1</b>	<b>First</b>	<b>1</b>
1.1	Example text . . . . .	1
<b>2</b>	<b>Last</b>	<b>1</b>

## 1 First

This is a simple example of using the `ledmac` package with ordinary LaTeX constructs.

### 1.1 Example text

- 1 The `ledmac` package lets you do some unusual things in a LaTeX document.
- 2 For example you can have lines numbered and there are several levels of foot-
- 3 notes. You can label lines within the numbered text and refer to them outside.
- 4 Do not try and use any normal LaTeX `marginpars`<sup>1</sup> or `exotica` within the num-
- 5 bered portions of the text.

Sidenotes  
are OK

## 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.

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.

---

<b>1</b> example:: eximemple C, D.	<b>6</b> <i>examples</i> :: eximples L, M.
<b>1</b> variant:: alternative, A, B.	<b>6</b> variant:: alternative, A, B.
<b>2</b> though:: however $\alpha, \beta$	

---

<b>2</b> ‘C’] B, <i>pace</i> the text	<b>9</b> shouldn’t] ought not to	<b>10</b> useful] very, very useful
<b>7</b> though] however $\alpha, \beta$	L, M	L, P
<b>7</b> ‘B’] B, as correctly	<b>10</b> forget the] omit to	<b>10</b> many] lots of Z
stated in the text	mention the §, ¶	<b>11</b> recorded] recorded and
<b>9</b> Finally] In the end X,	<b>10</b> column] blocked M, N	put down: M
Y	<b>10</b> notes] variants H	(repetition)
<b>9</b> we] we here K		

---

**2–4** For spice ... note: The note here is type ‘C’  
**8–9** For spice, ... note: This is a rogue note of type ‘C’.

---

**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.

---

**4** Finally: in the end X, Y    **4** we: us K    **4** shouldn’t: ought not to L, M    **4** forget  
 the: omit to mention the §, ¶    **4** paragraphed: blocked M, N    **4** notes: variants HH, KK  
**5** useful: truly useful L, P    **5** a great number of: many, many (preferably)    **5** recorded:  
 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 toung,  
 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 toung vnfoldes without affectes  
 „Then may procede 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, wherin 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.] *This swelling*  
*hart puft vp with wicked ire / Can scarce pronounce one inward louing thought.* MS 31  
*Medusaes*] *One of the furies.* 75m



[SCENE III.—*Venice.*]*Enter* JESSICA *and* [LAUNCELOT] *the clown.*

- Jes.* I am sorry thou wilt leave my father so,  
 Our house is hell, and thou (a merry devil)  
 Didst rob it of some taste of tediousness,—  
 But fare thee well, there is a ducat for thee,  
 And Launcelot, soon at supper shalt thou see 5  
 Lorenzo, who is thy new master's guest,  
 Give him this letter,—do it secretly,—  
 And so farewell: I would not have my father  
 See me in talk with thee.
- Laun.* Adieu! tears exhibit my tongue, most beautiful pagan, most sweet 10  
 Jew!—if a Christian do not play the knave and get thee, I am much  
 deceived; but adieu! these foolish drops do something drown my  
 manly spirit: adieu! [*Exit.*]
- Jes.* Farewell good Launcelot.  
 Alack, what heinous sin is it in me 15  
 To be ashamed to be my father's child!

Scene III] *Capell*; *om.* *Q, F*; Scene IV *Pope*. *Venice*] *om.* *Q, F*; *Shylock's house* *Theobald*;  
*The same. A Room in Shylock's House* *Capell*. Launcelot] *Rowe*; *om.* *Q, F*. 1. I am] *Q*,  
*F*; I'm *Pope*. 9. in] *Q*; *om.* *F*. 10. *Laun.*] *Q2*; *Clowne*. *Q, F*. 10. Adieu!] Adieu, *Q, F*.  
 11. Jew!] *Iewe, Q, F*. do] *Q, F*; did *F2*. 12. adieu!] adieu, *Q, F*. 12. something] *Q*;  
 somewhat *F*. 13. adieu!] adieu. *Q, F*. S. D.] *Q2, F*; *om.* *Q*; after l. 15 *Capell*. 16. child!]  
 child, *Q, F*; Child? *Rowe*.

5. *soon*] early.

10. *exhibit*] *Eccles* paraphrased "My tears  
 serve to express what my tongue should, if  
 sorrow would permit it," but probably it is  
 Launcelot's blunder for prohibit (*Halliwell*)  
 or inhibit (*Clarendon*).

10. *pagan*] This may have a scurrilous un-  
 dertone: cf. *2 H 4*, II. ii. 168.

11. *do*] *Malone* upheld the reading of *Qq*

and *F* by comparing II. vi. 23: "When you  
 shall please to play the thieves for wives";  
 Launcelot seems fond of hinting at what is  
 going to happen (cf. II. v. 22–3). If *F2*'s "did"  
 is accepted, *get* is used for *beget*, as in III. v.  
 9.

12–13. *foolish. . . spirit*] "tears do not be-  
 come a man" (*AYL.*, III. iv. 3); cf. also *H 5*,  
 IV. vi. 28–32.

Figure 4: Output from `ledarden.tex`.

ΑΝΑΚΕΦΑΛΙΟΣΙΣ NATVRARVM	Incipit Quartus ΠΕΡΙΦΥΣΕΩΝ	741C
	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 cuiusque 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>r</sup>  sicut ipsa reuelant: ΦΩΣ, ΠΥΡ, ΠΝΕΥΜΑ (hoc est lux, ignis, spiritus).</i>	5  10 742C  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 erudiat. 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 743A  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 ἀνακεφαλίωσις* 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-  
 5 tate, ut ecclesia eius parochialis extra civitatem sita destrueretur et infra muros  
 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] Coloniensi 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 Dhúnlaing  
 [Fonn: Máirseáil Uí Shúilleabháin (Páinseach na nUbh)]

- 1** A dhuine gan chéill do mhaisligh an chléir  
 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 chaithfidimid cruinn bheith ar mhaoleann  
 Josepha?  
 g Ní caraid Mac Crae chuim t'anama ' phlé  
 h ná Calvin bhiais 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 scoilaire 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 dorcha,  
 g tuigsint an léinn, gach cuirpeacht déin  
 h is Lucifer aosta 'na mháistir.

**22** *Teideal*: Dhuinnluing 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 phleidh T 1.h bhíos B  
 1.h leis B 2.a claon B 2.c glannuig T 2.d ainibhfios T, ainibhfios B  
 2.d Éabha B 2.g is B 2.h tuigíonn T 3.a sgollaire T 3.a mholluim T  
 3.b 'tág ccobar T 3.b re B 3.c dorbhchon daor B 3.d straodhaig T  
 3.e fhadoghthe tsíos T 3.e fadaighthe B 3.f sollus T 3.g cuirripeacht T  
 3.h Luicifer T, Lúcifer B 3.h mhaighistir T

Figure 7: Output from `ledbraonain.tex`.

## A.1 Simple example

This made-up example, `ledeasy.tex`, is included to show how simple it can be to use EDMAC in a LaTeX document. The code is given below and the result is shown in Figure 1.

---

```

3431 (*easy)
3432 % ledeasy.tex simple example of the ledmac package
3433 \documentclass{article}
3434 \usepackage{ledmac}
3435 %% number every line
3436 \setcounter{firstlinenum}{1}
3437 \setcounter{linenumincrement}{1}
3438 %% Show some B series familiar footnotes, lettered and paragraphed
3439 \renewcommand*{\thefootnoteB}{\alph{footnoteB}}
3440 \footparagraphX{B}
3441 %% no endnotes
3442 \noendnotes
3443 %% narrow sidenotes
3444 \setlength{\ledrsnotewidth}{4em}
3445 \title{Simple Example}
3446 \author{Peter Wilson\thanks{Standing on the shoulders of giants.}}
3447 \date{}
3448 \begin{document}
3449 \maketitle
3450 \tableofcontents
3451 \section{First}
3452   This is a simple example of using the \textsf{ledmac}
3453 package with ordinary LaTeX constructs.
3454
3455 \subsection{Example text}\label{subsec}
3456
3457 \beginnumbering
3458 \pstart
3459 The \textsf{ledmac} package lets you do some unusual things in
3460 a LaTeX document. For example you can have lines numbered and
3461 there are
3462 \edtext{several}{\Afootnote{This is an 'A' footnote.}}
3463 \edtext{levels}{\Bfootnote{This is a 'B' level footnote.}}
3464 of footnotes.
3465 You can label lines within the numbered text and refer to them
3466 outside. Do not try and use any normal LaTeX
3467 \marginpars\footnote{You will get a warning but no text.}%
3468 \ledrightnote{Sidenotes are OK}
3469 or \edtext{exotica}{\Afootnote{Like floats.}}
3470 within the numbered portions of the text\edlabel{line}.
3471 \pend
3472 \endnumbering
3473
3474 \section{Last}

```

```

3475
3476   I forgot to mention that you can use ordinary
3477 footnotes\footnote{An ordinary footnote}\footnote{And another}
3478 outside the numbered text. You can also\footnoteB{Additionally}
3479 have\footnoteB{Specify} formatted footnotes\footnoteB{Like this}
3480 in normal\footnoteB{Text that does not have line numbers} text.
3481
3482   There are \lineref{line} numbered lines in the example shown
3483 in section~\ref{subsec}.
3484
3485 \end{document}
3486 </easy>

```

## A.2 General example of features

This made-up example, `ledfeat.tex`, is included purely to illustrate some of `ledmac`'s main features. It is hard to find real-world examples that actually use as many layers of notes as this, so we made one up. The example is a bit tricky to read, but close study and comparison with the output (Figure 2) will be illuminating.

I have converted the original TeX code to look more like LaTeX code.

---

```

3487 (*features)
3488 % ledfeat.tex Small test file for ledmac package
3489 \documentclass{article}
3490 \usepackage{ledmac}
3491
3492 \noendnotes % we aren't having any endnotes
3493
3494 \makeatletter
3495 % I'd like a spaced out colon after the lemma:
3496 \newcommand{\spacedcolon}{\rmfamily\thinspace:\thinspace}
3497 \renewcommand*{\normalfootfmt}[3]{%
3498   \ledsetnormalparstuff
3499   {\notenumfont\printlines#1|}\strut\enspace
3500   {\select@lemmafnt#1/#2}\spacedcolon\enskip#3\strut\par}
3501
3502 % And I'd like the 3-col notes printed with a hanging indent:
3503 \renewcommand*{\threecolfootfmt}[3]{%
3504   \normal@pars
3505   \hsize .3\hsize
3506   \setlength{\parindent}{0pt}
3507   \tolerance=5000 % high, but not infinite
3508   \raggedright
3509   \hangindent1.5em \hangafter1
3510   \leavevmode
3511   \strut\hbox to 1.5em{\notenumfont\printlines#1|\hfil}\ignorespaces
3512   {\select@lemmafnt#1/#2}\rbracket\enskip
3513   #3\strut\par\allowbreak}
3514

```

```

3515 % And I'd like the 2-col notes printed with a double colon:
3516 \newcommand*{\doublecolon}{\rmfamily\thinspace::\thinspace}
3517 \renewcommand*{\twocolfootfmt}[3]{%
3518   \normal@pars
3519   \hsize .45\hsize
3520   \setlength{\parindent}{0pt}
3521   \tolerance=5000
3522   \raggedright
3523   \leavevmode
3524   \strut{\notenumfont\printlines#1|}\enspace
3525   {\select@lemmafnt#1|#2}\doublecolon\enskip
3526   #3\strut\par\allowbreak}
3527
3528 % And in the paragraphed footnotes, I'd like a colon too:
3529 \renewcommand*{\parafootfmt}[3]{%
3530   \ledsetnormalparstuff
3531   {\notenumfont\printlines#1|}\enspace
3532   {\select@lemmafnt#1|#2}\spacedcolon\enskip
3533   #3\penalty-10 }
3534 \makeatother
3535
3536 % I'd like the line numbers picked out in bold.
3537 \renewcommand{\notenumfont}{\bfseries}
3538 \lineation{page}
3539 \linenummargin{inner}
3540 \setcounter{firstlinenum}{3}      % just because I can
3541 \setcounter{linenumincrement}{1}
3542 \foottwocol{A}
3543 \footthreecol{B}
3544 \footparagraph{E}
3545 % I've changed \normalfootfmt, so invoke it again for C and D notes.
3546 \footnormal{C}
3547 \footnormal{D}
3548
3549 \begin{document}
3550
3551 \beginnumbering
3552
3553 \pstart
3554 This is an \edtext{example}{
3555   \Afootnote{eximemple C, D.}}
3556 of some %\footnote{A normal footnote}
3557 text with \edtext{variant}{
3558   \Afootnote{alternative, A, B.}}
3559 readings recorded as 'A' footnotes. From here on, \edtext{though}{
3560   \Afootnote{however $\alpha$, $\beta$}},
3561 we shall have \edtext{'C'}{
3562   \Bfootnote{B, \textit{pace} the text}}.
3563 \edtext{For spice, let us mark a longer passage, but give a different
3564 lemma for it, so that we don't get a \edtext{huge}{

```

```

3565     \Dfootnote{vast E, F; note that this is
3566     a 'D' note to section of text within a longer lemma}}
3567     amount of text in a note}{\lemma{For spice \dots\ note}
3568     \Cfootnote{The note here is type 'C'}}.
3569 \edtext{Finally}{
3570     \Efootnote{in the end X, Y}},
3571 \edtext{we}{
3572     \Efootnote{us K}}
3573 \edtext{shouldn't}{
3574     \Efootnote{ought not to L, M}}
3575 \edtext{forget the}{
3576     \Efootnote{omit to mention the \S, \P}}
3577 \edtext{paragraphed}{
3578     \Efootnote{blocked M, N}}
3579 \edtext{notes}{
3580     \Efootnote{variants HH, KK}},
3581 which are so \edtext{useful}{
3582     \Efootnote{truly useful L, P}}
3583 when there are \edtext{a great number of}{
3584     \Efootnote{many, many (preferably)}}
3585 short notes to be \edtext{recorded}{
3586     \Efootnote{noted: repetition}}.
3587 \pend
3588
3589 \pstart
3590 This is a second paragraph, giving more \textit{\edtext{examples}}{
3591     \Afootnote{eximples L, M.}}}
3592 of text with \edtext{variant}{
3593     \Afootnote{alternative, A, B.}}
3594 readings recorded as 'A' footnotes. From here on, \edtext{though}{
3595     \Bfootnote{however $\alpha$, $\beta$}},
3596 we shall have \edtext{'B'}{
3597     \Bfootnote{B, as correctly stated in the text}} notes in the text.
3598 \edtext{For spice, let us mark a longer passage, but give a different
3599 lemma for it, so that we don't get a \textit{\edtext{huge}}{
3600     \Dfootnote{vast E, F; note that this is
3601     a 'D' note to text within a longer lemma.}}}
3602 amount of text in a note}{\lemma{For spice, \dots\ note}
3603     \Cfootnote{This is a rogue note of type 'C'}}.
3604 \edtext{Finally}{
3605     \Bfootnote{In the end X, Y}},
3606 \edtext{we}{
3607     \Bfootnote{we here K}}
3608 \edtext{shouldn't}{
3609     \Bfootnote{ought not to L, M}}
3610 \edtext{forget the}{
3611     \Bfootnote{omit to mention the \S, \P}}
3612 \edtext{column}{
3613     \Bfootnote{blocked M, N}}
3614 \edtext{notes}{

```



```

3615 \Bfootnote{variants H}},
3616 which are so \edtext{useful}{
3617 \Bfootnote{very, very useful L, P}}
3618 when there are \edtext{many}{
3619 \Bfootnote{lots of Z}}
3620 short notes to be \edtext{recorded}{
3621 \Bfootnote{recorded and put down: M (repetition)}}.
3622 \pend
3623
3624 \endnumbering
3625 \end{document}
3626 </features>

```

---

### A.3 Gascoigne

The first real-life example is taken from an edition of George Gascoigne's *A Hundreth Sundrie Flowres* that is being prepared by G. W. Pigman III, at the California Institute of Technology. Figure 3 shows the result of setting the text with ledmac.

I have LaTeXified the original code, and removed all the code related to the main document layout, relying on the standard LaTeX layout parameters..

---

```

3627 (*ioc)
3628 %% ledioc.tex
3629 \documentclass{article}
3630 \usepackage{ledmac}
3631
3632 \noendnotes
3633 \makeatletter
3634
3635 \newcommand{\os}{\scriptsize}
3636 \setcounter{firstsublinenum}{1000}
3637 \frenchspacing \setlength{\parskip}{0pt} \hyphenpenalty=1000
3638
3639 % Say \nolinenums if you want no line numbers in the notes.
3640 \newif\ifnolinenums
3641 \newcommand{\nolinenums}{\global\nolinenumstrue}
3642 \newcommand{\linenums}{\global\nolinenumsfalse}
3643
3644 \renewcommand{\rightlinenum}{\ifbypage@\ifnum\line@num<10\kern.5em\fi\else
3645 \ifnum\line@num<10\kern1em\else\ifnum\line@num<100
3646 \kern.5em\fi\fi\fi\kern.5em\numlabfont\the\line@num
3647 \ifnum\subline@num>0:\the\subline@num\fi}
3648
3649 \renewcommand{\leftlinenum}{\numlabfont\the\line@num
3650 \ifnum\subline@num>0:\the\subline@num\fi \kern.5em}
3651 \linenummargin{outer}
3652 \lineation{page}

```

```

3653
3654 \newcommand{\ggfootfmt}[3]{%
3655   \notefontsetup
3656   \let\par=\endgraf
3657   \rightskip=0pt \leftskip=0pt
3658   \setlength{\parindent}{0pt} \parfillskip=0pt plus 1fil
3659   \ifnolinums\relax\else
3660     \begingroup \os \printlines#1|\endgroup
3661     \enskip
3662   \fi
3663   {\rmfamily #2\def\tempa{#2}\ifx\tempa\empty
3664     \else]\enskip\fi#3\penalty-10 }}
3665
3666 % Now reset the \Afootnote parameters and macros:
3667 \footparagraph{A}
3668 \let\Afootfmt=\ggfootfmt
3669 \dimen\Afootins=\vsize
3670 \skip\Afootins=3pt plus9pt
3671 \newcommand*{\ggfootstart}[1]{\vskip\skip\Afootins}
3672 \let\Afootstart=\ggfootstart
3673
3674 \newcommand*{\stage}[1]{\pstart\startsub\parindent=0pt
3675   \hangindent=3em\hangafter=0
3676   {\itshape #1}\let\par=\finishstage}
3677 \newcommand{\finishstage}{\pend\endsub}
3678 \newcommand{\sen}{\leavevmode\lower1ex\hbox{\textrm{'}}}%
3679 \newcommand{\senspeak}[1]{\pstart\obeylines\setbox0=\hbox{\textrm{'}}}%
3680   \leavevmode
3681   \lower1ex\copy0\kern-\wd0\hskip1em{\textit{#1}}%
3682   \hbox to1ex{}\ignorespaces}
3683 \newcommand*{\speak}[1]{\pstart\obeylines\hskip1em{\textit{#1}}%
3684   \hbox to1ex{}\ignorespaces}
3685 \def\nospeaker{\parindent=0em\pstart\let\par=\pend}
3686 \newcommand*{\nospeak}{\pstart\obeylines}
3687 \makeatother
3688
3689 \begin{document}
3690
3691 \setlength{\parindent}{0pt}
3692
3693 \beginnumbering
3694
3695 \stage{Oedipus \edtext{entreth}{\Afootnote{\textit{intrat} MS}}.}
3696
3697 \nospeak
3698 Or that with wrong the right and doubtlesse heire,
3699 Shoulde banisht be out of his princely seate.
3700 Yet thou O queene, so fyle thy sugred toung,
3701 And with suche counsell decke thy mothers tale,
3702 That peace may bothe the brothers heartes inflame,

```

```

3703 And rancour yelde, that erst possess the same.
3704 \pend
3705
3706 \speak[Eteocl.] Mother, beholde, youre hestes for to obey,
3707 In person nowe am I resorted hither:
3708 In haste therefore, fayne woulde I knowe what cause
3709 With hastie speede, so moued hath your mynde
3710 To call me nowe so causelesse out of tyme,
3711 When common wealth moste craues my onely ayde:
3712 Fayne woulde I knowe, what queynt commoditie
3713 Persuades you thus to take a truce for tyme,
3714 And yelde the gates wide open to my foe,
3715 The gates that myght our stately state defende,
3716 And nowe are made the path of our decay.
3717 \pend
3718
3719 \senspeak[Ioca.] Represse deare son, those raging stormes of wrath,
3720 \sen That so bedimme the eyes of thine intente,
3721 \edtext{\sen As when \edtext{the}{\Afootnote{thie MS}} tongue %
3722 (a redy Instrument)
3723 \sen Would \edtext{fayne pronounce}{\Afootnote{faynest tell MS}} %
3724 the meaning of \edtext{the minde}{\Afootnote{thy minde MS}},
3725 \sen \edtext{It}{\lemma{It \dots\ worde.}\Afootnote{Thie %
3726 swelling hart puft vp with wicked ire / Can scarce pronounce %
3727 one inward louing thought. MS}} cannot speake one honest %
3728 seemely worde.}{\lemma{As \dots\ worde.}\Afootnote{\textit{not %
3729 in} \os73}}
3730 \sen But when disdayne is shrunke, or sette asyde,
3731 \sen And mynde of man with leysure can discourse
3732 \sen What seemely woordes his tale may best beseeme,
3733 \sen And that the tounge vnfoldes without affectes
3734 \sen Then may proceede an answere sage and graue,
3735 \sen And euery sentence sawst with sobernesse:
3736 Wherefore vnbende thyne angrie browes deare chylde,
3737 And caste thy rolling eyes none other waye,
3738 That here doost not \edtext{\textit{Medusaes}}{\%
3739 \Afootnote{One of the furies. {\os75}m}} face beholde,
3740 But him, euen him, thy blood and brother deare.
3741 And thou beholde, my \textit{Polinices} eke,
3742 Thy brothers face, wherin when thou mayst see
3743 Thine owne image, remember therwithall,
3744 That what offence thou woldst to him were done,
3745 \pend
3746 \endnumbering
3747
3748 \end{document}
3749
3750 (/ioc)

```

---

## A.4 Shakespeare

The following text illustrates another input file of moderate complexity, with two layers of annotation in use. The example is taken from the Arden *Merchant of Venice*.

I have roughly converted the original TeX file to a LaTeX file. The file is below and the result of LaTeXing it is shown in Figure 4.

---

```

3751 (*arden)
3752 %% ledarden.tex
3753 \documentclass{article}
3754 \usepackage{ledmac}
3755
3756 \makeatletter
3757 \newcommand{\stage}[1]{\rlap{\hbox to \the\linenumsep{%
3758     \hfil\llap{[\textit{#1}]}}}
3759
3760 \newcommand{\speaker}[1]{\pstart\hangindent2em\hangafter1
3761     \leavevmode\textit{#1}\enspace\ignorespaces}
3762
3763 \newcommand{\exit}[1]{\hfill\stage{#1}}
3764
3765 % LEDMAC customizations:
3766 \noendnotes
3767 \setlength{\parindent}{0pt}
3768 \setlength{\linenumsep}{.4in}
3769 \rightskip\linenumsep
3770
3771 \renewcommand{\interparanote glue}{1em plus.5em minus.1em}
3772
3773 \newcommand{\scf}{\tiny}
3774 \let\Afootnoterule=\relax \let\Bfootnoterule=\relax
3775
3776 \renewcommand{\rightlinenum}{\numlabfont\llap{\the\line@num}}
3777 \frenchspacing
3778
3779 % Footnote formats:
3780 % \nonumparafootfmt is a footnote format without line numbers.
3781 \newcommand{\nonumparafootfmt}[3]{%
3782     \ledsetnormalparstuff
3783     \rightskip=0pt
3784     \select@lemmafонт#1/#2\rbracket\enskip
3785     \itshape #3\penalty-10 }
3786
3787 \newcommand{\newparafootfmt}[3]{%
3788     \ledsetnormalparstuff
3789     {\notenumfont\printlines#1/}\fullstop\enspace
3790     {\select@lemmafонт#1/#2\rbracket\enskip
3791     \itshape #3\penalty-10 }
3792

```

```

3793 \newcommand{\newtwocolfootfmt}[3]{%
3794   \normal@pars
3795   \hsize .48\hsize
3796   \tolerance=5000
3797   \rightskip=0pt \leftskip=0pt \parindent=5pt
3798   \strut\notenumfont\printlines#1/\fullstop\enspace
3799   \itshape #2/>\rbracket\penalty100\hskip .5em plus .5em
3800   \normalfont #3\strut\goodbreak}
3801
3802 % Footnote style selections etc. (done last):
3803 \footparagraph{A}
3804 \foottwocol{B}
3805 \let\Afootfmt=\newparafootfmt
3806 \let\Bfootfmt=\newtwocolfootfmt
3807 \let\collation=\Afootnote
3808 \let\note=\Bfootnote
3809 \lineation{section}
3810 \linenummargin{right}
3811 \makeatother
3812
3813 %%%%%%%%%%%
3814
3815 \begin{document}
3816 \pagestyle{empty}
3817
3818 % Initially, we don't want line numbers.
3819 \let\Afootfmt=\nonumparafootfmt
3820
3821 \beginnumbering
3822 \pstart
3823 \centerline{[\edtext{SCENE III}{
3824   \lemma{Scene III}
3825   \collation{Capell; om. Q, F; \textnormal{Scene IV} Pope.}].---%
3826   \edtext{\textit{Venice}}{
3827     \collation{om. Q, F; Shylock's house Theobald; The same.
3828     A Room in Shylock's House Capell.}].]}
3829 \pend
3830 \bigskip
3831
3832 \pstart
3833 \centerline{\textit{Enter} JESSICA \textit{and}
3834   [\edtext{LAUNCELOT}{
3835     \lemma{Launcelot}
3836     \collation{Rowe; om. Q, F.}}] \textit{the clown.}} \pend \bigskip
3837
3838 \let\Afootfmt=\newparafootfmt % we do want line numbers from now
3839
3840 \setline{0}%
3841
3842 \speaker{Jes.}\edtext{I am}{

```

```

3843 \collation{Q, F; \textnormal{I'm} Pope.}}
3844         sorry thou wilt leave my father so,\\
3845 Our house is hell, and thou (a merry devil)\\
3846 Didst rob it of some taste of tediousness,---\\
3847 But fare thee well, there is a ducat for thee,\\
3848 And Launcelot, \edtext{soon}{
3849   \note{early.}}
3850         at supper shalt thou see\\
3851 Lorenzo, who is thy new master's guest,\\
3852 Give him this letter,---do it secretly,---\\
3853 And so farewell: I would not have my father\\
3854 See me \edtext{in}{
3855   \collation{Q; om. F.}}
3856         talk with thee.
3857 \pend
3858
3859 \speaker{Laun.}
3860   \edtext{}{\lemma{\textit{Laun.}}\collation{Q2; Clowne. Q, F.}}%
3861 \edtext{Adieu!}{
3862   \collation{\textnormal{Adiew}, Q, F.}}
3863 tears \edtext{exhibit}{
3864   \note{Eccles paraphrased "My tears serve to express what my
3865     tongue should, if sorrow would permit it," but probably it is
3866     Launce-\lot's blunder for prohibit (Halliwell) or inhibit
3867     (Clarendon).}}
3868 my tongue, most beautiful \edtext{pagan}{
3869   \note{This may have a scurrilous undertone: cf. \textit{2 H 4,}
3870     {\scf II.} \textrm{iii. 168.}}}%
3871 , most sweet \edtext{Jew!}{
3872   \collation{\textnormal{Iewe}, Q, F. \quad \textnormal{do}} Q, F;
3873     \textnormal{did} F2.}}%
3874 ---if a Christian \edtext{do}{
3875   \note{Malone upheld the reading of Qq and F by comparing {\scf II.}
3876     vi. 23: "When you shall please to play the thieves for
3877     wives"; Launcelot seems fond of hinting at what is going to
3878     happen (cf. {\scf II.} v. 22--3). If F2's "did" is accepted,
3879     \textit{get} is used for beget, as in {\scf III.} v. 9.}}
3880 not play the knave and get thee, I am much deceived; but \edtext{adieu!}{
3881   \collation{\textnormal{adiew}, Q, F.}}
3882 these \edtext{foolish drops do \edtext{something}{
3883   \collation{Q; \textnormal{somewhat} F.}}
3884 drown my manly spirit}{
3885   \lemma{foolish\textnormal{\dots}spirit}
3886   \note{"tears do not become a man" (\textit{AYL.}, {\scf III.}
3887     iv. 3); cf. also \textit{H 5,} {\scf IV.} vi. 28--32.}}%
3888 : \edtext{adieu!}{
3889   \collation{\textnormal{adiew}. Q, F. \quad \textnormal{S. D.}} Q2, F; om. Q;
3890     after l. 15 Capell.}}
3891 \exit{Exit.}
3892 \pend

```

```

3893
3894 \speaker{Jes.}
3895 Farewell good Launcelot.\
3896 Alack, what heinous sin is it in me\
3897 To be ashamed to be my father's \edtext{child!}{
3898 \collation{\textnormal{child}, Q, F; \textnormal{Child?} Rowe.}}
3899 \pend
3900 \endnumbering
3901
3902 \end{document}
3903
3904 \</arden>

```

---

## A.5 Classical text edition

The next example, which was extracted from a longer file kindly supplied by Wayne Sullivan, University College, Dublin, Ireland, illustrates the use of `ledmac` to produce a Latin text edition, the *Periphyseon*, with Greek passages.<sup>31</sup> The Greek font used is that prepared by Silvio Levy and described in *TUGboat*.<sup>32</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.

---

```

3905 (*periph)
3906 % ledmixed.tex
3907 \documentclass{article}
3908 \usepackage{ledmac}
3909
3910 \noendnotes
3911 %% \overfullrule0 pt
3912 \lefthyphenmin=3
3913

```

The LaTeX version uses the `lgreek` package to access Silvio Levy's greek font. The `delims` package option subverts<sup>33</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.

```

3914 \let\Ma=$
3915 \let\aM=$
3916 \usepackage[delims]{lgreek}
3917
3918 % We need an addition to \no@expands since the \active $ in lgreek

```

---

<sup>31</sup>The bibliographic details of the forthcoming book are: Iohannis Scotti Erivgenae, *Periphyseon (De Divisione Naturae)* Liber Quartvs [Scriptores Latini Hiberniae vol. xii], (Dublin: School of Celtic Studies, Dublin Institute for Advanced Studies, forthcoming 1992).

<sup>32</sup>*TUGboat* 9 (1988), pp.20–24.

<sup>33</sup>It actually changes its category code.

```

3919 % causes problems:
3920 \newcommand{\morenoexpands}{\let$=0}
3921
3922 \makeatletter
3923
3924 \newbox\lp@rbox
3925
3926 \newcommand{\ffootnote}[1]{%
3927   \ifnumberedpar@
3928     \xright@appenditem{\noexpand\vffootnote{f}{\l@d@nums}{\@tag}{#1}}{%
3929       \to\inserts@list
3930     \global\advance\insert@count by 1
3931   % \else           %% may be used only in numbered text
3932   %   \vffootnote{f}{0|0|0|0|0|0}{#1}}%
3933   \fi\ignorespaces}
3934
3935 \newcommand{\gfootnote}[1]{%
3936   \ifnumberedpar@
3937     \xright@appenditem{\noexpand\vgfootnote{g}{#1}}{%
3938       \to\inserts@list
3939     \global\advance\insert@count by 1
3940   % \else           %% may be used only in numbered text
3941   %   \vgfootnote{g}{#1}}%
3942   \fi\ignorespaces}
3943
3944 \newcommand{\setlp@rbox}[3]{%
3945   {\parindent\z@\hspace=2.5cm\raggedleft\scriptsize
3946   \baselineskip 9pt%
3947   \global\setbox\lp@rbox=\vbox to\z@{\vss#3}}
3948
3949 \newcommand{\vffootnote}[2]{\setlp@rbox#2}
3950
3951 \newcommand{\vgfootnote}[2]{\def\rd@ta{#2}}
3952
3953
3954
3955 \renewcommand{\affixline@num}{%
3956   \ifsublines@
3957     \@l@tempcntb=\subline@num
3958     \ifnum\subline@num>\c@firstsublinenum
3959       \@l@tempcnta=\subline@num
3960       \advance\@l@tempcnta by-\c@firstsublinenum
3961       \divide\@l@tempcnta by\c@sublinenumincrement
3962       \multiply\@l@tempcnta by\c@sublinenumincrement
3963       \advance\@l@tempcnta by\c@firstsublinenum
3964     \else
3965       \@l@tempcnta=\c@firstsublinenum
3966     \fi
3967   %
3968   \ifcase\sub@lock

```



```

3969     \or
3970         \ifnum\sublock@disp=1
3971             \@l@tempcntb=0 \@l@tempcnta=1
3972         \fi
3973     \or
3974         \ifnum\sublock@disp=2 \else
3975             \@l@tempcntb=0 \@l@tempcnta=1
3976         \fi
3977     \or
3978         \ifnum\sublock@disp=0
3979             \@l@tempcntb=0 \@l@tempcnta=1
3980         \fi
3981     \fi
3982 \else
3983     \@l@tempcntb=\line@num
3984     \ifnum\line@num>\c@firstlinenum
3985         \@l@tempcnta=\line@num
3986         \advance\@l@tempcnta by-\c@firstlinenum
3987         \divide\@l@tempcnta by\c@linenumincrement
3988         \multiply\@l@tempcnta by\c@linenumincrement
3989         \advance\@l@tempcnta by\c@firstlinenum
3990     \else
3991         \@l@tempcnta=\c@firstlinenum
3992     \fi
3993 \ifcase\@lock
3994     \or
3995         \ifnum\lock@disp=1
3996             \@l@tempcntb=0 \@l@tempcnta=1
3997         \fi
3998     \or
3999         \ifnum\lock@disp=2 \else
4000             \@l@tempcntb=0 \@l@tempcnta=1
4001         \fi
4002     \or
4003         \ifnum\lock@disp=0
4004             \@l@tempcntb=0 \@l@tempcnta=1
4005         \fi
4006     \fi
4007 \fi
4008 %
4009 \ifnum\@l@tempcnta=\@l@tempcntb
4010     \@l@tempcntb=\line@margin
4011     \ifnum\@l@tempcntb>1
4012         \advance\@l@tempcntb by\page@num
4013     \fi
4014     \ifodd\@l@tempcntb
4015 %         #1\rlap{{\rightlinenum}}%
4016         \xdef\rd@ta{\the\line@num}%
4017     \else
4018         \llap{{\leftlinenum}}%#1%

```

```

4019     \fi
4020   \else
4021     %#1%
4022   \fi
4023   \ifcase\@lock
4024   \or
4025     \global\@lock=2
4026   \or \or
4027     \global\@lock=0
4028   \fi
4029   \ifcase\sub@lock
4030   \or
4031     \global\sub@lock=2
4032   \or \or
4033     \global\sub@lock=0
4034   \fi}
4035
4036 \lineation{page}
4037 \linenummargin{right}
4038 \footparagraph{A}
4039 \footparagraph{B}
4040
4041 \renewcommand{\notenumfont}{\footnotesize}
4042 \newcommand{\notetextfont}{\footnotesize}
4043
4044 \let\Afootnoterule=\relax
4045 \count\Afootins=825
4046 \count\Bfootins=825
4047
4048 \newcommand{\Aparafootfmt}[3]{%
4049   \ledsetnormalparstuff
4050   \scriptsize
4051   \notenumfont\printlines#1\enspace
4052   %       \lemmafont#1/#2\enskip
4053   \notetextfont
4054   #3\penalty-10\hskip 1em plus 4em minus.4em\relax}
4055
4056 \newcommand{\Bparafootfmt}[3]{%
4057   \ledsetnormalparstuff
4058   \scriptsize
4059   \notenumfont\printlines#1\enspace
4060   \select@lemmafont#1/#2\rbracket\enskip
4061   \notetextfont
4062   #3\penalty-10\hskip 1em plus 4em minus.4em\relax }
4063 \makeatother
4064
4065 \let\Afootfmt=\Aparafootfmt
4066 \let\Bfootfmt=\Bparafootfmt
4067 \def\lemmafont#1/#2/#3/#4/#5/#6/#7/{\scriptsize}
4068 \parindent=1em

```

```

4069
4070 \newcommand{\lmarpar}[1]{\edtext{}{\ffootnote{#1}}}
4071 \newcommand{\rmarpar}[1]{\edtext{}{\gfootnote{#1}}}
4072 \emergencystretch40pt
4073
4074 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4075
4076 \begin{document}
4077
4078 \beginnumbering
4079 \pstart
4080 \rmarpar{741C}
4081 \noindent \edtext{Incipit Quartus $PERIFUSEWN$}{%
4082 \lemma{incipit\ .~.\ $PERIFUSEWN$}\Bfootnote{\textit{om.\ R},
4083 incipit quartus \textit{M}}}
4084 \pend
4085 \medskip
4086
4087 \pstart
4088 \noindent \edtext{NVTRITOR}{\lemma{$ANAKEFALIOSIS$}\Bfootnote{\textit{FJP, lege}
4089 $<anakefala'iwsis$}}.\lmarpar{$ANAKEFALIOSIS$
4090 NATVRARVM} Prima nostrae
4091 \edtext{Physiologiae}{\lemma{physiologiae}\Bfootnote{phisiologiae
4092 \textit{P}, physeologiae \textit{R}}}
4093 intentio praecipuaque mat\-e\-ria erat
4094 \edtext{quod}{\Bfootnote{\textit{p}.\ natura \textit{transp.\ MR}}}
4095 \edtext{$UPEROUSIADES$}{\Bfootnote{\textit{codd.\ Vtrum}
4096 $<uperousi'wdhs$ (hoc est superessentialis) natura \textit{cum Gale
4097 (p.160) an} $<uperousi'oths$ (hoc est superessentialis natura)
4098 \textit{cum Floss (PL 122,741C) intelligendum sit, ambigitur}}}
4099 (hoc est superessentialis) natura sit causa creatrix existentium et
4100 non existentium omnium, a nullo creata, unum principium, una
4101 origo, unus et uniuersalis uniuersorum fons, a nullo manans, dum
4102 ab eo manant omnia, trinitas coessentialis in tribus substantiis,
4103 $ANARQOS$ (hoc est sine principio), principium et finis, una
4104 bonitas, deus unus,
4105 \edtext{$OMOUSIOS$}{\Bfootnote{\textit{codd., lege} $<omoo'usios$}}
4106 \edtext{et}{\lemma{\textbf{et}}\Bfootnote{\textit{R}\textsuperscript{1}, \textit{om.\ R}\textsuperscript{0}}}
4107 $UPEROUSIOS$ (id est coessentialis et superessentialis). Et, ut
4108 ait sanctus Epifanius, episcopus Constantiae Cypri, in
4109 \edtext{$AGKURATW$}{\Bfootnote{anchurato \textit{MR}}}
4110 sermone
4111 \edtext{de fide}{\Bfootnote{Glo\Ma\langle\am ssa\Ma\rangle\am: Ita
4112 enim uocatur sermo eius de fide $AGKURATOS$, id est procuratus
4113 \textit{mg.\ add.\ FJP}}}:
4114 \begin{itshape}Tria sancta, tria consancta, tria
4115 \edtext{agentia}{\Bfootnote{actiua \textit{MR}}},
4116 tria coagentia, tria
4117 \edtext{formantia}{\Bfootnote{formatiua \textit{MR}}},
4118

```

```

4119 tria conformantia, tria
4120 \edtext{operantia}{\Bfootnote{operatiua \textit{MR}}},
4121 tria cooperantia, tria subsistentia, tria\rmrpar{742C}
4122 consubsistentia sibi inuicem coexistentia. Trinitas haec
4123 sancta uocatur: tria existentia, una consonantia, una deitas
4124 \edtext{eiusdem}{\Bfootnote{eiusdemque \textit{M}}}}
4125 essentiae,
4126 \edtext{eiusdem uirtutis, eiusdem
4127 \edtext{subsistentiae}{\Bfootnote{substantiae \textit{R}}}}{\%
4128 \Bfootnote{\textit{om.\ M}}},
4129 similia
4130 \edtext{similiter}{\Bfootnote{ex simili \textit{MR}}}}
4131 aequalitatem gratiae operantur patris et filii et sancti spiritus.
4132 Quomodo autem
4133 \edtext{sunt}{\Bfootnote{\textit{om.\ M}}},
4134 ipsis relinquitur docere:
4135 \edtext{'Nemo enim nouit patrem nisi filius, neque filium nisi pater,
4136 et cuicumque filius reuelauerit'}{\Afootnote{Matth.\ 11, 27}};
4137 reuelatur autem per spiritum sanctum. Non ergo haec tria existentia
4138 aut ex ipso aut per ipsum aut ad ipsum in unoquoque digne intelliguntur,
4139 \Ma\!\mid! R, 264^{\rm r}\!\mid\!aM\ sicut ipsa reuelant:\end{itshape}
4140 $FWS, PUR, PNEUMA$
4141 \edtext{(hoc est lux, ignis, spiritus)}{\Afootnote{EPIPHANIVS,
4142 \textit{Ancoratus} 67; PG~43, 137C--140A; GCS 25, p.~82, 2--12}}.
4143 \pend
4144
4145 \pstart
4146 Haec, ut dixi, ab Epifanio tradita, ut quisquis interrogatus quae
4147 tria et quid unum in sancta trinitate debeat credere, sana fide
4148 \Ma\!\mid J, 1^{\rm v}\!\mid\!aM\ respondere ualeat, aut ad
4149 fidem accedens\rmrpar{743A} sic erudiatur. Et mihi uidetur
4150 spiritum pro calore posuisse, quasi dixisset in similitudine:
4151 lux, ignis, calor. Haec enim tria unius essentiae sunt. Sed cur
4152 lucem primo dixit, non est mirum. Nam et pater lux est et ignis
4153 et calor; et filius est lux, ignis, calor; et
4154 \edtext{spiritus sanctus}{\Bfootnote{sanctus spiritus \textit{R}}}}
4155 lux, ignis, calor. Illuminat enim pater, illuminat filius, illuminat
4156 spiritus sanctus: ex ipsis enim omnis scientia et sapientia donatur.
4157 \pend
4158 \endnumbering
4159
4160 \end{document}
4161
4162 </periph>

```

---

## A.6 Nijmegen

This example, illustrated in Figure 6, was provided in 2004 by Dirk-Jan Dekker of

the Department of Medieval History at the University of Nijmegen<sup>34</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.

---

```

4163 (*dekker)
4164 %% This is ledekker.tex, a sample critical text edition
4165 %% written in LaTeX2e with the ledmac package.
4166 %% (c) 2003--2004 by Dr. Dirk-Jan Dekker,
4167 %% University of Nijmegen (The Netherlands)
4168 %% (PRW) Modified slightly by PRW to fit the ledmac manual
4169
4170 \documentclass[10pt, letterpaper, onside]{article}
4171 \usepackage[latin]{babel}
4172 \usepackage{ledmac}
4173
4174 \lineation{section}
4175 \linenummargin{left}
4176 \sidenotemargin{outer}
4177
4178 \renewcommand{\notenumfont}{\footnotesize}
4179 \newcommand{\notetextfont}{\footnotesize}
4180
4181 %\let\Afootnoterule=\relax
4182 \let\Bfootnoterule=\relax
4183 \let\Cfootnoterule=\relax
4184
4185 \addtolength{\skip\Afootins}{1.5mm}
4186 %\addtolength{\skip\Bfootins}{1.5mm}
4187 %\addtolength{\skip\Cfootins}{1.5mm}
4188
4189 \makeatletter
4190
4191 \renewcommand*{\para@vfootnote}[2]{%
4192   \insert\csname #1footins\endcsname
4193   \bgroup
4194     \notefontsetup
4195     \footsplitskips
4196     \l@dparsespec #2\ledplinenumtrue % new from here
4197     \ifnum\@nameuse{previous@#1@number}=\l@dparsedstartline\relax
4198       \ledplinenumfalse
4199       \fi
4200     \ifnum\previous@page=\l@dparsedstartpage\relax
4201     \else \ledplinenumtrue \fi
4202     \ifnum\l@dparsedstartline=\l@dparsedendline\relax
4203     \else \ledplinenumtrue \fi
4204     \expandafter\xdef\csname previous@#1@number\endcsname{\l@dparsedstartline}
4205     \xdef\previous@page{\l@dparsedstartpage} % to here

```

<sup>34</sup>On 1st September 2004 the University changed its name to Radboud University.

```

4206     \setbox0=\vbox{\hsize=\maxdimen
4207         \noindent\csname #1footfmt\endcsname#2}%
4208     \setbox0=\hbox{\unvvh0}%
4209     \dp0=Opt
4210     \ht0=\csname #1footfudgefactor\endcsname\wd0
4211     \box0
4212     \penalty0
4213 \egroup
4214 }
4215
4216 \newcommand*\previous@A@number}{-1}
4217 \newcommand*\previous@B@number}{-1}
4218 \newcommand*\previous@C@number}{-1}
4219 \newcommand*\previous@page}{-1}
4220
4221 \newcommand{\abb}[1]{#1%
4222     \let\rbracket\nobrak\relax}
4223 \newcommand{\nobrak}{\textnormal{}}
4224 \newcommand{\morenoexpands}{%
4225     \let\abb=0%
4226 }
4227
4228 \newcommand{\Aparafootfmt}[3]{%
4229     \ledsetnormalparstuff
4230     \scriptsize
4231     \notenumfont\printlines#1\enspace
4232 % \lemmafont#1/#2\enskip
4233     \notetextfont
4234     #3\penalty-10\hskip 1em plus 4em minus.4em\relax}
4235
4236 \newcommand{\Bparafootfmt}[3]{%
4237     \ledsetnormalparstuff
4238     \scriptsize
4239     \notenumfont\printlines#1/%
4240     \ifledplinenum
4241         \enspace
4242     \else
4243         {\hskip 0em plus 0em minus .3em}
4244     \fi
4245     \select@lemmafont#1/#2\rbracket\enskip
4246     \notetextfont
4247     #3\penalty-10\hskip 1em plus 4em minus.4em\relax }
4248
4249 \newcommand{\Cparafootfmt}[3]{%
4250     \ledsetnormalparstuff
4251     \notenumfont\printlines#1\enspace
4252 % \lemmafont#1/#2\enskip
4253     \notetextfont
4254     #3\penalty-10\hskip 1em plus 4em minus.4em\relax}
4255

```

```

4256 \makeatother
4257
4258 \footparagraph{A}
4259 \footparagraph{B}
4260 \footparagraph{C}
4261
4262 \let\Afootfmt=\Aparafootfmt
4263 \let\Bfootfmt=\Bparafootfmt
4264 \let\Cfootfmt=\Cparafootfmt
4265
4266 \emergencystretch40pt
4267
4268 \author{Guillelmus de Berchen}
4269 \title{Chronicle of Guelders}
4270 \date{}
4271 \hyphenation{archi-epi-sco-po Huns-dis-brug li-be-ra No-vi-ma-gen-si}
4272 \begin{document}
4273 \maketitle
4274 \thispagestyle{empty}
4275
4276 \section*{St.\ Stephen's Church in Nijmegen}
4277 \beginnumbering
4278 \autopar
4279
4280 \noindent
4281 Nobilis itaque comes Otto imperio et dominio Novimagensi sibi,
4282 ut praefertur, impignoratis et commissis
4283 \edtext{proinde}{\Bfootnote{primum D}} praeesse cupiens, anno
4284 \textsc{liiii}\ledsidenote{1254} superius descripto, mense
4285 Iu\edtext{}{\Afootnote{p.\ 227~R}}nio, una cum iudice, scabinis
4286 ceterisque civibus civitatis Novimagensis, pro ipsius et
4287 inhabitantium in ea necessitate,\edtext{}{\Afootnote{p.\ 97~N}}
4288 commodo et utilitate, ut
4289 \edtext{ecclesia eius}{\Bfootnote{ecclesia D: eius eius H}}
4290 parochialis
4291 \edtext{\abb{extra civitatem}}{\Bfootnote{\textit{om.}~H}} sita
4292 destrueretur et \edtext{infra}{\Bfootnote{intra D}} muros
4293 \edtext{transfer}\edtext{}{\Afootnote{p.\ 129~D}}retur}%
4294 {\Bfootnote{transferreretur NH}}
4295 ac de novo construeretur, \edtext{a reverendo patre domino
4296 \edtext{Conrado de \edtext{Hofsteden}}%
4297 {\Bfootnote{Hoffstede D: Hoffsteden H}},
4298 archiepiscopo
4299 \edtext{Coloniensi}{\Bfootnote{Colononiensi H}}}%
4300 {\Cfootnote{Conrad of Hochstaden was archbishop of Cologne in
4301 1238--1261}}, licentiam}{\Cfootnote{William is confusing two
4302 charters that are five years apart. Permission from St.\ Apostles'
4303 Church in Cologne had been obtained as early as 1249. Cf.\ Sloet,
4304 \textit{Uorkondenboek} nr.\ 707 (14 November 1249):
4305 '\ldots{nos devotionis tue precibus annuentes, ut ipsam

```





## A.7 Irish verse

This example, illustrated in Figure 7, is a somewhat modified and shortened version of Wayne Sullivan's example demonstration for EDSTANZA.

The stanza lines are numbered according to the source verse lines, not according to the printed lines. For example, the sixth ('f') line in the first stanza is printed as two lines as the source line was too long to fit on one printed line. Note that if you process this yourself you will get error reports about counters the first time through; this is because alphabetic counters, like roman numerals, have no notion of zero.

As is fairly typical of critical edition typesetting, some of ledmac's internal macros had to be modified to get the desired effects.

```

4352 (*braonain)
4353 %% This is ledbraonain.tex, a sample critical verse edition.
4354 %% Originally written for TeX processing with edmac and edstanza
4355 %% by Wayne Sullivan.
4356 %% Extensively modified by Peter Wilson for LaTeX and the ledmac package.
4357
4358 \documentclass{article}
4359 \usepackage{ledmac}
4360
4361 \setlength{\textheight}{40pc}
4362 \setlength{\textwidth}{24pc}
4363 \bigskipamount=12pt plus 6pt minus 6pt
4364 \newcommand*\notetextfont{\footnotesize}
4365
4366 %% Just one footnote series
4367 \footparagraph{C}
4368 \count\Cfootins=800
4369 \makeatletter
4370 %% but using two different formats
4371 \def\xparafootfmt#1#2#3{%
4372   \ledsetnormalparstuff
4373   {\notenumfont\printlines#1}\enspace
4374   {\select@lemfont#1/#2}\rbracket\enskip
4375   \notetextfont #3\penalty-10 }
4376 \def\yparafootfmt#1#2#3{%
4377   \ledsetnormalparstuff
4378   {\notenumfont\printlines#1}\enspace
4379   {\select@lemfont#1/#2}\rbracket\enskip
4380   \notetextfont #3\penalty-10 }
4381
4382 \let\Cfootfmt=\xparafootfmt
4383 \skip\Cfootins=\bigskipamount
4384 \makeatother
4385
4386 %% This is the default, but just to demonstrate...
4387 \setlength{\stanzaindentbase}{20pt}
4388

```

```

4389 %%                MUST SET THE INDENTS
4390 %% indent multiples; first=hangindent.
4391 %% Must all be non-negative whole numbers
4392 \setstanzaindents{4,1,2,1,2,3,3,1,2,1}
4393
4394 %%                Set stanza line penalties
4395 %% Must be nonnegative whole numbers.
4396 %% An initial zero indicates no penalties.
4397 \setstanzapenalties{1,5000,10500,5000,10500,5000,5000,5000,0}
4398 %\setstanzapenalties{0}% the default
4399
4400 %%                Put some space between stanzas
4401 \let\endstanzaextra=\bigbreak % ==> \bigskip \penalty -200
4402
4403 %% (almost) force line break in foot paragraph
4404 \mathchardef\IMM=9999
4405 \def\lbreak{\hfil\penalty-\IMM}
4406
4407 %%                Number each stanza in bold
4408 \newcounter{stanzanum}
4409 \setcounter{stanzanum}{0}
4410 \newcommand*{\numberit}{%
4411   \flagstanza[0.5\stanzaindentbase]{\textbf{\thestanzanum}}}
4412 %% Use the hook to insert the number (and counteract a new line)
4413 %% and reset the line number to zero
4414 \newcommand*{\startstanzahook}{\refstepcounter{stanzanum}%
4415   \numberit\vskip-\baselineskip%
4416   \setlinenum{0}}
4417
4418 %% Want to label the footnotes with the stanza and line number
4419 %% We'll use \linenum to replace the sub-line number
4420 %% with the stanza number, redefining \edtext to do this
4421 %% automatically for us.
4422 %%%%%%%%%%%%%%%
4423 \makeatletter
4424
4425 \renewcommand{\edtext}[2]{\leavevmode
4426   \begingroup
4427     \no@expands
4428     \xdef\@tag{#1}%
4429     \set@line
4430     \global\insert@count=0
4431     \ignorespaces \linenum{||\the\c@stanzanum}#2\relax
4432     \flag@start
4433   \endgroup
4434   #1%
4435   \ifx\end@lemmas\empty \else
4436     \glp\end@lemmas\to\x@lemma
4437     \x@lemma
4438     \global\let\x@lemma=\relax

```

```

4439 \fi
4440 \flag@end}
4441
4442 %% We need only a very simple macro for footnote numbers,
4443 %% to produce the stanza number (sub-line) then the line number.
4444 \def\printstanzalines#1|#2|#3|#4|#5|#6|#7|{\begingroup
4445 #3\fullstop \linenumrep{#2}
4446 \endgroup}
4447 \let\oldprintlines\printlines
4448
4449 \makeatother
4450 %%%%%%%%%%%
4451
4452 \pagestyle{empty}
4453
4454 \begin{document}
4455
4456 \beginnumbering
4457
4458 \pstart \centering \textbf{22} \pend
4459
4460 \bigskip
4461 %% do not print line number beside heading
4462 \setcounter{firstlinenum}{1000}
4463 %% and heading footnotes use a different format
4464 \let\Cfootfmt=\yparafootfmt
4465
4466 \pstart
4467 \centerline{[Se\'an \'O Braon\'ain cct] chuim Tom\'ais U\'{i}
4468 \edtext{Dh\'unlaing}{\Cfootnote{\textbf{22} \textit{Teideal}: Dhuinnluinnng T,
4469 Se\'aghan Mac Domhnaill cct B\lbreak}}}
4470 \pend
4471
4472 \pstart
4473 \centerline{[Fonn: M\'airse\'ail U\'{i} Sh\'uilleabh\'ain (P\'ainseach
4474 na nUbh]}
4475 \pend
4476
4477 \bigskip
4478
4479 %% revert to the regular footnote format
4480 \let\Cfootfmt=\xparafootfmt
4481 %% but use our special number printing routine
4482 \let\printlines\printstanzalines
4483 %% Use letters for line numbering
4484 \linenumberstyle{alph}
4485 %% number lines from the second onwards
4486 \setcounter{firstlinenum}{2}
4487 \setcounter{linenumincrement}{1}
4488

```

```

4489 %% Each verse starts with \stanza.
4490 %% Lines end with &; the last line with \&.
4491
4492 \stanza
4493 A \edtext{dhuine}{\Cfootnote{dhuinne T}} gan ch\’eill do
4494 \edtext{mhaisligh}{\Cfootnote{mhaslaidh T, mhaslaig B}} an chl\’eir&
4495 is tharcaisnigh naomhscriupt na bhf\’aige,&
4496 na haitheanta \edtext{r\’eab}{\Cfootnote{raob T}} ’s an
4497 t-aifreann thr\’eig&
4498 \edtext{re}{\Cfootnote{le B}} taithneamh do chlaonchreideamh
4499 Mh\’artain,&
4500 c\’a rachair \edtext{’od}{\Cfootnote{dod B}} dh\’{\i}on ar
4501 \’Iosa Nasardha&
4502 nuair \edtext{chaithfimid}{\Cfootnote{chaithfam\’{\i}d T}} cruinn
4503 bheith ar \edtext{mhaicileann}{\Cfootnote{maoilinn B}} Josepha?&
4504 N\’{\i} caraid Mac Crae chuim t’anama ’
4505 \edtext{phl\’e}{\Cfootnote{phleidh T}}&
4506 n\’a Calvin \edtext{bhiais}{\Cfootnote{bh\’{\i}os B}} taobh
4507 \edtext{ris}{\Cfootnote{leis B}} an l\’a sin.\&
4508
4509 \stanza
4510 N\’ach damanta an sc\’eal don chreachaire
4511 \edtext{chlaon}{\Cfootnote{claon B}}&
4512 ghlac baiste na cl\’eire ’na ph\’aiste&
4513 ’s do \edtext{glanadh}{\Cfootnote{glannuig T}} mar ghr\’ein
4514 \’on bpeaca r\’o-dhaor&
4515 tr\’{\i} \edtext{ainibhfios}{\Cfootnote{ainnibhfios T, ainnbhfios B}}
4516 \edtext{\’Eva}{\Cfootnote{\’Eabha B}} rinn \’Adam,&
4517 tuitim ar\’{\i}s f\’e chuing na haicme sin&
4518 tug atharrach br\’{\i} don scr\’{\i}bhinn bheannaithe,&
4519 d’aistrigh b\’easa \edtext{agus}{\Cfootnote{is B}} reachta na cl\’eire&
4520 ’s n\’ach \edtext{tugann}{\Cfootnote{tuigionn T}} aon
4521 gh\’eilleadh don Ph\’apa?\&
4522
4523 \stanza
4524 Gach \edtext{scolaire}{\Cfootnote{sgollaire T}} baoth, n\’{\i}
4525 \edtext{mholaim}{\Cfootnote{mholluim T}} a cheird&
4526 \edtext{’t\’a ag obair}{\Cfootnote{’t\’ag ccobar T}}
4527 \edtext{le}{\Cfootnote{re B}} g\’eilleadh d\’a th\’aille&
4528 don \edtext{doirbhchoin chlaon}{\Cfootnote{dorbhchon daor B}}
4529 d\’a ngorthar Mac Crae,&
4530 deisceabal \edtext{straeigh}{\Cfootnote{straodhaig T}} as an
4531 gcoll\’aiste.&
4532 T\’a \edtext{\edtext{adaithe}{\Cfootnote{fadaighthe B}}}
4533 th\’{\i}os}{\Cfootnote{fhadoghthe ts\’{\i}os T}} in
4534 \’{\i}ochtar ifrinn,&
4535 gan \edtext{solas}{\Cfootnote{sollus T}} gan soilse i
4536 dt\’{\i}orthaibh dorcha,&
4537 tuigsint an l\’einn, gach
4538 \edtext{cuirpeacht}{\Cfootnote{cuirripeacht T}} d\’ein&

```

```
4539 is \edtext{Lucifer}{\Cfootnote{Luicifer T, L'ucifer B}} aosta
4540 'na \edtext{mh'aistir}{\Cfootnote{mhaighistir T}}.\&
4541
4542 \endnumbering
4543
4544 \end{document}
4545
4546 </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.

<b>Symbols</b>	
<code>\&amp;</code>	<i>20</i> , 2714, 2718, 2719, 2768, 2777, 2783, 2785, 4490, 4507, 4521, 4540
<code>\-</code>	..... 3866, 4093
<code>\@@line</code>	..... 1528
<code>\@wrindexm@m</code>	.. 2635, 2640, 2643, 2667
<code>\@EDROWFILL@</code>	..... 3006, <u>3213</u>
<code>\@M</code>	..... 1528, 2762, 2772
<code>\@MM</code>	..... 1273
<code>\@adv</code>	..... <u>463</u> , 624
<code>\@arabic</code>	..... 754
<code>\@aux</code>	..... 1808
<code>\@auxout</code>	.. 1810, 2634, 2639, 2642, 2666
<code>\@botlist</code>	..... 1761, 1763
<code>\@cclv</code>	1656, 1660, 1661, 1759, 1760, 1788
<code>\@checkend</code>	..... 2695
<code>\@colht</code>	..... 1638, 1764, 1776
<code>\@colroom</code>	..... 1764
<code>\@combinefloats</code>	..... 1633
<code>\@currenvir</code>	..... 2680, 2683, 2684
<code>\@currlist</code>	..... 1765, 1768
<code>\@dbldeferlist</code>	..... 1774, 1779, 1781
<code>\@dblfloatplacement</code>	..... 1778
<code>\@dbltoplist</code>	..... 1774, 1775
<code>\@deferlist</code>	..... 1761, 1770, 1771
<code>\@doclearpage</code>	..... <u>1743</u>
<code>\@edrowfill@</code>	..... <u>3213</u>
<code>\@ehb</code>	..... 1767
<code>\@emptytoks</code>	..... <u>2671</u> , 2681
<code>\@footnotemark</code>	..... <u>2119</u>
<code>\@footnotemarkA</code>	..... 2152, 2164
<code>\@footnotemarkB</code>	..... 2436, 2447
<code>\@footnotemarkC</code>	..... 2454, 2463
<code>\@footnotetext</code>	..... ..... <u>2115</u> , <u>2132</u> , 2537, 2563, 2590
<code>\@freelist</code>	..... 1631
<code>\@gobble</code>	658–662, 1930, 2692, 2975, 2992
<code>\@gobblethree</code>	..... <u>1922</u>
<code>\@h</code>	..... <u>1526</u>
<code>\@hilfs@count</code>	..... <u>2908</u>
<code>\@idxfile</code>	2625, 2635, 2640, 2643, 2667
<code>\@ifclassloaded</code>	..... ..... 31, 1706, 1731, 2114, 2612
<code>\@ifnextchar</code>	..... 2615, 2888
<code>\@ifpackageloaded</code>	..... 2663
<code>\@iiiminipage</code>	..... <u>2526</u>
<code>\@iiiparbox</code>	..... 2553
<code>\@indexfile</code>	..... 2657
<code>\@inputcheck</code>	..... 350
<code>\@insert</code>	..... 1126–1128, 1162–1164
<code>\@k</code>	..... <u>1526</u>
<code>\@kludgeins</code>	..... 1635, 1703
<code>\@l</code>	..... <u>380</u> , 607
<code>\@l@dttempcnta</code>	... <u>29</u> , 497, 499, 501, 502, 910, 911, 913, 915, 918, 919, 934, 970–974, 976, 983– 987, 989, 992, 995, 997, 1001, 1031, 1035, 1039, 1046, 1050, 1054, 1135, 1139, 1143, 1146, 1149, 1152, 1153, 3959–3963, 3965, 3971, 3975, 3979, 3985– 3989, 3991, 3996, 4000, 4004, 4009
<code>\@l@dttempcntb</code>	..... ... <u>29</u> , 217, 218, 223, 227, 231, 235, 238, 261, 262, 269, 273, 277, 279, 287, 288, 968, 980, 1001, 1009–1011, 1013, 1031, 1035, 1039, 1046, 1050, 1054, 1084–1086, 1088, 1141, 1142, 1988, 1989, 1994, 1998, 2002, 2006, 2009, 2088–2090, 2092, 3957, 3971, 3975, 3979, 3983, 3996, 4000, 4004, 4009–4012, 4014
<code>\@l@reg</code>	..... <u>380</u>
<code>\@lab</code>	..... 580, 1799, <u>1845</u>
<code>\@latexerr</code>	..... 1767
<code>\@led@extranofeet</code>	.. <u>1728</u> , 1741, 1754
<code>\@led@nofootfalse</code>	..... 1745–1753
<code>\@led@nofoottrue</code>	..... 1744
<code>\@led@testifnofoot</code>	..... <u>1743</u>
<code>\@line@@num</code>	..... <u>2904</u> , 3018
<code>\@listdepth</code>	..... 2539
<code>\@lock</code>	..... 146, <u>338</u> , 405, 407, 409, 422, 530, 531, 533, 534, 550, 551, 553, 841, 881,

- 940, 942, 943, 945, 1043, 1058,  
1060, 1062, 3993, 4023, 4025, 4027
- `\@lopL` ..... [447](#)  
`\@lopR` ..... [447](#)  
`\@makecol` ..... 1710  
`\@makefcolumn` .. 1770, 1771, 1779, 1781  
`\@makespecialcolbox` ..... 1636  
`\@maxdepth` ..... 1651, 1659  
`\@mem@extranofeet` ..... [1732](#)  
`\@mem@nofootfalse` ..... 1733–1740  
`\@midlist` ..... 1631, 1632  
`\@minipagefalse` ..... 2550  
`\@minipagerestore` ..... 2540  
`\@minus` ..... 1403, 1408, 2257, 2263  
`\@mpargs` ..... 2530, 2553  
`\@mpfn` ..... 2536, 2562, 2589  
`\@mpfootins` 2546, 2556, 2559, 2568, 2595  
`\@mpfootnotetext` ... 2537, 2563, 2590  
`\@mplistdepth` ..... 2539  
`\@nameuse` ..... 294,  
296, 1278, 1279, 1384, 1386,  
1457, 1458, 1498, 1500, 1576,  
1578, 1619, 1621, 1717, 1718,  
1720, 1721, 1723, 1725, 2181,  
2185, 2187, 2191, 2194, 2195,  
2200, 2204, 2208, 2211, 2214,  
2219, 2222, 2224, 2225, 2229,  
2236, 2242, 2287, 2297, 2305,  
2307, 2332, 2342, 2350, 2352,  
2378, 2382, 2383, 2391, 2399,  
2400, 2404, 2414, 2424, 2426,  
2490, 2491, 2493, 2494, 2499, 4197  
`\@nobreakfalse` ..... 761  
`\@nobreaktrue` ..... 759, 763  
`\@nowrindex` ..... 2624  
`\@oldnobreak` ..... 759, 761, 802  
`\@opcol` ..... 1771, 1789  
`\@opxtrafeetii` ..... 1680, [1681](#), 1716  
`\@outputbox` .....  
. 1638, 1640, 1641, 1656, 1658,  
1678, 1679, 2472, 2473, 2488, 2489  
`\@outputpage` ..... 1780  
`\@page` ..... [427](#)  
`\@parboxrestore` .... 1282, 2198, 2535  
`\@pboxswfalse` ..... 2528  
`\@pend` ..... [447](#)  
`\@pendR` ..... [447](#)  
`\@plus` ..... 1288,  
1403, 1408, 2257, 2263, 2293, 2338  
`\@ref` ..... [565](#), 610  
`\@ref@reg` ..... [567](#)  
`\@reinserts` ..... 1711  
`\@set` ..... [478](#), 629  
`\@setminipage` ..... 2541  
`\@showidx` ..... 2632, 2665  
`\@tag` ..... [670](#), 688, 729, 1181,  
1189, 1197, 1205, 1213, 1228,  
1236, 1244, 1252, 1260, 1904,  
1908, 1912, 1916, 1920, 3928, 4428  
`\@tempboxa` .... 1759, 1760, 2531, 2553  
`\@tempdima` ..... 1660, 2529, 2533  
`\@templ@d` ..... 2079, 2080  
`\@textbottom` ..... 1643  
`\@texttop` ..... 1639  
`\@thefnmarkA` ..... 2151, 2163  
`\@thefnmarkB` ..... 2435, 2446  
`\@thefnmarkC` ..... 2453, 2462  
`\@toksa` ..... [321](#), 329  
`\@toksb` ..... [321](#), 328–330  
`\@toplist` ..... 1761, 1762  
`\@whilenum` ..... 3187  
`\@whilesw` ..... 1771, 1780  
`\@wredindex` ..... 2654, [2656](#), 2895  
`\@x@sf` 2108, 2111, 2122, 2128, 2171, 2177  
`\@xloop` ..... 1160, [1167](#)  
`\@xympar` ..... [1976](#)  
`\^` ..... 374, 2793
- `\_` ..... 3221, 3226, 3234, 3567,  
3602, 3725, 3728, 4082, 4094,  
4095, 4107, 4114, 4128, 4133,  
4136, 4139, 4148, 4276, 4285,  
4287, 4293, 4302–4304, 4321–  
4323, 4327, 4334, 4337, 4340, 4343
- A**
- `\A@@footnote` ... [2899](#), 2977, 2994, 3013  
`\abb` ..... 4221,  
4225, 4291, 4322, 4327, 4333, 4337  
`\absline@num` ... 143, [337](#), 385, 388,  
391, 492, 495, 504, 518, 540,  
562, 572, 871, 893, 894, 902, 1125  
Abu Kamil Shuja' b. Aslam ..... 8  
`\actionlines@list` .....  
..... [340](#), 357, 360, 367, 492,  
495, 504, 518, 540, 562, 923, 926  
`\actions@list` .....  
. [340](#), 361, 368, 493, 502, 506,  
508, 520, 529, 542, 549, 563, 927



- \add@inserts ..... 848, [1114](#)  
 \add@inserts@next ..... [1114](#)  
 \add@penalties ..... 853, [1135](#)  
 \addfootins ..... [1714](#)  
 \addfootinsX ..... [2485](#)  
 \addtocounter ..... 803  
 \addtol@denvbody ... [2675](#), 2696, 2698  
 \addtolength ..... 4185–4187  
 Adelard II ..... 7  
 \advancelabel@refs ..... 1805, [1814](#)  
 \advanceline . 13, 69, 72, [624](#), 647, 662  
 \advancepageno ..... [1626](#)  
 \Aend ..... 1902, [1922](#)  
 \Aendnote ..... 15, [1901](#)  
 \affixline@num ..... 846, [962](#), 3955  
 \affixpstart@num ..... 847, [1073](#)  
 \affixside@note ..... 848, [2078](#)  
 \Afootfmt .....  
     3668, 3805, 3819, 3838, 4065, 4262  
 \Afootgroup ..... 1682  
 \Afootins ..... 1391, 1682, 1693,  
     1733, 1746, 3669–3671, 4045, 4185  
 \Afootnote ..... 14,  
     [1178](#), 2505, 2899, 2969, 2977,  
     2986, 2994, 3013, 3462, 3469,  
     3555, 3558, 3560, 3591, 3593,  
     3666, 3695, 3721, 3723–3725,  
     3728, 3739, 3807, 4136, 4141,  
     4285, 4287, 4293, 4321, 4323, 4340  
 \Afootnoterule ..... 3774, 4044, 4181  
 \Afootstart ..... 1682, 3672  
 \allowbreak .....  
     1570, 1609, 2298, 2343, 3513, 3526  
 \alph ..... 3439  
 \alpha ..... 3560, 3595  
 \aM ..... 3915, 4112, 4139, 4148  
 \ampersand ..... 21, [2714](#), 2785  
 \Aparafootfmt .. 4048, 4065, 4228, 4262  
 \AtBeginDocument ..... 1841, 2663  
 \author ..... 3446, 4268  
 \autopar ..... 10, 90, [810](#), 4278  
 \autoparfalse ..... 172, 811  
 \autopartrue ..... 824
- B**
- \B@@footnote ... [2899](#), 2978, 2995, 3014  
 \ballast ..... 36  
 \ballast@count ... 888, 891, 896, 1135  
 Beeton, Barbara Ann Neuhaus Friend 10
- \beginnumbering .....  
     .. 9, [127](#), 185, 766, 821, 3457,  
     3551, 3693, 3821, 4078, 4277, 4456  
 \beginnumberingR ..... 816  
 \Bend ..... 1906, [1922](#)  
 \Bendnote ..... 15, [1901](#)  
 \beta ..... 3560, 3595  
 \Bfootfmt ..... 3806, 4066, 4263  
 \Bfootgroup ..... 1683  
 \Bfootins ..... 1391,  
     1683, 1694, 1734, 1747, 4046, 4186  
 \Bfootnote .. 14, [1186](#), 2506, 2900,  
     2970, 2978, 2987, 2995, 3014,  
     3463, 3562, 3595, 3597, 3605,  
     3607, 3609, 3611, 3613, 3615,  
     3617, 3619, 3621, 3808, 4082,  
     4088, 4091, 4094, 4095, 4105,  
     4106, 4110, 4112, 4116, 4118,  
     4120, 4124, 4127, 4128, 4130,  
     4133, 4154, 4283, 4289, 4291,  
     4292, 4294, 4297, 4299, 4308,  
     4310, 4313, 4316–4318, 4320,  
     4322, 4324, 4326, 4327, 4330,  
     4334, 4335, 4337–4339, 4341, 4342  
 \Bfootnoterule ..... 3774, 4182  
 \Bfootstart ..... 1683  
 \bfseries ..... 754, 3537  
 \bigbreak ..... 4401  
 \bigskip . 3830, 3836, 4401, 4460, 4477  
 \bigskipamount ..... 4363, 4383  
 \body ..... 1168, 1169, 2716, 2784  
 \bodyfootmarkA ..... 26  
 \box 862, 864, 1452, 1466, 1512, 1527,  
     1656, 1760, 1788, 2395, 2408, 4211  
 \boxmaxdepth ..... 1659  
 \Bparafootfmt .. 4056, 4066, 4236, 4263  
 Bredon, Simon ..... 7  
 Breger, Herbert ..... 5, 8, 152  
 Brey, Gerhard ..... 7, 8  
 \brokenpenalty ..... 806  
 Burt, John ..... 6  
 Busard, Hubert L. L. .... 7  
 \bypage@false ..... [188](#), 201, 206  
 \bypage@true ..... [188](#), 196  
 \bypstart@false ..... [188](#), 197, 207  
 \bypstart@true ..... [188](#), 202
- C**
- \C@@footnote ... [2899](#), 2979, 2996, 3015  
 \c@addcolcount ..... [3181](#)

- \c@ballast ..... 888, 896
  - \c@firstlinenum ... 244, 982, 984,  
987, 989, 3984, 3986, 3989, 3991
  - \c@firstsublinenum . 248, 969, 971,  
974, 976, 3958, 3960, 3963, 3965
  - \c@footnoteA ..... 2156
  - \c@footnoteB ..... 2439
  - \c@footnoteC ..... 2456
  - \c@labidx ..... 2601
  - \c@linenumincrement .....  
..... 244, 985, 986, 3987, 3988
  - \c@mpfootnote ..... 2536, 2562, 2589
  - \c@page ..... 607
  - \c@pstart ..... 754
  - \c@stanzaum ..... 4431
  - \c@sublinenumincrement .....  
..... 248, 972, 973, 3961, 3962
  - \Cend ..... 1910, 1922
  - \Cendnote ..... 15, 1901
  - \centering ..... 4458
  - \centerline ..... 3117,  
3122, 3128, 3133, 3139, 3144,  
3365, 3367, 3823, 3833, 4467, 4473
  - \Cfootfmt ..... 4264, 4382, 4464, 4480
  - \Cfootgroup ..... 1684
  - \Cfootins ..... 1392, 1684,  
1695, 1735, 1748, 4187, 4368, 4383
  - \Cfootnote ..... 14, 1186,  
2507, 2901, 2971, 2979, 2988,  
2996, 3015, 3568, 3603, 4300,  
4301, 4343, 4468, 4493, 4494,  
4496, 4498, 4500, 4502, 4503,  
4505–4507, 4511, 4513, 4515,  
4516, 4519, 4520, 4524–4528,  
4530, 4532, 4533, 4535, 4538–4540
  - \Cfootnoterule ..... 4183
  - \Cfootstart ..... 1684
  - \ch@ck@l@ck ..... 999, 1027
  - \ch@cksub@l@ck ..... 978, 1027
  - \changes ..... 19, 1816
  - \char ..... 2714
  - \chardef ..... 1975, 2716, 2718
  - Chester, Robert of ..... 7
  - Claassens, Geert H. M. .... 8
  - class 1 feet ..... 114, 140
  - class 2 feet ..... 114, 115
  - \cleaders ..... 3176
  - \closeout ..... 599, 603, 1893
  - \clubpenalty ..... 806, 1139
  - \collation ..... 3807, 3825,  
3827, 3836, 3843, 3855, 3860,  
3862, 3872, 3881, 3883, 3889, 3898
  - \color@begingroup .....  
1283, 1462, 1663, 2199, 2404, 2532
  - \color@endgroup .....  
1284, 1462, 1667, 2200, 2404, 2551
  - \columnwidth .....  
.... 1281, 1433, 2197, 2534, 2577
  - Copernicus, Nicolaus ..... 7
  - \copy ..... 3681
  - \count ..... 1401, 1406, 1422,  
1426, 1550, 1553, 1594, 1616,  
2255, 2260, 2276, 2279, 2322,  
2325, 2363, 2366, 4045, 4046, 4368
  - \countdef ..... 1626
  - \Cparafootfmt ..... 4249, 4264
  - \cr ..... 1529, 1532
  - \CRITEXT ..... 2879
  - \critext 40, 663, 669, 2881, 2968, 3010
  - \cs .. 19, 707–709, 711–715, 1814–1817
  - \ctab ..... 2799, 3324, 3415
  - \ctabtext ..... 2803, 3334, 3419
- D**
- \D@footnote ... 2899, 2980, 2997, 3016
  - \date ..... 3447, 4270
  - \dcolerr ..... 2855, 2867
  - \dcoli ... 2825, 2857, 2858, 3225, 3230
  - \dcolii ..... 2826, 2858
  - \dcoliii ..... 2827, 2858
  - \dcoliv ..... 2828, 2859
  - \dcolix ..... 2833, 2860
  - \dcolv ..... 2829, 2859
  - \dcolvi ..... 2830, 2859
  - \dcolvii ..... 2831, 2860
  - \dcolviii ..... 2832, 2860
  - \dcolx ..... 2834, 2860
  - \dcolxi ..... 2835, 2861
  - \dcolxii ..... 2836, 2861
  - \dcolxiii ..... 2837, 2861
  - \dcolxiv ..... 2838, 2862
  - \dcolxix ..... 2843, 2863
  - \dcolxv ..... 2839, 2862
  - \dcolxvi ..... 2840, 2862
  - \dcolxvii ..... 2841, 2863
  - \dcolxviii ..... 2842, 2863
  - \dcolxx ..... 2844, 2863
  - \dcolxxi ..... 2845, 2864
  - \dcolxxii ..... 2846, 2864

- \dcolxxiii ..... 2847, 2864
  - \dcolxxiv ..... 2848, 2865
  - \dcolxxix ..... 2853, 2866
  - \dcolxxv ..... 2849, 2865
  - \dcolxxvi ..... 2850, 2865
  - \dcolxxvii ..... 2851, 2866
  - \dcolxxviii ..... 2852, 2866
  - \dcolxxx ..... 2854, 2866
  - \DeclareOption ..... 7–9
  - Dekker, Dirk-Jan .... 6, 31–33, 38, 196
  - \Dend ..... 1914, 1922
  - \Dendnote ..... 15, 1901
  - \Dfootgroup ..... 1685
  - \Dfootins 1392, 1685, 1696, 1736, 1749
  - \Dfootnote *14*, 1186, 2508, 2902, 2972,  
2980, 2989, 2997, 3016, 3565, 3600
  - \Dfootstart ..... 1685
  - \dimen 615, 616, 618–620, 622, 1402,  
1407, 1431–1433, 1436, 1534–  
1536, 1551, 1554, 1595, 1617,  
2256, 2261, 2262, 2277, 2280,  
2323, 2326, 2370–2372, 2375, 3669
  - \dimen@ ..... 1640, 1642
  - \disablel@dtabfeet .....  
..... 3031, 3048, 3062, 3076,  
3089, 3104, 3232, 3239, 3244,  
3252, 3257, 3265, 3283, 3299, 3422
  - \displaystyle ..... 2918, 3033,  
3035, 3064, 3066, 3091, 3093,  
3232, 3244, 3257, 3345, 3397, 3398
  - \displaywidowpenalty ..... 807
  - \divide ..... 972,  
985, 1433, 1535, 2372, 3961, 3987
  - \do@actions ..... 873, 900
  - \do@actions@fixedcode ..... 920, 933
  - \do@actions@next ..... 900
  - \do@ballast ..... 874, 888
  - \do@line ..... 789, 832
  - \do@linehook ..... 836, 855
  - \do@lockoff ..... 539
  - \do@lockoffL ..... 539
  - \do@lockon ..... 510
  - \do@lockonL ..... 510
  - \documentclass ..... 3433,  
3489, 3629, 3753, 3907, 4170, 4358
  - \doedindexlabel 2606, 2626, 2651, 2892
  - \doendnotes ..... 23, 1968
  - \doreinxtrafeeti ... 1689, 2471, 2492
  - \doreinxtrafeetii .. 1690, 1692, 1719
  - \dosplits ..... 1526
  - \dots 2797, 3567, 3602, 3725, 3728, 3885
  - \doublecolon ..... 3516, 3525
  - Downes, Michael ..... 37, 102, 104
  - \doxtrafeet ..... 1673
  - \doxtrafeeti ..... 1674, 2471, 2487
  - \doxtrafeetii ..... 1675, 1677
  - \dp ..... 1274, 1450,  
1464, 1640, 1660, 2393, 2406, 4209
  - \dummy@edtext ..... 651, 664
  - \dummy@ref ..... 566, 576
  - \dummy@text ..... 650, 663
- ### E
- \E@footnote ... 2899, 2981, 2998, 3017
  - \edaftertab .....  
.. *30*, *165*, 2805, 3272, 3306, 3324
  - edarrayc (environment) ..... 27, 3414
  - edarrayl (environment) ..... 27, 3414
  - edarrayr (environment) ..... 27, 3414
  - \EDATAB ..... 3364, 3372
  - \edatab ..... 2806, 3370
  - \edatabell ..... 2807, 3354
  - \edatleft ..... 29, 2808, 3156
  - \edatright ..... 29, 2809, 3164
  - \edbforetab .....  
.. *30*, *165*, 2804, 3272, 3306, 3324
  - \edfilldimen .....  
..... 3178, 3188, 3189, 3191, 3215
  - \edfont@info ..... 720, 723, 727
  - \EDINDEX ..... 2885
  - \edindex .... 27, 2613, 2649, 2885,  
2974, 2982, 2991, 2999, 3012,  
3024, 3041, 3057, 3071, 3084, 3099
  - \edindexlab ..... 27, 2601, 2607, 2610
  - \EDLABEL ..... 2883
  - \edlabel ..... 23, 658, 1798, 2607,  
2883, 3002, 3021, 3023, 3040,  
3056, 3070, 3083, 3098, 3231,  
3238, 3243, 3251, 3256, 3264, 3470
  - \edmakelabel ..... 24, 1888
  - \edpageref ..... 23, 659, 1849
  - \edrowfill . 29, 2812, 3003, 3006, 3213
  - \EDTAB ..... 3360, 3396
  - \edtabcolsep ..... 28, 2947,  
3036, 3053, 3066, 3080, 3094,  
3109, 3189, 3246, 3259, 3268, 3382
  - \EDTABINDENT ..... 3377, 3390
  - \edtabindent ..... 3220,  
3224, 3229, 3240, 3253, 3266, 3386
  - \EDTABtext ..... 3404

- edtabularc (environment) . . . . 27, [3418](#)
  - edtabularl (environment) . . . . 27, [3418](#)
  - edtabularr (environment) . . . . 27, [3418](#)
  - \EDTEXT . . . . . [2879](#)
  - \edtext . . . . . [14](#), [664](#), [685](#),  
2026–2028, 2135, 2240, 2879,  
2985, 3011, 3462, 3463, 3469,  
3554, 3557, 3559, 3561, 3563,  
3564, 3569, 3571, 3573, 3575,  
3577, 3579, 3581, 3583, 3585,  
3590, 3592, 3594, 3596, 3598,  
3599, 3604, 3606, 3608, 3610,  
3612, 3614, 3616, 3618, 3620,  
3695, 3721, 3723–3725, 3738,  
3823, 3826, 3834, 3842, 3848,  
3854, 3860, 3861, 3863, 3868,  
3871, 3874, 3880, 3882, 3888,  
3897, 4070, 4071, 4081, 4088,  
4091, 4094, 4095, 4105, 4106,  
4110, 4112, 4116, 4118, 4120,  
4124, 4126, 4127, 4130, 4133,  
4135, 4141, 4154, 4283, 4285,  
4287, 4289, 4291–4293, 4295,  
4296, 4299, 4308, 4310, 4313,  
4316–4318, 4320–4324, 4326,  
4327, 4330, 4333, 4335, 4337–  
4342, 4420, 4425, 4468, 4493,  
4494, 4496, 4498, 4500, 4502,  
4503, 4505–4507, 4511, 4513,  
4515, 4516, 4519, 4520, 4524–  
4528, 4530, 4532, 4535, 4538–4540
  - \advertdots . . . . . [30](#), 2811, [3175](#)
  - \advertline . . . . . [30](#), 2810, [3173](#)
  - \End . . . . . 1918, [1922](#)
  - \Endnote . . . . . [15](#), [1901](#)
  - \Efootgroup . . . . . 1686
  - \Efootins 1393, 1686, 1697, 1737, 1750
  - \Efootnote . . . . . [14](#), [1186](#),  
2509, 2903, 2973, 2981, 2990,  
2998, 3017, 3570, 3572, 3574,  
3576, 3578, 3580, 3582, 3584, 3586
  - \Efootstart . . . . . 1686
  - \emergencystretch . . . . . 4072, 4266
  - \empty 27, 118, 159, 162, 319, 320, 357,  
679, 695, 718, 732, 736, 742,  
773, 923, 981, 997, 1116–1118,  
1129, 1161, 1800, 2729, 3663, 4435
  - \enablel@dtabfeet . . . . . 3280,  
3296, 3314, 3322, 3332, 3340, [3422](#)
  - \end@lemmas . . . . .  
[649](#), 679, 680, 695, 696, 4435, 4436
  - \endashchar . . . . . [18](#), [1294](#), 1370, 1960
  - \endgraf . . . . . 784, 826, 830, 3656
  - \endline@num . . . . . [345](#), 583, 589
  - \endlock [13](#), [639](#), 657, 2771, 2775, 2777
  - \endminipage . . . . . [2543](#)
  - \endnumbering . . . . .  
. . . . . [9](#), 130, [152](#), 174, 184, 3472,  
3624, 3746, 3900, 4158, 4347, 4542
  - \endpage@num . . . . . [345](#), 582, 589
  - \endprint . . . . . [23](#), [1922](#), 1971
  - \endstanzaextra . . . . . [21](#), [2764](#), 4401
  - \endsub . . . . . [12](#), [615](#), 656, 3677
  - \endsubline@num . . . . . [345](#), 584, 590
  - \enskip . . . . . 1292, 1488, 1569, 1608,  
1923, 3500, 3512, 3525, 3532,  
3661, 3664, 3784, 3790, 4052,  
4060, 4232, 4245, 4252, 4374, 4379
  - \enspace . . . . . 1291,  
1487, 1568, 1607, 1923, 2204,  
2297, 2342, 2414, 3499, 3524,  
3531, 3761, 3789, 3798, 4051,  
4059, 4231, 4241, 4251, 4373, 4378
  - environments:  
  edarrayc . . . . . [27](#), [3414](#)  
  edarrayl . . . . . [27](#), [3414](#)  
  edarrayr . . . . . [27](#), [3414](#)  
  edtabularc . . . . . [27](#), [3418](#)  
  edtabularl . . . . . [27](#), [3418](#)  
  edtabularr . . . . . [27](#), [3418](#)  
  ledgroup . . . . . [22](#), [2561](#)  
  ledgroupsize . . . . . [22](#), [2574](#)  
  minipage . . . . . [22](#)
  - Euclid . . . . . 7
  - \ExecuteOptions . . . . . 10
  - \exit . . . . . 3763, 3891
  - \extensionchars . . . . . [31](#), [116](#), 136, 180
- F**
- \f@encoding . . . . . 727
  - \f@family . . . . . 727
  - \f@series . . . . . 727
  - \f@shape . . . . . 727
  - \f@x@l@cks . . . . . 1022, [1027](#)
  - Fairbairns, Robin . . . . . 26
  - \tfootnote . . . . . 3926, 4070
  - \finishstage . . . . . 3676, 3677
  - \first@linenum@out@false . . . . . [594](#), 600
  - \first@linenum@out@true . . . . . [594](#)

`\firstlinenum` ..... 10, 12, 253  
`\firstsublinenum` ..... 10, 12, 253  
`\fix@page` ..... 381, 434  
`\flag@end` ..... 608, 684, 700, 4440  
`\flag@start` ..... 608, 676, 692, 4432  
`\flag@stanza` ..... 21, 2780, 4411  
`\floatingpenalty` ..... 1273  
`\flush@notes` ..... 796, 1159  
Folkerts, Menso ..... 7  
`\fontencoding` ..... 1175  
`\fontfamily` ..... 1175  
`\fontseries` ..... 1175  
`\fontshape` ..... 1175  
`\footfootmarkA` ..... 26, 2159  
`\footfudgefiddle` 37, 1429, 1433, 2372  
`\footgroupA` ..... 2474  
`\footgroupB` ..... 2475  
`\footgroupC` ..... 2476  
`\footins` . 1655, 1662, 1666, 1701, 1745  
`\footinsA` 1738, 1751, 2155, 2474, 2480  
`\footinsB` 1739, 1752, 2442, 2475, 2481  
`\footinsC` 1740, 1753, 2456, 2476, 2482  
`\footnormal` .....  
1389, 1411–1415, 1715, 3546, 3547  
`\footnormalX` 26, 2245, 2467–2469, 2486  
`\footnote` ..... 3467, 3477, 3556  
`\footnoteA` ..... 26, 2149, 2518  
`\footnoteB` . 26, 2433, 2519, 3478–3480  
`\footnoteC` ..... 26, 2451, 2520  
`\footnoterule` . 1380, 1665, 2216, 2558  
`\footnotesize` ..... 1171, 2023,  
2024, 4041, 4042, 4178, 4179, 4364  
`\footparagraph` 16, 1417, 3544, 3667,  
3803, 4038, 4039, 4258–4260, 4367  
`\footparagraphX` ..... 26, 2357, 3440  
`\footsplitskips` ..... 1268,  
1271, 1446, 1460, 1559, 1598,  
2189, 2285, 2331, 2389, 2402, 4195  
`\footstartA` ..... 2474  
`\footstartB` ..... 2475  
`\footstartC` ..... 2476  
`\footthreecol` ..... 16, 1539, 3543  
`\footthreecolX` ..... 26, 2312  
`\foottwocol` ..... 16, 1583, 3542, 3804  
`\foottwocolX` ..... 26, 2266  
`\foottwocolX` ..... 2266  
`\fullstop` .....  
18, 309, 1294, 1367, 1369, 1371,  
1373, 1959, 1963, 3789, 3798, 4445

## G

`\g@addto@macro` ..... 1707, 1708,  
1716, 1719, 1722, 1724, 1732,  
2487, 2492, 2495, 2498, 2613, 2649  
Gädeke, Nora ..... 8  
`\get@linelistfile` ..... 353, 369  
`\getline@num` ..... 840, 869  
`\gfootnote` ..... 3935, 4071  
`\ggfootfmt` ..... 3654, 3668  
`\ggfootstart` ..... 3671, 3672  
`\gl@p` 331, 360, 361, 680, 696, 722, 926,  
927, 1122, 1126, 1162, 1803, 4436  
`\gl@poff` ..... 331, 332  
`\goodbreak` ..... 3800

## H

`\hangafter` .... 2760, 3509, 3675, 3760  
`\hangingsymbol` ..... 22, 2703, 2710  
`\hb@xt@` ..... 848,  
850, 862, 864, 3215, 3220, 3224,  
3229, 3240, 3253, 3266, 3343, 3345  
`\hfilneg` ..... 1528  
`\Hilfsbox` ..... 2820  
`\hilfsbox` ..... 2820, 2875, 2876,  
2918, 2930, 3020, 3033, 3050,  
3064, 3078, 3091, 3106, 3231,  
3233, 3238, 3242, 3243, 3245,  
3251, 3255, 3256, 3258, 3264, 3267  
`\hilfscount` .... 2820, 3381–3383, 3389  
`\HILFSskip` ..... 3374  
`\Hilfsskip` ..... 3221,  
3225, 3226, 3230, 3233, 3234,  
3241, 3242, 3245–3247, 3254,  
3255, 3258–3260, 3267–3269,  
3374, 3380, 3382, 3388, 3392, 3393  
`\hilfsskip` .....  
. 2820, 3019, 3020, 3035, 3052,  
3066, 3080, 3093, 3108, 3391–3393  
`\hphantom` ..... 2795  
`\Hy@temp@A` ..... 2637, 2638  
`\HyInd@ParenLeft` ..... 2638  
`\hyphenation` ..... 4271  
`\hyphenpenalty` ..... 3637

I

`\if@colmade` ..... 1771, 1780  
`\if@firstcolumn` 1003, 1078, 1773, 2082  
`\if@led@nofoot` ..... 1728, 1758  
`\if@nobreak` ..... 758  
`\ifautopar` ..... 775, 810

- \ifbypage@ [188](#), 428, 439, 905, 1330, 3644
  - \ifbypstart@ ..... [188](#), 790, 1355
  - \ifdim 616, 618, 620, 622, 2107, 2875, 3381
  - \iffirst@linenum@out@ ..... [594](#), 598
  - \ifhbox ..... 1511, 1516
  - \ifhmode ..... 2121, 2128, 2170, 2177
  - \ifinsethangingsymbol ..... [2706](#)
  - \ifinstanza ..... 775, 827, [2703](#), 2709
  - \ifl@d@dash ..... [1298](#), 1370, 1960
  - \ifl@d@elin ..... [1298](#),  
1348, 1372, 1373, 1950, 1962, 1963
  - \ifl@d@esl ..... [1298](#), 1373, 1963
  - \ifl@d@pnum .....  
[1298](#), 1336, 1367, 1371, 1938, 1961
  - \ifl@d@ssub ..... [1298](#), 1369, 1959
  - \ifl@dend@ ..... [1890](#), 1896
  - \ifl@dmemoir ..... 30, 2886
  - \ifl@dpairing ..... [120](#), 156, 1356
  - \ifl@dskipnumber ..... [642](#), 964
  - \ifl@dstartendok ..... 3185, [3195](#)
  - \ifledfinal ..... [5](#), [31](#)
  - \ifledplinenum ..... [1308](#), 1368, 4240
  - \ifledRcol ..... [120](#), 813, 1357
  - \ifleftnoteup ..... 2059, 2073
  - \ifnoeledmac ..... 6, 35
  - \ifnolinenums ..... 3640, 3659
  - \ifnoteschanged@ ..... 166, [349](#)
  - \ifnumberedpar@ ..... [747](#), 768,  
780, 1179, 1187, 1195, 1203,  
1211, 1226, 1234, 1242, 1250,  
1258, 1903, 1904, 1907, 1908,  
1911, 1912, 1915, 1916, 1919,  
1920, 1978, 2034, 2040, 2046,  
2134, 2140, 2228, 2239, 3927, 3936
  - \ifnumbering ..... [119](#),  
128, 153, 176, 191, 764, 777, 819
  - \ifnumberingR ..... [120](#), 814
  - \ifnumberline ..... [702](#), 870, 875, 963
  - \ifnumberpstart ... 755, 775, 799, 827
  - \ifodd ..... 1013, 1088, 2092, 4014
  - \ifpst@rtedL ..... [120](#)
  - \ifpstartnum ..... 1099, 1102, 1107
  - \ifreportnoidxfile ..... 2619
  - \ifrightnoteup ..... 2031, 2068
  - \ifshowindexmark ..... 2632, 2665
  - \ifsidepstartnum ..... 775, [1073](#)
  - \ifsublines@ .....  
307, [336](#), 417, 452, 457, 463,  
478, 496, 505, 519, 541, 588,  
590, 876, 912, 967, 1822, 1847, 3956
  - \ifvbox ..... 787, 1635, 1703
  - \ifvmode ..... 1804
  - \ifvoid ..... 1655,  
1682–1686, 1693–1697, 1701,  
1717, 1720, 1725, 1733–1740,  
1745–1753, 2474–2476, 2480–  
2482, 2490, 2493, 2499, 2511–  
2515, 2522–2524, 2546, 2568, 2595
  - \IMM ..... 4404, 4405
  - \indexentry ..... 2658
  - \initnumbering@reg ..... [127](#)
  - \InputIfFileExists ..... 370
  - \insert ..... 1266,  
1443, 1557, 1596, 1693–1697,  
1701, 1703, 1721, 2187, 2283,  
2329, 2386, 2480–2482, 2494, 4192
  - \insert@count ..... 564, 565,  
610, 674, 690, 1182, 1190,  
1198, 1206, 1214, 1229, 1237,  
1245, 1253, 1261, 2037, 2043,  
2049, 2143, 2231, 3930, 3939, 4430
  - \inserthangingsymbol ..... 848, 2707
  - \inserthangingsymbolfalse ..... 844
  - \inserthangingsymboltrue ..... 842
  - \inserthangingsymbol ..... [2706](#)
  - \insertlines@list .....  
... 159, [340](#), 366, 572, 1118, 1122
  - \insertparafootftmsep 1485, [1519](#), 2112
  - \inserts@list ..... 772, [1113](#),  
1116, 1126, 1161, 1162, 1181,  
1189, 1197, 1205, 1213, 1228,  
1236, 1244, 1252, 1260, 2036,  
2042, 2048, 2142, 2230, 3929, 3938
  - \instanzafalse ..... 2705, 2777
  - \instanzatrue ..... 2766
  - \interfootnotelinepenalty ..... 1272
  - \interlinepenalty 807, 1146, 1272, 2771
  - \interparanoteglue .... [16](#), [1479](#), 3771
  - \ipn@skip ..... 1477, [1479](#)
  - \itshape ..... 3676, 3785, 3791, 3799
- J**
- Jayaditya ..... 8
- K**
- Kabelschacht, Alois ..... 91
  - Krukov, Alexej ..... 67
- L**
- \l@d@wrindexhyp ..... [2630](#), [2663](#)

- \l@d@add . . . . . 737, 739, 743, 745
- \l@d@dashfalse . . . . . 1307, 1329, 1933
- \l@d@dashtrue . . . . .  
1333, 1339, 1351, 1936, 1941, 1953
- \l@d@elinfalse . . . . . 1303, 1336, 1938
- \l@d@elintrue . . 1336, 1338, 1938, 1940
- \l@d@end . . 1890, 1892, 1893, 1899,  
1902, 1906, 1910, 1914, 1918, 1975
- \l@d@err@UnequalColumns . . . . . 2961
- \l@d@eslfalse . . . . .  
. . . . . 1305, 1345, 1348, 1947, 1950
- \l@d@esltrue . . . 1348, 1350, 1950, 1952
- \l@d@index . . . . . 2615, 2617, 2888
- \l@d@makecol . . . . . 1647, 1710, 1789
- \l@d@nums 673, 720, 723, 731, 732, 745,  
1181, 1189, 1197, 1205, 1213,  
1228, 1236, 1244, 1252, 1260,  
1903, 1907, 1911, 1915, 1919, 3928
- \l@d@pnumfalse . . . . . 1299, 1329, 1933
- \l@d@pnumtrue . . . . . 1332, 1935
- \l@d@reinserts . . . . . 1700, 1711
- \l@d@section . . . . . 1899, 1922
- \l@d@set . . . . . 485, 636
- \l@d@ssubfalse . . . . . 1301, 1341, 1943
- \l@d@ssubtrue . . . . . 1343, 1945
- \l@d@wrindexm@m . . . . . 2629, 2630
- \l@dampcount . . . . . 2815,  
2957, 2959, 2964, 3229, 3239,  
3240, 3252, 3253, 3288, 3304, 3342
- \l@d@begin@stack . . . . . 2681, 2691–2693
- \l@d@bfnote . . . . . 2135, 2139
- \l@d@checkcols . . . . . 2914, 2926, 2954
- \l@d@checkstartend . . . . . 3184, 3195
- \l@d@chset@num . . . . . 384, 387, 485
- \l@d@colcount . . . . . 2815, 2857,  
2869, 2870, 2913, 2915, 2925,  
2927, 2955, 2959, 2964, 3025,  
3027, 3042, 3044, 3058, 3059,  
3072, 3073, 3085, 3086, 3100,  
3101, 3151, 3152, 3229, 3239,  
3240, 3252, 3253, 3284, 3286,  
3300, 3302, 3342, 3348, 3378, 3387
- \l@d@collect@tbody . . . . . 2683, 2690
- \l@d@collect@body . . . . .  
. . . . . 2678, 3414–3416, 3418–3420
- \l@d@colwidth . . . . . 2857, 2875, 2876,  
3019, 3150, 3215, 3241, 3254,  
3343, 3345, 3350, 3359, 3380, 3381
- \l@d@csnote . . . . . 2028, 2031
- \l@d@csnotetext . . . . . 856, 2054,  
2075, 2080, 2083, 2085, 2093, 2095
- \l@d@do@re@in@tra@feet 1688, 1702, 1708
- \l@d@do@foot@insert . . . . . 1648, 1653
- \l@d@do@x@tra@feet . . . . . 1670, 1673, 1707
- \l@d@ded@begin@mini . . . . . 1722, 2501, 2504
- \l@d@ded@end@mini . . . . . 1724, 2502, 2504
- \l@d@empty@d@ta . . . . . 837, 856
- \l@d@end@close . . . . . 1892, 1968
- \l@d@end@false . . . . . 1890, 1893
- \l@d@end@open . . . . . 1892, 1897
- \l@d@end@stuff . . . 137, 181, 1895, 1974
- \l@d@end@true . . . . . 1890, 1892
- \l@d@env@body . . . . . 2673, 2676, 2679–2681
- \l@d@f@am@begin@mini . . . 2495, 2501, 2517
- \l@d@f@am@end@mini . . . . . 2498, 2502, 2517
- \l@d@feet@begin@mini . . . . .  
. . . . . 2501, 2538, 2564, 2591
- \l@d@feet@end@mini 2501, 2549, 2571, 2598
- \l@d@get@line@margin . . . . . 214
- \l@d@get@lock@disp . . . . . 258, 286
- \l@d@get@ref@num . . . . . 1849,  
1850, 1852, 1853, 1855, 1856, 1863
- \l@d@get@sidenote@margin . . . . . 1985
- \l@d@gobble@arg . . . . . 2905, 2986–2990
- \l@d@gobble@darg . . . . . 2905, 2969–2973
- \l@d@label@parse . . . . . 1869, 1872
- \l@d@ld@ta . . 848, 856, 1002, 1079, 1091
- \l@d@lp@r@box . . . . . 862, 2016, 2058, 2064
- \l@d@l@sn@te . . . . . 849, 861
- \l@d@l@sn@te . . . . . 2026, 2031
- \l@d@make@labels 1808, 1811, 1833, 1842
- \l@d@memoir@false . . . . . 31
- \l@d@memoir@true . . . . . 31
- \l@d@mod@for@critext . . . . . 2967, 3423
- \l@d@mod@for@edtext . . . . . 2984, 3426
- \l@d@null@fills . . . . . 3001,  
3273, 3291, 3307, 3317, 3325, 3335
- \l@d@nump@starts@L . . . . . 120, 142
- \l@d@old@footnotetext . . . . . 2115, 2117
- \l@d@old@x@y@par . . . . . 1976
- \l@d@old@old@footnotetext . . 2132, 2147
- \l@d@p@r@se@foot@spec . . . . . 1312
- \l@d@p@airing@false . . . . . 120
- \l@d@p@airing@true . . . . . 120
- \l@d@p@arse@d@end@line . . . . . 1312, 4202
- \l@d@p@arse@d@end@page . . . . . 1312
- \l@d@p@arse@d@end@sub . . . . . 1312
- \l@d@p@arse@d@start@line . . . . .  
. . . . . 1312, 4197, 4202, 4204

- \l@dparsedstartpage . [1312](#), [4200](#), [4205](#)
- \l@dparsedstartsub . . . . . [1312](#)
- \l@dparsefootspec . . . . . [1312](#), [4196](#)
- \l@dpush@begins . . . . . [2687](#), [2691](#)
- \l@drd@ta . . . . . [850](#), [856](#), [1002](#), [1081](#), [1089](#)
- \l@dref@undefined . . . . .  
     . . . . . [1849](#), [1852](#), [1855](#), [1858](#)
- \l@drestorefills . . . . . [3001](#),  
     [3277](#), [3293](#), [3311](#), [3319](#), [3329](#), [3337](#)
- \l@drestoreforcritext . . . . . [2967](#), [3424](#)
- \l@dstoreforedtext . . . . . [2984](#), [3427](#)
- \l@drp@rbox . . . . . [864](#), [2016](#), [2067](#)
- \l@drsn@te . . . . . [851](#), [861](#)
- \l@drsnote . . . . . [2027](#), [2031](#)
- \l@dsetmaxcolwidth . . . . . [2874](#), [2920](#), [2932](#)
- \l@dskipnumberfalse . . . . . [642](#), [965](#)
- \l@dskipnumbertrue . . . . . [642](#), [956](#)
- \l@dstartendokfalse . . . . . [3199](#), [3203](#), [3207](#)
- \l@dstartendoktrue . . . . . [3197](#)
- \l@dtabaddcols . . . . . [3183](#), [3214](#)
- \l@dtabnoexpands . . . . . [665](#), [2788](#)
- \l@dunboxmpfoot . . . . . [2547](#), [2555](#), [2569](#), [2596](#)
- \l@dunhbox@line . . . . . [832](#)
- \l@dzeropenalties . . . . . [783](#), [805](#)
- Lück, Uwe . . . . . [6](#)
- \label . . . . . [24](#), [3455](#)
- \label@refs . . . . . [1801](#), [1803](#),  
     [1808](#), [1811](#), [1820](#), [1824](#), [1826](#), [1828](#)
- \labelref@list . . . . . [1794](#), [1800](#), [1803](#), [1847](#)
- \labelrefsparseline . . . . . [1814](#)
- \labelrefsparsesubline . . . . . [1814](#)
- \last@page@num . . . . . [434](#)
- \lastbox . . . . . [825](#), [839](#), [1472](#), [1510](#), [1515](#)
- \lastkern . . . . . [2107](#)
- \lastskip . . . . . [615](#), [619](#)
- Lavagnino, John . . . . . [5](#), [7](#)
- \lbreak . . . . . [4405](#), [4469](#)
- \ldots . . . . . [4305](#), [4307](#), [4333](#)
- Leal, Jeronimo@Leal, Jerónimo . . . . . [6](#)
- \led@err@AutoparNotNumbered . . . . .  
     . . . . . [78](#), [815](#), [820](#)
- \led@err@HighEndColumn . . . . . [105](#), [3204](#)
- \led@err@LineationInNumbered . . . . . [55](#), [192](#)
- \led@err@LowStartColumn . . . . . [105](#), [3200](#)
- \led@err@NumberingNotStarted . . . . . [39](#), [170](#)
- \led@err@NumberingShouldHaveStarted . . . . .  
     . . . . . [39](#), [183](#)
- \led@err@NumberingStarted . . . . . [39](#), [129](#)
- \led@err@PendNoPstart . . . . . [78](#), [781](#)
- \led@err@PendNotNumbered . . . . . [78](#), [778](#)
- \led@err@PstartInPstart . . . . . [78](#), [769](#)
- \led@err@PstartNotNumbered . . . . . [78](#), [765](#)
- \led@err@ReverseColumns . . . . . [105](#), [3208](#)
- \led@err@TooManyColumns . . . . . [105](#), [2871](#)
- \led@err@UnequalColumns . . . . . [105](#)
- \led@mess@NotesChanged . . . . . [45](#), [167](#)
- \led@mess@SectionContinued . . . . . [53](#), [179](#)
- \led@warn@BadAction . . . . . [92](#), [958](#)
- \led@warn@BadAdvancelineLine . . . . . [68](#), [472](#)
- \led@warn@BadAdvancelineSubline . . . . .  
     . . . . . [68](#), [466](#)
- \led@warn@BadLineation . . . . . [58](#), [209](#)
- \led@warn@BadLinenummargin . . . . . [58](#), [237](#)
- \led@warn@BadLockdisp . . . . . [58](#), [264](#)
- \led@warn@BadSetline . . . . . [74](#), [627](#)
- \led@warn@BadSetlinenum . . . . . [74](#), [634](#)
- \led@warn@BadSidenotemargin . . . . . [101](#), [2008](#)
- \led@warn@BadSublockdisp . . . . . [58](#), [290](#)
- \led@warn@DuplicateLabel . . . . . [94](#), [1836](#)
- \led@warn@NoIndexFile . . . . . [103](#), [2620](#)
- \led@warn@NoLineFile . . . . . [66](#), [375](#)
- \led@warn@NoMarginpars . . . . . [99](#), [1979](#)
- \led@warn@RefUndefined . . . . . [94](#), [1860](#)
- \ledfinalfalse . . . . . [8](#)
- \ledfinaltrue . . . . . [7](#)
- \ledfootinsdim . . . . .  
     . . . . . [1389](#), [1402](#), [1407](#), [2256](#), [2262](#)
- ledgroup (environment) . . . . . [22](#), [2561](#)
- ledgroupsize (environment) . . . . . [22](#), [2574](#)
- \ledleftnote . . . . . [25](#), [2026](#)
- \ledlinenum . . . . . [302](#)
- \ledllfill . . . . . [850](#), [866](#), [2578](#), [2582](#)
- \ledlsnotefontsetup . . . . . [25](#), [2019](#), [2057](#)
- \ledlsnotesep . . . . . [25](#), [862](#), [2019](#)
- \ledlsnotewidth . . . . . [25](#), [2019](#), [2057](#)
- \ledmac@error . . . . . [34](#),  
     [37](#), [40](#), [42](#), [44](#), [56](#), [79](#), [82](#), [85](#),  
     [88](#), [90](#), [106](#), [108](#), [111](#), [113](#), [115](#), [3209](#)
- \ledmac@warning . . . . .  
     . . . . . [33](#), [59](#), [61](#), [63](#), [65](#), [67](#), [69](#),  
     [72](#), [75](#), [77](#), [93](#), [95](#), [97](#), [100](#), [102](#), [104](#)
- \ledplinenumfalse . . . . . [31](#), [4198](#)
- \ledplinenumtrue . . . . .  
     . . . . . [31](#), [1309](#), [4196](#), [4201](#), [4203](#)
- \ledrightnote . . . . . [25](#), [2026](#), [3468](#)
- \ledrlfill . . . . . [850](#), [866](#), [2579](#), [2586](#)
- \ledrsnotefontsetup . . . . . [25](#), [2019](#), [2066](#)
- \ledrsnotesep . . . . . [25](#), [864](#), [2019](#)
- \ledrsnotewidth . . . . . [25](#), [2019](#), [2066](#), [3444](#)



- `\ledsetnormalparstuff` .....  
     ..... [1286](#), [1486](#), [2203](#), [2413](#),  
     3498, [3530](#), [3782](#), [3788](#), [4049](#),  
     4057, [4229](#), [4237](#), [4250](#), [4372](#), [4377](#)  
`\ledsidenote` ..... [25](#), [2026](#), [4284](#)  
`\left` ..... [3158](#), [3161](#), [3166](#), [3169](#)  
`\leftctab` ..... [3228](#), [3326](#)  
`\lefthyphenmin` ..... [3912](#)  
`\leftlinenum` .....  
     .. [12](#), [302](#), [1004](#), [1016](#), [3649](#), [4018](#)  
`\leftltab` ..... [3219](#), [3308](#)  
`\leftnoteuptrue` ..... [2074](#)  
`\leftpstartnum` ..... [1073](#)  
`\leftrtab` ..... [3223](#), [3274](#)  
`Leibniz` ..... [8](#)  
`\lemma` ..... [15](#), [729](#), [3567](#), [3602](#),  
     3725, [3728](#), [3824](#), [3835](#), [3860](#),  
     3885, [4082](#), [4088](#), [4091](#), [4106](#), [4333](#)  
`\lemmafont` .... [4052](#), [4067](#), [4232](#), [4252](#)  
`\letsforverteilen` ..... [3009](#),  
     3034, [3051](#), [3065](#), [3079](#), [3092](#), [3107](#)  
`Levy, Silvio` ..... [191](#)  
`\line@list` [162](#), [340](#), [365](#), [590](#), [718](#), [722](#)  
`\line@list@stuff` ..... [136](#), [180](#), [596](#)  
`\line@margin` ... [214](#), [1009](#), [1084](#), [4010](#)  
`\line@num` ..... [144](#), [306](#),  
     [334](#), [389](#), [423](#), [429](#), [440](#), [470](#),  
     [471](#), [473](#), [481](#), [486](#), [487](#), [499](#),  
     [583](#), [587](#), [882](#), [906](#), [915](#), [980](#),  
     [982](#), [983](#), [992](#), [993](#), [1846](#), [3644](#)–  
     [3646](#), [3649](#), [3776](#), [3983](#)–[3985](#), [4016](#)  
`\line@set` ..... [733](#), [734](#)  
`\lineation` ..... [11](#), [56](#), [59](#),  
     [190](#), [3538](#), [3652](#), [3809](#), [4036](#), [4174](#)  
`\linenum` ..... [15](#), [730](#), [1885](#),  
     [2904](#), [2975](#), [2992](#), [3018](#), [4419](#), [4431](#)  
`\linenum@out` ... [593](#), [599](#), [601](#), [603](#),  
     [604](#), [607](#), [609](#), [612](#), [617](#), [621](#),  
     [624](#), [629](#), [636](#), [639](#), [640](#), [646](#), [1799](#)  
`\linenumberlist` ..... [12](#), [27](#), [981](#), [993](#)  
`\linenumberstyle` ..... [13](#), [293](#), [4484](#)  
`\linenumincrement` ..... [10](#), [12](#), [253](#)  
`\linenummargin` ..... [12](#), [61](#),  
     [214](#), [3539](#), [3651](#), [3810](#), [4037](#), [4175](#)  
`\linenumr@p` ..... [293](#)  
`\linenumrep` ..... [293](#), [306](#),  
     [1368](#), [1372](#), [1846](#), [1958](#), [1962](#), [4445](#)  
`\linenums` ..... [3642](#)  
`\linenumsep` ..... [12](#), [302](#), [1103](#),  
     [1108](#), [2021](#), [2022](#), [3757](#), [3768](#), [3769](#)  
`\lineref` .... [23](#), [660](#), [1852](#), [2610](#), [3482](#)  
`\linewidth` ..... [848](#)  
`\list@clear` ..... [320](#), [365](#)–[368](#), [772](#)  
`\list@clearing@reg` ..... [352](#), [364](#)  
`\list@create` .....  
     ... [319](#), [340](#)–[343](#), [649](#), [1113](#), [1794](#)  
`\lmarpar` ..... [4070](#), [4089](#)  
`\lock@disp` ..... [258](#),  
     [1045](#), [1049](#), [1053](#), [3995](#), [3999](#), [4003](#)  
`\lock@off` ..... [512](#), [513](#), [539](#), [640](#)  
`\lock@on` ..... [510](#), [639](#)  
`\lockdisp` ..... [13](#), [63](#), [258](#)  
`Lorch, Richard` ..... [8](#)  
`\lp@rbox` ..... [3924](#), [3947](#)  
`\ltab` ..... [2800](#), [3306](#), [3414](#)  
`\ltabtext` ..... [2802](#), [3316](#), [3418](#)  
`Luecking, Dan` ..... [43](#)
- M
- `\m@m@makecolfloats` ..... [1630](#), [1649](#)  
`\m@m@makecolintro` ..... [1630](#)  
`\m@m@makecoltext` ..... [1630](#), [1650](#)  
`\m@mdodoreinextrafeet` ..... [1708](#)  
`\m@mdoextrafeet` ..... [1707](#)  
`\m@mmf@check` ..... [2106](#), [2123](#), [2172](#)  
`\m@mmf@prepare` .....  
     ..... [2103](#), [2118](#), [2127](#), [2153](#),  
     [2165](#), [2176](#), [2437](#), [2448](#), [2455](#), [2464](#)  
`\m@th` ..... [3176](#)  
`\Ma` ..... [3914](#), [4112](#), [4139](#), [4148](#)  
`\makehboxofhboxes` .....  
     ..... [1492](#), [1502](#), [1507](#), [2419](#), [2428](#)  
`\makeindex` ..... [2613](#), [2649](#)  
`\makememindexhook` ..... [2613](#)  
`\maketitle` ..... [3449](#), [4273](#)  
`\managestanza@modulo` .... [2735](#), [2752](#)  
`\marg` ..... [17](#)  
`\marginparwidth` ..... [2019](#), [2020](#)  
`\mathchardef` ..... [2730](#), [4404](#)  
`\maxdepth` ..... [1651](#)  
`\maxdimen` [1447](#), [1461](#), [2390](#), [2403](#), [4206](#)  
`Mayer, Gyula` ..... [8](#)  
`\measurebody` .. [3276](#), [3282](#), [3310](#), [3328](#)  
`\measurecell` ..... [2912](#), [2938](#)  
`\measuremrow` ..... [2936](#), [3287](#)  
`\measuretbody` .. [3292](#), [3298](#), [3318](#), [3336](#)  
`\measuretcell` ..... [2924](#), [2943](#)  
`\measuretrow` ..... [2941](#), [3303](#)  
`\medskip` ..... [4085](#)  
`\message` ..... [54](#), [135](#)

- \MessageBreak . . . . . 37
  - \mid . . . . . 4139, 4148
  - Middleton, Thomas . . . . . 8, 55
  - minipage (environment) . . . . . 22
  - Mittelbach, Frank . . . . . 7
  - \morenoexpands . . . . . 38, 652, 3920, 4224
  - \moveleft . . . . . 3221, 3226, 3234
  - \moveright . . . . . 3247, 3260, 3269
  - \mpAfootgroup . . . . . 2511
  - \mpAfootins . . . . . 1219, 2511
  - \mpAfootnote . . . . . 1225, 2505
  - \mpBfootgroup . . . . . 2512
  - \mpBfootins . . . . . 1219, 2512
  - \mpBfootnote . . . . . 1225, 2506
  - \mpCfootgroup . . . . . 2513
  - \mpCfootins . . . . . 1219, 2513
  - \mpCfootnote . . . . . 1225, 2507
  - \mpDfootgroup . . . . . 2514
  - \mpDfootins . . . . . 1219, 2514
  - \mpDfootnote . . . . . 1225, 2508
  - \mpEfootgroup . . . . . 2515
  - \mpEfootins . . . . . 1219, 2515
  - \mpEfootnote . . . . . 1225, 2509
  - \mpfootgroupA . . . . . 2522
  - \mpfootgroupB . . . . . 2523
  - \mpfootgroupC . . . . . 2524
  - \mpfootinsA . . . . . 2161, 2522
  - \mpfootinsB . . . . . 2444, 2523
  - \mpfootinsC . . . . . 2460, 2524
  - \mpfootnoteA . . . . . 2161, 2518
  - \mpfootnoteB . . . . . 2444, 2519
  - \mpfootnoteC . . . . . 2460, 2520
  - \mpnormalfootgroup . . . . . 1383, 1405
  - \mpnormalfootgroupX . . . . . 2221, 2259
  - \mpnormalvfootnote . . . . .  
     . . . . . 1277, 1404, 1544, 1588
  - \mpnormalvfootnoteX . . . . .  
     . . . . . 2193, 2258, 2271, 2317
  - \mppara@footgroup . . . . . 1425, 1497
  - \mppara@footgroupX . . . . . 2365, 2417
  - \mppara@vfootnote . . . . . 1424, 1456
  - \mppara@vfootnoteX . . . . . 2364, 2385
  - \mpthreecolfootgroup . . . . . 1545, 1575
  - \mpthreecolfootgroupX . . . . . 2318, 2345
  - \mpthreecolfootsetup . . . . . 1546, 1552
  - \mpthreecolfootsetupX . . . . . 2319, 2321
  - \mptwocolfootgroup . . . . . 1589, 1615
  - \mptwocolfootgroupX . . . . . 2272, 2300
  - \mptwocolfootsetup . . . . . 1590, 1615
  - \mptwocolfootsetupX . . . . . 2273, 2275
  - \mpvAfootnote . . . . . 1227, 1231
  - \mpvBfootnote . . . . . 1235, 1239
  - \mpvCfootnote . . . . . 1243, 1247
  - \mpvDfootnote . . . . . 1251, 1255
  - \mpvEfootnote . . . . . 1259, 1263
  - \mpvfootnoteA . . . . . 2165
  - \mpvfootnoteB . . . . . 2448
  - \mpvfootnoteC . . . . . 2464
  - \multfootsep . . . . . 26, 2100, 2110
  - \multiplefootnotemarker . . . . .  
     . . . . . 2100, 2104, 2105, 2107
- N
- \n@num . . . . . 560, 646
  - \n@num@reg . . . . . 560
  - \NeedsTeXFormat . . . . . 2
  - \newline . . . . . 607, 850
  - \newbox . . . . . 747, 750, 2016,  
     2017, 2820, 2822, 3411, 3412, 3924
  - \newcounter . . . . . 244, 246, 248,  
     250, 753, 889, 1819, 1823, 2156,  
     2439, 2456, 2603, 2738, 3181, 4408
  - \newif . . . . . 5, 6, 30, 119, 120, 122,  
     125, 126, 188, 189, 336, 349,  
     594, 642, 702, 748, 755, 810,  
     1074, 1099, 1298, 1300, 1302,  
     1304, 1306, 1308, 1728, 1891,  
     2031, 2073, 2704, 2706, 3195, 3640
  - \newinsert . . . . . 1219–1223, 1391–1393,  
     2155, 2166, 2442, 2449, 2458, 2465
  - \newlength . . . . . 302, 2722
  - \newlinechar . . . . .  
     . . . . . 1901, 1905, 1909, 1913, 1917
  - \newparafootfmt . . . . . 3787, 3805, 3838
  - \newread . . . . . 350
  - \newtwocolfootfmt . . . . . 3793, 3806
  - \newwrite . . . . . 593, 1890
  - \NEXT . . . . . 2908,  
     2913, 2916, 2921, 2922, 2925,  
     2928, 2933, 2934, 2937, 2939,  
     2940, 2942, 2944, 2945, 2950,  
     3113, 3116, 3118, 3119, 3121,  
     3123, 3124, 3127, 3129, 3130,  
     3132, 3134, 3135, 3138, 3140,  
     3141, 3143, 3145, 3146, 3151,  
     3153, 3154, 3348, 3351, 3352,  
     3357, 3361, 3362, 3378, 3384, 3385
  - \Next . . . . . 2950, 3026,  
     3028, 3037, 3038, 3043, 3045,  
     3054, 3055, 3058, 3060, 3067,

- 3068, 3072, 3074, 3081, 3082,  
3085, 3087, 3095, 3096, 3100,  
3102, 3110, 3111, 3366, 3368, 3369
- `\next@action` ..... 93, 361,  
895, 903, 904, 909, 910, 918, 927
- `\next@actionline` .....  
..... 358, 360, 894, 902, 924, 926
- `\next@insert` .....  
773, 1117, 1120, 1122, 1125, 1129
- `\next@page@num` .....  
..... 149, 392, 394, 432, 444, 493
- `\no@expands` .. [652](#), 671, 687, 3918, 4427
- `\noalign` ..... 1531
- `\nobrak` ..... 4222, 4223
- `\noeledmactrue` ..... 9
- `\noendnotes` ..... [23](#),  
[1974](#), 3442, 3492, 3632, 3766, 3910
- `\noindent` ..... 827, 1071, 1448,  
1462, 1495, 1505, 2391, 2404,  
2422, 2431, 4081, 4088, 4207, 4280
- `\nolinenums` ..... 3639, 3641
- `\nolinenumsfalse` ..... 3642
- `\nolinenumstrue` ..... 3641
- `\nonumparafootfmt` .. 3780, 3781, 3819
- `\normal@footnotemarkX` ... [2179](#), [2247](#)
- `\normal@pars` .....  
... 155, 774, [830](#), 1287, 1562,  
1601, 2290, 2335, 3504, 3518, 3794
- `\normalbfnoteX` ..... [2227](#), 2240
- `\normalbodyfootmarkX` .... [2184](#), 2248
- `\normalcolor` 1385, 1499, 1577, 1620,  
1664, 2223, 2306, 2351, 2425, 2557
- `\normalfont` 304, 1172, 2101, 2185, 3800
- `\normalfootfmt` . [1286](#), 1397, 3497, 3545
- `\normalfootfmtX` ..... [2202](#), 2251
- `\normalfootfootmarkX` .... [2207](#), 2252
- `\normalfootgroup` ..... [1381](#), 1398
- `\normalfootgroupX` ..... [2218](#), 2253
- `\normalfootnoterule` ..... [1380](#), 1400
- `\normalfootnoteruleX` [2216](#), 2254, 2362
- `\normalfootstart` ..... [1376](#), 1395
- `\normalfootstartX` ..... [2210](#), 2246
- `\normalvfootnote` ..... [1265](#), 1396
- `\normalvfootnoteX` ..... [2186](#), 2249
- `\nospeak` ..... 3686, 3697
- `\nospeaker` ..... 3685
- `\note` 3808, 3849, 3864, 3869, 3875, 3886
- `\notefontsetup` ..... [18](#), [1171](#),  
1267, 1280, 1430, 1445, 1459,  
1481, 1494, 1504, 1558, 1571,  
1597, 1610, 1922, 2188, 2196,  
2284, 2300, 2330, 2345, 2369,  
2388, 2401, 2421, 2430, 3655, 4194
- `\notenumfont` ..... [18](#),  
[1172](#), 1291, 1487, 1568, 1607,  
1922, 2204, 2297, 2342, 2414,  
3499, 3511, 3524, 3531, 3537,  
3789, 3798, 4041, 4051, 4059,  
4178, 4231, 4239, 4251, 4373, 4378
- `\noteschanged@false` ..... [349](#), 371
- `\noteschanged@true` .....  
.... 160, 163, [349](#), 376, 719, 1119
- `\notetextfont` 4042, 4053, 4061, 4179,  
4233, 4246, 4253, 4364, 4375, 4380
- `\nulledindex` .... [2885](#), 2974, 2991,  
3024, 3041, 3057, 3071, 3084, 3099
- `\nullsetzen` ..... [3148](#), 3285, 3301
- `\num@lines` . [747](#), 784, 1136, 1142, 1145
- `\numberedpar@false` ..... [747](#)
- `\numberedpar@true` ..... [747](#), 776
- `\numberingfalse` ..... [119](#), 154
- `\numberingtrue` ..... [119](#), 132, 174
- `\numberit` ..... 4410, 4415
- `\numberlinefalse` ..... [11](#)
- `\numberlinetrue` ..... [11](#), 703
- `\numberpstartfalse` ..... [11](#), [752](#), 756
- `\numberpstarttrue` ..... [11](#), [752](#)
- `\numlabfont` .. [18](#), [302](#), 3646, 3649, 3776
- O**
- `\oldprintlines` ..... 4447
- `\one@line` ..... [747](#), 838, 839, 850
- `\openout` ..... 601, 604, 1892
- `\os` ..... 3635, 3660, 3729, 3739
- `\overfullrule` ..... 3911
- P**
- `\PackageError` ..... 34
- `\PackageWarning` ..... 33
- `\page@action` ..... 393, [491](#), 577
- `\page@num` ..... [345](#),  
356, 431, 442, 582, 587, 904,  
1011, 1086, 1521, 1524, 2090, 4012
- `\page@start` ..... [613](#), 1654
- `\pagelinesep` ..... [27](#), [2601](#), 2610
- `\pageno` ..... 95, 97, [1626](#)
- `\pageparbreak` ..... [37](#), [1071](#)
- `\pageref` ..... [24](#)
- `\pagestyle` ..... 3816, 4452
- `\par@line` . [747](#), 1137, 1138, 1141, 1145

- `\para@footgroup` . . . . . 1421, [1490](#)
  - `\para@footgroupX` . . . . . 2361, [2417](#)
  - `\para@footsetup` . . . . . 1423, [1430](#)
  - `\para@footsetupX` . . . . . 2367, [2369](#)
  - `\para@vfootnote` . . . . . 1419, [1442](#), 4191
  - `\para@vfootnoteX` . . . . . 2359, [2385](#)
  - `\parafootfmt` . . . . . 1420, [1484](#), 3529
  - `\parafootfmtX` . . . . . 2360, [2411](#)
  - `\parafootftmsep` . . . . . [1518](#), 1522
  - `\parafootstart` . . . . . 1418, [1438](#)
  - `\parafootstartX` . . . . . 2358, [2377](#)
  - `\pausenumbering` . . . . . [11](#), [173](#)
  - `\pend` . . . . . 9, 85, 88, 770, [777](#), 828, 1071,  
2775, 2777, 3471, 3587, 3622,  
3677, 3685, 3704, 3717, 3745,  
3829, 3836, 3857, 3892, 3899,  
4084, 4143, 4157, 4458, 4470, 4475
  - `\phantom` . . . . . 2794
  - Pigman, III<sup>rd</sup>, G. W. . . . . 185
  - Plato of Tivoli . . . . . 7
  - `\postbodyfootmark` . . . . . [2168](#), 2182
  - `\postdisplaypenalty` . . . . . 808
  - `\prebodyfootmark` . . . . . [2168](#), 2180
  - `\predisplaypenalty` . . . . . 807
  - `\prevgraf` . . . . . 784
  - `\previous@A@number` . . . . . 4216
  - `\previous@B@number` . . . . . 4217
  - `\previous@C@number` . . . . . 4218
  - `\previous@page` . . . . . 4200, 4205, 4219
  - `\prevpage@num` . . . . . [1519](#)
  - `\printendlines` . . . . . 1922, [1956](#)
  - `\printlines` . . . . .  
. . . . . 1291, [1354](#), 1487, 1568, 1607,  
3499, 3511, 3524, 3531, 3660,  
3789, 3798, 4051, 4059, 4231,  
4239, 4251, 4373, 4378, 4447, 4482
  - `\printnpnum` . . . . . 23, 1958, 1961, [1966](#)
  - `\printstanzalines` . . . . . 4444, 4482
  - `\processl@denbody` . . . . .  
. . . . . 2680, 2684, 2685, 2700
  - `\ProcessOptions` . . . . . 11
  - `\protected@write` . . . . .  
1810, 2634, 2639, 2642, 2657, 2666
  - `\protected@xdef` . . . . .  
2151, 2163, 2435, 2446, 2453, 2462
  - `\ProvidesPackage` . . . . . 3
  - `\pst@rtedLfalse` . . . . . [120](#), 141, 157
  - `\pst@rtedLtrue` . . . . . [120](#), 177
  - `\pstart` . . . . . 9, 79, 82, 83,  
88, [752](#), 827, 1071, 2756, 3458,  
3553, 3589, 3674, 3679, 3683,  
3685, 3686, 3760, 3822, 3832,  
4079, 4087, 4145, 4458, 4466, 4472
  - `\pstartline` . . . . . 785, 786, 788, 791
  - `\pstartnum` . . . . . [1073](#)
  - `\pstartnumfalse` . . . . . 1104, 1111
  - `\pstartnumtrue` . . . . . 800, 1100
- Q**
- `\quad` . . . . . 3872, 3889
- R**
- `\raggedright` . . . . . 1566,  
1605, 2024, 2295, 2340, 3508, 3522
  - `\raw@text` . . . . . [747](#), 775, 787, 838
  - `\rbracket` . . . . . 18, 1292, [1294](#), 1488,  
1569, 1608, 3512, 3784, 3790,  
3799, 4060, 4222, 4245, 4374, 4379
  - `\rd@ta` . . . . . 3951, 4016
  - `\read@linelist` . . . . . [350](#), 597
  - `\ref` . . . . . [24](#), 3483
  - `\refstepcounter` . . . . . 4414
  - `\Relax` . . . . . [2908](#), 3365, 3372
  - `\rem@inder` . . . . . 993, 995–997
  - `\removehboxes` . . . . .  
. . . . . 1493, 1503, [1507](#), 2420, 2429
  - `\removelastskip` . . . . . 3025, 3042
  - `\resumenumbering` . . . . . [11](#), [173](#)
  - `\right` . . . . . 3159, 3162, 3167, 3170
  - `\rightctab` . . . . . [3237](#), 3327
  - `\rightlinenum` . . . . . [12](#),  
[302](#), 1006, 1014, 3644, 3776, 4015
  - `\rightltab` . . . . . [3250](#), 3309
  - `\rightnoteuptrue` . . . . . 2032
  - `\rightpstartnum` . . . . . 1081, 1089, 1106
  - `\rightrtab` . . . . . [3263](#), 3275
  - `\rightstartnum` . . . . . [1073](#)
  - `\rigidbalance` . . . . . [1526](#), 1574, 1581,  
1613, 1624, 2303, 2310, 2348, 2355
  - `\rlap` 1006, 1014, 1081, 1089, 3757, 4015
  - `\rmarpar` . . . . . 4071, 4080, 4121, 4149
  - `\rmfamily` . . . . . 3496, 3516, 3663
  - Robinson, Peter . . . . . 6
  - `\roman` . . . . . 3182
  - `\rtab` . . . . . 2798, [3272](#), 3416
  - `\rtabtext` . . . . . 2801, [3290](#), 3420

## S

- Sacrobosco ..... 8  
 \savel@dcnote ..... 2075  
 \sc@n@list ..... 994, 996  
 \scf ..... 3773,  
     3870, 3875, 3878, 3879, 3886, 3887  
 Schöpf, Rainer ..... 7  
 \section@num .....  
     116, 133, 135, 136, 178–180, 1899  
 \select@lemmafонт ..... 654, 1173  
 \select@lemmafонт ..... 18,  
     1173, 1292, 1488, 1569, 1608,  
     1923, 3500, 3512, 3525, 3532,  
     3784, 3790, 4060, 4245, 4374, 4379  
 \sen ..... 3678,  
     3720, 3721, 3723, 3725, 3730–3735  
 \senspeak ..... 3679, 3719  
 \set@line .... 673, 689, 706, 717, 4429  
 \set@line@action 386, 476, 483, 494, 579  
 \setl@dlp@rbox . 2052, 2056, 2083, 2095  
 \setl@drp@rbox . 2053, 2065, 2085, 2093  
 \setl@drpr@box ..... 2056  
 \setline ... 13, 75, 625, 662, 792, 3840  
 \setlinenum ..... 13, 77, 632, 4416  
 \setlp@rbox ..... 3944, 3949  
 \setmcellcenter ..... 3083, 3139  
 \setmcellleft ..... 3056, 3128  
 \setmcellright ..... 3023, 3117  
 \setmrowcenter ..... 3137, 3331  
 \setmrowleft ..... 3126, 3313  
 \setmrowright ..... 3115, 3279  
 \setprintendlines ..... 1932, 1957  
 \setprintlines ..... 1328, 1366  
 \setstanzaindents .... 20, 2735, 4392  
 \setstanzapenalties .....  
     ..... 21, 2735, 4397, 4398  
 \setstanzavalues 2725, 2735, 2736, 2786  
 \settcclcenter ..... 3098, 3144  
 \settcclleft ..... 3070, 3133  
 \settcclright ..... 3040, 3122  
 \settrrowcenter ..... 3142, 3339  
 \settrrowleft ..... 3131, 3321  
 \settrrowright ..... 3120, 3295  
 \Setzen ..... 3356, 3365, 3367  
 Shakespeare, William ..... 188  
 \showlemma . 16, 17, 22, 24, 31, 678, 694  
 \sidenote@margin ..... 1985, 2088  
 \sidenotemargin ..... 25, 1985, 4176  
 \sidenotemmargin ..... 102  
 \skip ..... 1377, 1384, 1403,  
     1408, 1440, 1498, 1576, 1619,  
     1662, 2211, 2222, 2257, 2263,  
     2305, 2350, 2378, 2382, 2424,  
     2556, 3670, 3671, 4185–4187, 4383  
 \skip@lockoff ..... 513, 539  
 \skipnumbering ..... 14, 642  
 \skipnumbering@reg ..... 642  
 \sl ..... 652  
 \spacedcolon ..... 3496, 3500, 3532  
 \spacefactor .....  
     2108, 2111, 2122, 2128, 2171, 2177  
 \spaceskip ..... 1269, 2190, 2286  
 \speak ..... 3683, 3706  
 \speaker ..... 3760, 3842, 3859, 3894  
 \splitmaxdepth ..... 1274  
 \splitoff ..... 1526  
 \splittopskip .... 835, 1274, 1528,  
     1572, 1574, 1579, 1581, 1611,  
     1613, 1622, 1624, 2301, 2303,  
     2308, 2310, 2346, 2348, 2353, 2355  
 \spreadmath ..... 28, 3344  
 \spreadtext ..... 28, 3342  
 \ss ..... 2789  
 \stage ..... 3674, 3695, 3757, 3763  
 \stanza 20, 2764, 4489, 4492, 4509, 4523  
 \stanza@count ..... 2716, 2727,  
     2730, 2732, 2749, 2761, 2768, 2776  
 \stanza@hang ..... 2747, 2770  
 \stanza@line ..... 2747, 2776, 2778  
 \stanza@modulo .....  
     ..... 2739, 2742–2744, 2753, 2768  
 \stanzaindentbase ..... 20, 2716,  
     2750, 2754, 2759, 2780, 4387, 4411  
 \startlock ..... 13, 639, 657, 2757  
 \startstanzahook ..... 21, 2764, 4414  
 \startsub ..... 12, 615, 656, 3674  
 \stepcounter 1821, 1825, 2150, 2162,  
     2434, 2445, 2452, 2461, 2606, 3190  
 \stepl@dcolcount .... 2869, 2919,  
     2931, 3032, 3049, 3063, 3077,  
     3090, 3105, 3149, 3349, 3358, 3379  
 \strip@pt ..... 1436, 2375  
 \strip@szacnt ..... 2725  
 \sub@action ..... 402, 503, 578  
 \sub@change ..... 150,  
     396, 397, 403, 453, 455, 458, 460  
 \sub@lock ..... 147, 338,  
     411, 413, 415, 418, 521, 522,  
     524, 525, 543, 544, 546, 877,

- 948, 950, 951, 953, 1028, 1064,  
1066, 1068, 3968, 4029, 4031, 4033  
`\sub@off` ..... [452](#), 621  
`\sub@on` ..... [452](#), 617  
`\subline@num` ..... 145, 308,  
309, [335](#), 419, 423, 429, 440,  
464, 465, 467, 479, 497, 584,  
588, 878, 883, 906, 913, 968–  
970, 1847, 3647, 3650, 3957–3959  
`\sublinenumberstyle` ..... [13](#), [293](#)  
`\sublinenumincrement` .... [10](#), [12](#), [253](#)  
`\sublinenumr@p` ..... [293](#)  
`\sublinenumrep` ..... [293](#),  
309, 1369, 1373, 1847, 1959, 1963  
`\sublineref` ..... [23](#), 661, [1855](#)  
`\sublines@false` ... 148, [336](#), 400, 938  
`\sublines@true` ..... [336](#), 398, 936  
`\sublock@disp` ..... [284](#),  
1030, 1034, 1038, 3970, 3974, 3978  
`\sublockdisp` ..... [65](#), [284](#)  
`\subsection` ..... 3455  
Sullivan, Wayne ..... 7, 8, 20,  
37, 43, 48, 103, 104, 118, 149, 191  
`\symplinenum` ..... [31](#), [1308](#), 1368  
`\sza@penalty` ..... [2747](#), 2774, 2775
- T**
- `\tabellzwischen` ..... [3347](#), 3355  
`\tabelskip` 3359, 3399–3401, 3407–3409  
`\tabHilfbox` ..... 3398,  
3400, 3402, 3406, 3408, 3410, [3411](#)  
`\tabhilfbox` ..... 3397,  
3399, 3401, 3405, 3407, 3409, [3411](#)  
`\tableofcontents` ..... 3450  
Tapp, Christian ..... 6  
`\textbf` ..... 4106, 4411, 4458, 4468  
`\textheight` ..... 1776, 4361  
`\textnormal` ..... 1294–1296,  
3825, 3843, 3862, 3872, 3873,  
3881, 3883, 3885, 3889, 3898, 4223  
`\textrm` ..... 3678, 3679, 3870  
`\textsc` ..... 4284  
`\textsf` ..... 3452, 3459  
`\textsuperscript` .....  
.... 2101, 2159, 2185, 2208, 4107  
`\textwidth` ..... 2534, 2576, 4362  
`\thanks` ..... 3446  
`\theadcolcount` .... [3181](#), 3188, 3191  
`\thefootnoteA` [26](#), 2151, [2156](#), 2159, 2163  
`\thefootnoteB` .. 2435, [2439](#), 2446, 3439
- `\thefootnoteC` ..... 2453, [2456](#), 2462  
`\thelabidx` ..... 2607, 2610  
`\theline` ..... 1826, 1828  
`\thempfn` ..... 2536, 2562, 2589  
`\thempfootnote` ..... 2536, 2562, 2589  
Theodosius ..... 8  
`\thepage` ..... 607, 1811, 2610  
`\thepageline` .....  
[2609](#), 2635, 2640, 2643, 2658, 2667  
`\thepstart` ..... [11](#),  
[752](#), 754, 775, 827, 1102, 1109, 1363  
`\thepstartL` ..... 1360  
`\thepstartR` ..... 1358  
`\thestanzanum` ..... 4411  
`\thesubline` ..... 1826  
`\thinspace` ..... 1296, 3496, 3516  
`\thr@@` ..... 235, 524, 533, 544,  
551, 943, 951, 1551, 1554, 1574,  
1581, 2006, 2323, 2326, 2348, 2355  
`\threecolfootfmt` ... 1541, [1561](#), 3503  
`\threecolfootfmtX` ..... 2314, [2334](#)  
`\threecolfootgroup` ..... 1542, [1571](#)  
`\threecolfootgroupX` ..... 2315, [2345](#)  
`\threecolfootsetup` ..... 1543, [1549](#)  
`\threecolfootsetupX` ..... 2316, [2321](#)  
`\threecolvfootnote` ..... 1540, [1556](#)  
`\threecolvfootnoteX` ..... 2313, [2328](#)  
`\tiny` ..... 3773  
`\title` ..... 3445, 4269  
`\tolerance` ..... 1565,  
1604, 2294, 2339, 3507, 3521, 3796  
`\twocolfootfmt` ..... 1585, [1593](#), 3517  
`\twocolfootfmtX` ..... 2268, [2289](#)  
`\twocolfootgroup` ..... 1586, [1593](#)  
`\twocolfootgroupX` ..... 2269, [2300](#)  
`\twocolfootsetup` ..... 1587, [1593](#)  
`\twocolfootsetupX` ..... 2270, [2275](#)  
`\twocolvfootnote` ..... 1584, [1593](#)  
`\twocolvfootnoteX` ..... 2267, [2282](#)
- U**
- `\underbrace` ..... 2796  
`\unhbox` 832, 1473, 1493, 1495, 1503,  
1505, 1512, 1516, 2420, 2422,  
2429, 2431, 3401, 3402, 3409, 3410  
`\unkern` ..... 2109  
`\unpenalty` ..... 1476, 1509  
`\unvbox` ..... 839,  
1279, 1381, 1387, 1458, 1471,  
1491, 1501, 1537, 1641, 1661,

- 1666, 1679, 1693–1697, 1701,  
1703, 1721, 1759, 2195, 2219,  
2225, 2400, 2418, 2427, 2473,  
2480–2482, 2489, 2494, 2553, 2559  
`\unvvh` 1449, 1463, 1470, 2392, 2405, 4208  
`\usepackage` . . . . . 3434, 3490, 3630,  
3754, 3908, 3916, 4171, 4172, 4359  
`\usingcritext` . . . . . 3422  
`\usingdtext` . . . . . 3422
- V**
- `\vAfootnote` . . . . . 1180, 1184  
`\valign` . . . . . 1529  
`\value` . . . . . 2743, 2748, 3187  
Vamana . . . . . 8  
`\variab` . . . . . 2952, 3278,  
3294, 3312, 3320, 3330, 3338, 3371  
`\vbadness` . . . . . 834, 1528  
`\vbfnoteX` . . . . . 2229, 2234  
`\vBfootnote` . . . . . 1188, 1192  
`\vbox` . . . . . 775, 1278,  
1447, 1457, 1461, 1471, 1638,  
1658, 1678, 1784, 1788, 2060,  
2062, 2064, 2069, 2071, 2194,  
2390, 2399, 2403, 2472, 2488,  
2531, 3158, 3161, 3166, 3169,  
3173, 3175, 3176, 3220, 3224,  
3229, 3240, 3253, 3266, 3947, 4206  
`\vCfootnote` . . . . . 1196, 1200  
`\vDfootnote` . . . . . 1204, 1208  
`\vEfootnote` . . . . . 1212, 1216  
`\vffootnote` . . . . . 3928, 3932, 3949  
`\vfil` . . . . . 1529, 1788,  
3159, 3162, 3167, 3170, 3173, 3176  
`\vfootnoteA` . . . . . 2153  
`\vfootnoteB` . . . . . 2437  
`\vfootnoteC` . . . . . 2455  
`\vgfootnote` . . . . . 3937, 3941, 3951  
`\vl@dbfnote` . . . . . 2139  
`\vl@dcsnote` . . . . . 2047, 2052  
`\vl@dlsnote` . . . . . 2035, 2052
- `\vl@drsnote` . . . . . 2041, 2052  
`\vnumfootnoteX` . . . . . 2238, 2250  
`\vrule` . . . . . 3158, 3161, 3166, 3169, 3173  
`\vsize` . . . . . 1389, 2261, 3669  
`\vsplit` . . . . . 838, 1536, 1759
- W**
- `\wd` . . . . . 827, 850, 1451, 1465, 2394,  
2407, 2875, 2876, 3020, 3233,  
3242, 3245, 3255, 3258, 3267,  
3399, 3400, 3407, 3408, 3681, 4210  
Whitney, Ron . . . . . 7  
`\widowpenalty` . . . . . 808, 1143  
Wujastyk, Dominik . . . . . 5, 7
- X**
- `\x@lemma` . . . . . 680–682, 696–698, 4436–4438  
`\xcritext` . . . . . 2879, 3010  
`\xedindex` . . . . . 2885, 2982, 2999, 3012  
`\xedlabel` . . . . . 2883, 3021  
`\xedtext` . . . . . 2879, 3011  
`\xleft@appenditem` . . . . . 327  
`\xlineref` . . . . . 24, 1852  
`\xpageref` . . . . . 24, 1849  
`\xparafootfmt` . . . . . 4371, 4382, 4480  
`\xright@appenditem` . . . . . 321, 492,  
493, 495, 502, 504, 506, 508,  
518, 520, 529, 540, 542, 549,  
562, 563, 572, 586, 1180, 1188,  
1196, 1204, 1212, 1227, 1235,  
1243, 1251, 1259, 1845, 2035,  
2041, 2047, 2141, 2229, 3928, 3937  
`\xspaceskip` . . . . . 1269, 2190, 2286  
`\xsublineref` . . . . . 24, 1855  
`\xxref` . . . . . 24, 1880
- Y**
- `\yparafootfmt` . . . . . 4376, 4464
- Z**
- `\z@skip` . . . . . 1269, 1275, 2190, 2286  
`\zz@@@` . . . . . 1795, 1801, 1882, 1884

## Change History

v0.1	General: First public release . . . . . 1	mand is first element of a paragraph. . . . . 1
v0.10	General: Corrections to <code>\section</code> and other titles in numbered sections . . . . . 1	<code>\edlabel</code> : Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph. . . . . 118
v0.11	General: Makes it possible to add a symbol on each verse's hanging, as in French typography. Redefines the command <code>\hangingsymbol</code> to define the character. . . . . 1	v0.15
v0.12	General: For compatibility with <code>ledpar</code> , possibility to use <code>\autopar</code> on the right side. . . . . 1	General: Line numbering can be reset at each <code>pstart</code> . . . . . 49
	Possibility to number the <code>pstart</code> with the commands <code>\numberpstarttrue</code> . . . . . 1	New management of hanging symbol insertion, preventing undesirable insertions. . . . . 149
	Possibility to number <code>\pstart</code> . 11	Possibility to print <code>\pstart</code> number in side. . . . . 11
<code>\ifledRcol</code> : Added <code>\ifledRcol</code> and <code>\ifnumberingR</code> for/from <code>ledpar</code> . . . . . 47		<code>\affixline@num</code> : Line numbering can be disabled. . . . . 84
v0.12.1	General: Don't number <code>\pstarts</code> of stanza. . . . . 1	<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code> . . . . . 98
	The numbering of <code>\pstarts</code> restarts on each <code>\beginnumbering</code> . . . . . 1	v0.16
v0.13	General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every $n$ verses. . . . . 1, 20	General: <code>\parafootftmsep</code> command is printed between each paragraphed footnote . . . . . 101
	<code>\managestanza@modulo</code> : New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every $n$ verses. . . . . 150	v0.17
v0.13.1	General: <code>\thepstartL</code> and <code>\thepstartR</code> use now <code>\bfseries</code> and not <code>\bf</code> , which is deprecated and makes conflicts with memoir class. . . . . 1	General: New new management of hanging symbol insertion, preventing undesirable insertions. 149
v0.14	General: Tweaked <code>\edlabel</code> to get correct line number if the com-	v0.18
		General: Debug <code>\lineation</code> command. . . . . 49
		v0.19
		General: Add an optional message to suggest to migrate to <code>eledmac</code> . . . . . 44
		v0.19.2a
		General: <code>Reledmac</code> is released. . . . 1
		Suggest now to migrate to <code>reledmac</code> . . . . . 44
		v0.2
		General: Added <code>tabmac</code> code, and extended indexing . . . . . 1
		<code>\ifl@dmemoir</code> : Added <code>\ifl@dmemoir</code> for memoir class having been used . . . . . 43
		<code>\ledmac@error</code> : Added <code>\ledmac@error</code> and replaced error messages . . 44
		<code>\morenoexpands</code> : Added <code>\l@dtabnoexpands</code> to <code>\no@expands</code> . . . . . 72



v0.2.1			
	<code>\@lab</code> : Removed page setting from <code>\@lab</code> . . . . .	120	
	General: Added text about normal labeling . . . . .	24	
	Bug fixes and match with mem-patch v1.8 . . . . .	1	
	Major changes to insert code when memoir is loaded . . . .	115	
	<code>\doxtrafeet</code> : Renamed <code>\doxtrafeet</code> to <code>\l@ddoxtrafeet</code> . . . . .	114	
	<code>\edlabel</code> : Tweaked <code>\edlabel</code> to get correct page numbers . . .	118	
	<code>\l@d@makecol</code> : Rewrote <code>\@makecol</code> , calling it <code>\l@d@makecol</code> . . . .	113	
	<code>\l@ddodoreintrafeet</code> : Renamed <code>\dodoreintrafeet</code> to <code>\l@ddodoreintrafeet</code> . . . .	115	
	<code>\l@ddofootinsert</code> : Renamed <code>\dofootinsert</code> as <code>\l@ddofootinsert</code> . . . . .	114	
	<code>\m@m@makecolintro</code> : Added <code>\m@m@makecolfloats</code> , <code>\m@m@makecoltext</code> and <code>\m@m@makecolintro</code> . . .	113	
	<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 . . . . .	72	
	<code>\zz@@@</code> : Minor change to <code>\zz@@@</code> .	118	
v0.2.2			
	General: Added the Dekker example . . . . .	196	
	Improved paragraph footnotes .	1	
	New Dekker example . . . . .	1	
	<code>\footfudgefiddle</code> : Added <code>\footfudgefiddle</code> . . . . .	101	
	<code>\l@d@section</code> : Used <code>\providecommand</code> for <code>\@gobblethree</code> to avoid clash with the <code>amsfonts</code> package . . . . .	124	
	<code>\line@list@stuff</code> : Added initial write of page number in <code>\line@list@stuff</code> . . . . .	67	
	<code>\para@footsetup</code> : Added <code>\footfudgefiddle</code> to <code>\para@footsetup</code> . . . . .	102	
	<code>\para@footsetupX</code> : Added <code>\footfudgefiddle</code> to		
	<code>\para@footsetupX</code> . . . . .	137	
	<code>\symplinenum</code> : Added <code>\symplinenum</code> . . . . .	96	
v0.3			
	<code>\@l@reg</code> : Added a bunch of code to <code>\@l</code> for handling <code>\setlinenum</code> .	60	
	<code>\@lab</code> : Replaced <code>\the\line@num</code> by <code>\linenumr@p\line@num</code> in <code>\@lab</code> , and similar for sub-lines	120	
	General: Added the Braonain example . . . . .	201	
	Includes <code>edstanza</code> and more . . .	1	
	Two more Dekker examples . .	32	
	<code>\ledlinenum</code> : Added <code>\linenumr@p</code> and <code>\sublinenumr@p</code> to <code>\leftlinenum</code> and <code>\rightlinenum</code> . . . . .	53	
	<code>\linenumberlist</code> : Added <code>\linenumberlist</code> mechanism .	43	
	<code>\printendlines</code> : Added <code>\linenumr@p</code> and <code>\sublinenumr@p</code> to <code>\printendlines</code> . . . . .	125	
	<code>\printlines</code> : Added <code>\linenumr@p</code> and <code>\sublinenumr@p</code> to <code>\printlines</code> . . . . .	98	
	<code>\sublinenumr@p</code> : Added <code>\linenumberstyle</code> and <code>\sublinenumberstyle</code> . . .	52	
v0.3.1			
	General: Not released. Added remarks about the parallel package . . . . .	1	
v0.31			
	General: Added remarks about <code>ledmac/parallel</code> package incompatibility . . . . .	38	
v0.4			
	<code>\@iiiminipage</code> : Modified kernel <code>\@iiiminipage</code> and <code>\endminipage</code> to cater for critical footnotes . . . . .	141	
	General: Added <code>\showlemma</code> to <code>\edtext</code> (and <code>\critext</code> ) . . . .	74	
	Added <code>minipage</code> , etc., support .	1	
	<code>ledgroup</code> : Added <code>ledgroup</code> environment . . . . .	142	
	<code>ledgroupsize</code> : Added <code>ledgroupsize</code> environment . . . . .	143	
	<code>\footnormal</code> : Added <code>minpage</code> footnote setup to <code>\footnormal</code> .	100	

- `\ifledfinal`: Added final/draft options . . . . . 43  
`\l@dfeetendmini`: Added `\l@dfeetbeginmini`, `\l@dfeetendmini` and all their supporting code 141  
`\mpEfootins`: Added `\mpAfootins` and friends . . . . . 93  
`\mpEfootnote`: Added `\mpAfootnote` and friends . . . . . 93  
`\mpfootinsA`: Familiar footnotes extended for minipages . . . . 131  
`\mpfootinsB`: Familiar footnotes extended for minipages . . . . 139  
`\mpfootinsC`: Familiar footnotes extended for minipages . . . . 139  
`\mpnormalfootgroup`: Added `\mpnormalfootgroup` . . . . . 99  
`\mpnormalvfootnote`: Added `\mpnormalvfootnote` . . . . . 95  
v0.4.1  
`\@opxtrafeetii`: Added `\@opxtrafeetii` . . . . . 114  
General: Added code for changing `\@doclearpage` . . . . . 116  
Let ledmac take advantage of memoir's indexing . . . . . 144  
Not released. Minor editorial improvements and code tweaks . . 1  
Only change `\@footnotetext` and `\@footnotemark` if memoir not used . . . . . 129  
`\addfootins`: Added `\addfootins` 116  
`\addfootinsX`: Added minpage setup to `\addfootinsX` . . . . 140  
`\doxtrafeetii`: Changed `\doxtrafeetii` code for easier extensions . . . . . 114  
`\ledfootinsdim`: Added `\ledfootinsdim` . . . . . 100  
v0.5  
`\@footnotetext`: Enabled regular `\footnote` in numbered text 130  
`\@xympar`: Eliminated `\marginpar` disturbance . . . . . 126  
General: Added left and right side notes . . . . . 126  
Added sidenotes, familiar footnotes in numbered text . . . . . 1  
v0.5.1  
General: Added moveable side note 126  
Fixed right line numbers killed in v0.5 . . . . . 1  
`\affixline@num`: Changed `\affixline@num` to cater for sidenotes . . . . . 84  
`ledgroupsize`: Only change `\hsize` in `ledgroupsize` environment otherwise page number can be in wrong place . . . . 143  
`\l@dgetsidenote@margin`: Added `\sidenotemargin` and `\sidenote@margin` . . . . . 126  
v0.6  
`\@l@reg`: Added `\fix@page` to `\@l` 60  
Extended `\@l` to include the page number . . . . . 60  
`\@lopR`: Added `\@pend`, `\@pendR`, `\@lopL` and `\@lopR` in anticipation of parallel processing . . 62  
General: Changed version of the Dekker example . . . . . 196  
Fixed long paragraphs looping . 1  
Fixed minor typos . . . . . 1  
Prepared for ledpar package . . 1  
`\fix@page`: Added `\last@page@num` and `\fix@page` . . . . . 61  
`\footnoteA`: Modified `\footnoteA` and friends to include `\@thefnmarkA` etc . . . . . 131  
`\new@line`: Extended `\new@line` to output page numbers . . . . . 67  
`\page@start`: Made `\page@start` a no-op . . . . . 67  
`\vl@dbfnote`: Changed `\l@dbfnote` and `\vl@dbfnote` as originals could give incorrect markers in the footnotes . . . . . 130  
v0.7  
`\@l@reg`: Added `\@l@reg` . . . . . 60  
`\@ref@reg`: Added `\@ref@reg` . . . 65  
General: Added bits about ledpar package . . . . . 38  
ledmac having been available for 2 years, deleted the commented out original edmac texts . . . . 1  
Maïeul Rouquette new maintainer . . . . . 1  
Made macros of all messages . 44

Replaced all <code>\interfootnotelinepenalty</code> , etc., by just <code>\interfootnotelinepenalty\line@list@stuff</code> : ..... 1	<code>\rightlinenum</code> ..... 53 Deleted <code>\page@start from \line@list@stuff</code> ..... 67
Tidying up for <code>ledpar</code> and <code>ledarab</code> packages ..... 1	<code>\list@clearing@reg</code> : Added <code>\list@clearing@reg</code> ..... 59
<code>\affixline@num</code> : Added skipnum- mering to <code>\affixline@num</code> .. 84	<code>\n@num@reg</code> : Added <code>\n@num</code> .... 65
<code>\do@actions@fixedcode</code> : Added <code>\do@actions@fixedcode</code> .... 83	<code>\normalbfnoteX</code> : Removed extraneous space from <code>\normalbfnoteX</code> ..... 133
<code>\do@actions@next</code> : Added number skipping to <code>\do@actions</code> .... 82	<code>\resumenummering</code> : Changed <code>\resumenummering</code> for <code>ledpar</code> . 48
<code>\do@linehook</code> : Added <code>\do@linehook</code> for use in <code>\do@line</code> ..... 80	<code>\setprintendlines</code> : Added <code>\setprintendlines</code> for use by <code>\printendlines</code> ..... 124
<code>\endnumbering</code> : Changed <code>\endnumbering</code> for <code>ledpar</code> ... 48	<code>\setprintlines</code> : Added <code>\setprintlines</code> for use by <code>\printlines</code> .... 97
<code>\fx@l@cks</code> : Added <code>\ch@cksub@l@ck</code> , <code>\ch@ck@l@ck</code> and <code>\fx@l@cks</code> 86	<code>\skipnumbering@reg</code> : Added <code>\skipnumbering</code> and supports 69
<code>\footsplitskips</code> : Added <code>\footsplitskips</code> for use in many footnote styles ..... 94	<code>\sublinenumincrement</code> : Added <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> and <code>\linenumincrement</code> ..... 51
<code>\get@linelistfile</code> : Added <code>\get@linelistfile</code> ..... 59	<code>\sublinenumr@p</code> : Using <code>\linenumrep</code> instead of <code>\linenumr@p</code> ..... 52 Using <code>\sublinenumrep</code> instead of <code>\sublinenumr@p</code> ..... 52
<code>\ifledRcol</code> : Added <code>\l@dnumpstartsL</code> , <code>\ifl@dpairing</code> and <code>\ifpst@rted</code> for/from <code>ledpar</code> ..... 47	<code>\vnumfootnoteX</code> : Removed extraneous space from <code>\vnumfootnoteX</code> ..... 133
<code>\initnumbering@reg</code> : Added <code>\initnumbering@reg</code> ..... 47	v0.8 General: Bug on endnotes fixed : in a // text, all endnotes will print and be placed at the ends of columns () ..... 1
<code>\l@dcsnotetext</code> : Added <code>\l@demptyd@ta</code> ..... 80	v0.8.1 General: Bug on <code>\edtext</code> ; <code>\critex</code> ; <code>\lemma</code> fixed : we can now us non switching commands ..... 1
<code>\l@ddofootinsert</code> : Deleted <code>\page@start from \l@ddofootinsert</code> ..... 114	v0.9 General: No more <code>ledpatch</code> . All patches are now in the main file. .... 1
<code>\l@dgetline@margin</code> : Added <code>\l@dgetline@margin</code> ..... 50	v0.9.1 General: Fix some bugs linked to in- tegrating <code>ledpatch</code> on the main file. .... 1
<code>\l@dgetlock@disp</code> : Added <code>\l@dgetlock@disp</code> ..... 51	
<code>\l@dgetsidenote@margin</code> : Added <code>\l@dgetsidenote@margin</code> .. 126	
<code>\l@drsn@te</code> : Added <code>\l@dlsn@te</code> and <code>\l@drsn@te</code> for use in <code>\do@line</code> ..... 80	
<code>\l@dunboxmpfoot</code> : Added <code>\l@dunboxmpfoot</code> containing some common code ..... 142	
<code>\l@dzeropenalties</code> : Added <code>\l@dzeropenalties</code> ..... 78	
<code>\ledlinenum</code> : Added <code>\ledlinenum</code> for use by <code>\leftlinenum</code> and	