

Parallel typesetting for critical editions: the **ledpar** package*

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

Abstract

The **ledmac** package, which is based on the PLAIN T_EX set of **EDMAC** macros, has been used for some time for typesetting critical editions. The **ledpar** package is an extension to **ledmac** which enables texts and their critical apparatus to be typeset in parallel, either in two columns or on pairs of facing pages.

Please, for all bug's report, open a ticket on <https://github.com/maieul/ledmac/issues/>

Contents

1	Introduction	3
2	The ledpar package	4
2.1	General	4
3	Parallel columns	5
4	Facing pages	5
5	Left and right texts	6
6	Numbering text lines	8
7	Verse	9
8	Implementation overview	11
9	Preliminaries	11
9.1	Messages	12

*This file (**ledpar.dtx**) has version number v0.7, last revised 2011/08/31.

[†]herries dot press at earthlink dot net

[‡]maieul at maieul dot net

10 Sectioning commands	12
11 Line counting	15
11.1 Choosing the system of lineation	15
11.2 Line-number counters and lists	18
11.3 Reading the line-list file	18
11.4 Commands within the line-list file	19
11.5 Writing to the line-list file	27
12 Marking text for notes	30
13 Parallel environments	31
14 Paragraph decomposition and reassembly	33
14.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	33
14.2 Processing one line	36
14.3 Line and page number computation	38
14.4 Line number printing	40
14.5 Add insertions to the vertical list	42
14.6 Penalties	42
14.7 Printing leftover notes	43
15 Footnotes	44
15.1 Outer-level footnote commands	44
15.2 Normal footnote formatting	47
16 Cross referencing	48
17 Side notes	49
18 Familiar footnotes	51
19 Verse	51
20 Naming macros	52
21 Counts and boxes for parallel texts	53
22 Fixing babel	54
23 Parallel columns	56
24 Parallel pages	59
25 The End	67

<i>List of Figures</i>	3
------------------------	---

A Examples	68
A.1 Parallel column example	76
A.2 Example parallel facing pages	78
A.3 Example poetry on parallel facing pages	84
References	89
Index	89
Change History	97

List of Figures

1	Output from <code>villon.tex</code>	69
2	Left page output from <code>djd17nov.tex</code>	70
3	Right page output from <code>djd17nov.tex</code>	71
4	First left page output from <code>djdpoems.tex</code>	72
5	First right page output from <code>djdpoems.tex</code>	73
6	Second left page output from <code>djdpoems.tex</code>	74
7	Second right page output from <code>djdpoems.tex</code>	75

1 Introduction

The **EDMAC** macros [LW90] for typesetting critical editions of texts have been available for use with TeX for some years. Since **EDMAC** became available there had been a small but constant demand for a version of **EDMAC** that could be used with LaTeX. The **ledmac** package was introduced in 2003 in an attempt to satisfy that request.

Some critical editions contain texts in more than one form, such as a set of verses in one language and their translations in another. In such cases there is a desire to be able to typeset the two texts, together with any critical apparatus, in parallel. The **ledpar** package is an extension to **ledmac** that enables two texts and their apparatus to be set in parallel, either in two columns or on pairs of facing pages.

The package has to try and coerce TeX into paths it was not designed for. Use of the package, therefore, may produce some surprising results.

This manual contains a general description of how to use **ledpar** starting in section 2; the complete source code for the package, with extensive documentation (in sections 8 through 25); and an Index to the source code. As **ledpar** is an adjunct to **ledmac** I assume that you have read the **ledmac** manual. Also **ledpar** requires **ledmac** to be used, preferably at least version 0.10 (2011/08/22). You do not need to read the source code for this package in order to use it but doing so may help to answer any questions you might have. On a first reading, I suggest that you should skip anything after the general documentation in sections 2 until 8, unless you are particularly interested in the innards of **ledpar**.

2 The *ledpar* package

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.

The *ledpar* package lets you typeset two *numbered* texts in parallel. This can be done either as setting the 'Leftside' and 'Rightside' texts in two columns or on facing pages. In the paired pages case footnotes are placed at the bottom of the page on which they are called out — that is, footnotes belonging to the left are set at the foot of a left (even numbered) page, and those for right texts are at the bottom of the relevant right (odd numbered) page. However, in the columnar case, all footnotes are set at the bottom left of the page on which they are called out — they are not set below the relevant column. The line numbering schemes need not be the same for the two texts.

2.1 General

ledmac essentially puts each chunk of numbered text (the text within a `\pstart` ... `\pend`) into a box and then following the `\pend` extracts the text line by line from the box to number and print it. More precisely, the text is first put into the box as though it was being typeset as normal onto a page and any notes are stored without being typeset. Then each typeset line is extracted from the box and any notes for that line are recalled. The line, with any notes, is then output for printing, possibly with a line number attached. Effectively, all the text is typeset and then afterwards all the notes are typeset.

ledpar similarly puts the left and right chunks into boxes but can't immediately output the text after a `\pend` — it has to wait until after both the left and right texts have been collected before it can start processing. This means that several boxes are required and possibly TeX has to store a lot of text in its memory; both the number of potential boxes and memory are limited. If TeX's memory is overfilled the recourse is to reduce the amount of text stored before printing.

`\maxchunks` It is possible to have multiple chunks in the left and right texts before printing them. The macro `\maxchunks{<num>}` specifies the maximum number of chunks within the left or right texts. This is initially set as:

`\maxchunks{10}`

meaning that there can be up to 10 chunks in the left text and up to 10 chunks in the right text, requiring a total of 20 boxes. If you need more chunks then you can increase `\maxchunks`.

TeX has a limited number of boxes; if you get an error message along the lines of 'no room for a new box', then decrease the number. A chunk also requires a counter so you may get a message along the lines 'no room for a new count', which may be resolved by reducing `\maxchunks`.

On the other hand, if you get a *ledmac* error message along the lines: 'Too

many `\pstart` without printing. Some text will be lost.’ then you will have to either increase `\maxchunks` or use the parallel printing commands (`\Columns` or `\Pages`) more frequently.

When typesetting verse using `\syntax`, each line is treated as a chunk, so be warned that if you are setting parallel verses you might have to increase `\maxchunks` much more than it appears at first sight.

In general, `ledmac` is a TeX resource hog, and `ledpar` only makes things worse in this respect.

3 Parallel columns

pairs Numbered text that is to be set in columns must be within a **pairs** environment. Within the environment the text for the lefthand and righthand columns is placed within the **Leftside** and **Rightside** environments, respectively; these are described in more detail below in section 5.

\Columns The command `\Columns` typesets the texts in the previous pair of **Leftside** and **Rightside** environments. The general scheme for parallel columns looks like this:

```
\begin{pairs}
\begin{Leftside} ... \end{Leftside}
\begin{Rightside} ... \end{Rightside}
\Columns
\begin{Leftside} ... \end{Leftside}
...
\Columns
\end{pairs}
```

There is no required pagebreak before or after the columns.

\Lcolwidth The lengths `\Lcolwidth` and `\Rcolwidth` are the widths of the left and right columns, respectively. By default, these are:

```
\setlength{\Lcolwidth}{0.45\textwidth}
\setlength{\Rcolwidth}{0.45\textwidth}
```

They may be adjusted if one text tends to be ‘bulkier’ than the other.

\columnrulewidth The macro `\columnseparator` is called between each left/right pair of lines. By default it inserts a vertical rule of width `\columnrulewidth`. As this is initially defined to be 0pt the rule is invisible. For a visible rule between the columns you could try:

```
\setlength{\columnrulewidth}{0.4pt}
```

You can also modify `\columnseparator` if you want more control.

4 Facing pages

pages Numbered text that is to be set on facing pages must be within a **pages** environ-

ment. Within the environment the text for the lefthand and righthand pages is placed within the **Leftside** and **Rightside** environments, respectively.

\Pages The command **\Pages** typesets the texts in the previous pair of **Leftside** and **Rightside** environments. The general scheme for parallel pages looks like this:

```
\begin{pages}
\begin{Leftside} ... \end{Leftside}
\begin{Rightside} ... \end{Rightside}
\Pages
\begin{Leftside} ... \end{Leftside}
...
\Pages
\end{pages}
```

The **Leftside** text is set on lefthand (even numbered) pages and the **Rightside** text is set on righthand (odd numbered) pages. Each **\Pages** command starts a new even numbered page. After parallel typesetting is finished, a new page is started.

\Lcolwidth Within the **pages** environment the lengths **\Lcolwidth** and **\Rcolwidth** are the widths of the left and right pages, respectively. By default, these are set to the normal textwidth for the document, but can be changed within the environment if necessary.

\goalfraction When doing parallel pages **ledpar** has to guess where TeX is going to put pagebreaks and hopefully get there first in order to put the pair of texts on their proper pages. When it thinks that the fraction **\goalfraction** of a page has been filled, it finishes that page and starts on the other side's text. The definition is:

```
\newcommand*{\goalfraction}{0.9}
```

If you think you can get more on a page, increase this. On the other hand, if some left text overflows onto an odd numbered page or some right text onto an even page, try reducing it, for instance by:

```
\renewcommand*{\goalfraction}{0.8}
```

5 Left and right texts

Parallel texts are divided into **Leftside** and **Rightside**. The form of the contents of these two are independent of whether they will be set in columns or pages.

Leftside The left text is put within the **Leftside** environment and the right text likewise in the **Rightside** environment. The number of **Leftside** and **Rightside** environments must be the same.

\firstlinenum Within these environments you can designate the line numbering scheme(s) to be used. The **ledmac** package originally used counters for specifying the numbering scheme; now both **ledmac**¹ and the **ledpar** package use macros instead. **\linenumincrement** Following **\firstlinenum{<num>}** the first line number will be *<num>*, and following **\sublinenumincrement{<num>}** only every *<num>*th line will have a printed

¹when used with **ledpatch** v0.2 or greater.

number. Using these macros inside the `Leftside` and `Rightside` environments gives you independent control over the left and right numbering schemes. The `\firstsublinenum` and `\sublinenumincrement` macros correspondingly set the numbering scheme for sublines.

`\pstart` In a serial (non-parallel) mode, each numbered paragraph, or chunk, is contained between the `\pstart` and `\pend` macros, and the paragraph is output when the `\pend` macro occurs. The situation is somewhat different with parallel typesetting as the left text (contained within `\pstart` and `\pend` groups within the `Leftside` environment) has to be set in parallel with the right text (contained within its own `\pstart` and `\pend` groups within the corresponding `Rightside` environment) the `\pend` macros cannot immediately initiate any typesetting — this has to be controlled by the `\Columns` or `\Pages` macros. Several chunks may be specified within a `Leftside` or `Rightside` environment. A multi-chunk text then looks like:

```
\begin{...side}
  % \beginnumbering
  \pstart first chunk \pend
  \pstart second chunk \pend
  ...
  \pstart last chunk \pend
  % \endnumbering
\end{...side}
```

Numbering, via `\beginnumbering` and `\endnumbering`, may extend across several `Leftside` or `Rightside` environments. Remember, though, that the Left/Right sides are effectively independent of each other.

Generally speaking, controls like `\firstlinenum` or `\linenummargin` apply to sequential and left texts. To effect right texts only they have to be within a `Rightside` environment.

If you are using the `babel` package with different languages (via, say, `\selectlanguage`) for the left and right texts it is particularly important to select the appropriate language within the `Leftside` and `Rightside` environments. The initial language selected for the right text is the `babel` package's default. Also, it is the *last* `\selectlanguage` in a side that controls the language used in any notes for that side when they get printed. If you are using multilingual notes then it is probably safest to explicitly specify the language(s) for each note rather than relying on the language selection for the side. The right side language is also applied to the right side line numbers.

Corresponding left and right sides must have the same number of paragraph chunks — if there are four on the left there must be four on the right, even if some are empty. The start of each pair of left and right chunks are aligned horizontally on the page. The ends may come at different positions — if one chunk is shorter than the other then blank lines are output on the shorter side until the end of the longer chunk is reached.

6 Numbering text lines

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

```
\beginnumbering
<text>
\endnumbering
```

These have to be separately specified within `Leftside` and `Rightside` environments.

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. Separate auxiliary files are maintained for right hand texts and these are named `<jobname>.nnR`, using the ‘R’ to distinguish them from the left hand and serial (non-parallel) texts.

`\memorydump` The command `\memorydump` effectively performs an `\endnumbering` immediately followed by a `\beginnumbering` while not restarting the numbering sequence. This has the effect of clearing TeX’s memory of previous texts and any associated notes, allowing longer apparent streams of parallel texts. The command should be applied to both left and right texts, and after making sure that all previous notes have been output. For example, along the lines of:

```
\begin{Leftside}
  \beginnumbering
  ...
\end{Leftside}
\begin{Rightside}
  \beginnumbering
  ...
\end{Rightside}
\Pages
\begin{Leftside}
  \memorydump
  ...
\end{Leftside}
\begin{Rightside}
  \memorydump
  ...
```

`\Rlineflag` The value of `\Rlineflag` is appended to the line numbers of the right texts. Its default definition is:

```
\newcommand*{\Rlineflag}{R}
```

This may be useful for parallel columns but for parallel pages it might be more appropriate to redefine it as:

```
\renewcommand*{\Rlineflag}{.}
```

`\printlinesR` The `\printlines` macro is ordinarily used to print the line number referred to by `\ledsavedprintlines`.

ences for critical footnotes. For footnotes from right side texts a special version is supplied, called `\printlinesR`, which incorporates `\Rlineflag`. (The macro `\ledsavedprintlines` is a copy of the original `\printlines`, just in case ...). As provided, the package makes no use of `\printlinesR` but you may find it useful. For example, if you only use the B footnote series in righthand texts then you may wish to flag any line numbers in those footnotes with the value of `\Rlineflag`. You could do this by putting the following code in your preamble:

```
\let\oldBfootfmt\Bfootfmt
\renewcommand{\Bfootfmt}[3]{%
  \let\printlines\printlinesR
  \oldBfootfmt{#1}{#2}{#3}}
```

7 Verse

If you are typesetting verse with `ledmac` you can use the `\stanza` construct, and you can also use this in right or left parallel texts. In this case each verse line is a chunk which has two implications. (1) you can unexpectedly exceed the `\maxchunks` limit or the overall limit on the number of boxes, and (2) left and right verse lines are matched, which may not be desirable if one side requires more print lines for verse lines than the other does.

`astanza` `ledpar` provides an `astanza` environment which you can use instead of `\stanza` (simply replace `\stanza` by `\begin{astanza}` and add `\end{astanza}` after the ending `\&`). Within the `astanza` environment each verse line is treated as a paragraph, so there must be no blank lines in the environment otherwise there will be some extraneous vertical spacing.

If you get an error message along the lines of ‘Missing number, treated as zero `\sza000`’ it is because you have forgotten to use `\setstanzaindents` to set the stanza indents.

`\skipnumbering` The command `\skipnumbering` when inserted in a line of parallel text causes the numbering of that particular line to be skipped. This can be useful if you are putting some kind of marker (even if it is only a blank line) between stanzas. Remember, parallel texts must be numbered and this provides a way to slip in an ‘unnumbered’ line.

The `astanza` environment forms a chunk but you may want to have more than one stanza within the chunk. Here are a couple of ways of doing that with a blank line between each internal stanza, and with each stanza numbered. First some preliminary definitions:

```
\newcommand*{\stanzanum}[2][\stanzaindentbase]{%
  \hskip -#1\llap{\textbf{#2}}\hskip #1\ignorespaces}
\newcommand{\interstanza}{\par\mbox{} \skipnumbering}
```

And now for two stanzas in one. In this first example the line numbering repeats for each stanza.

```

\setstanzaindents{1,0,1,0,1,0,1,0,1,0,1}
\begin{pairs}
\begin{Leftside}
\firstlinenum{2}
\linenumincrement{1}
\beginnumbering
\begin{astanza}
\stanzanum{1} First in first stanza &
                Second in first stanza &
                Second in first stanza &
                Third in first stanza &
                Fourth in first stanza &

\interstanza
\setline{2}\stanzanum{2} First in second stanza &
                Second in second stanza &
                Second in second stanza &
                Third in second stanza &
                Fourth in second stanza \&

\end{astanza}
...

```

And here is a slightly different way of doing the same thing, but with the line numbering being continuous.

```

\setstanzaindents{1,0,1,0,1,0,0,1,0,1,0,1}
\begin{pairs}
\begin{Leftside}
\firstlinenum{2}
\linenumincrement{1}
\beginnumbering
\begin{astanza}
\stanzanum{1} First in first stanza &
                Second in first stanza &
                Second in first stanza &
                Third in first stanza &
                Fourth in first stanza &

\strut &
\stanzanum{2}\advanceline{-1} First in second stanza &
                Second in second stanza &
                Second in second stanza &
                Third in second stanza &
                Fourth in second stanza \&

\end{astanza}
...

```

8 Implementation overview

TeX is designed to process a single stream of text, which may include footnotes, tables, and so on. It just keeps converting its input into a stream typeset pages. It was not designed for typesetting two texts in parallel, where it has to alternate from one to the other. Further, TeX essentially processes its input one paragraph at a time — it is very difficult to get at the ‘internals’ of a paragraph such as the individual lines in case you want to number them or put some mark at the start or end of the lines.

`ledmac` solves the problem of line numbering by putting the paragraph in typeset form into a box, and then extracting the lines one by one from the box for TeX to put them onto the page with the appropriate page breaks. Most of the `ledmac` code is concerned with handling this box and its contents.

`ledpar`’s solution to the problem of parallel texts is to put the two texts into separate boxes, and then appropriately extract the pairs of lines from the boxes. This involves duplicating much of the original box code for an extra right text box. The other, smaller, part of the code is concerned with coordinating the line extractions from the boxes.

The package code is presented in roughly in the same order as in `ledmac`.

9 Preliminaries

Announce the name and version of the package, which is targetted for LaTeX2e. The package also requires the `ledmac` package, preferably at least version 0.10 (2011/08/22).

```

1 (*code)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{ledpar}[2011/08/31 v0.7 ledmac extension for parallel texts]
4
```

With the option ‘`shiftedverses`’ a long verse on the left side (or in the right side) don’t make a blank on the corresponding verse, but the blank is put on the bottom of the page. Consequently, the verses on the parallel pages are shifted, but the shifted stop at every end of pages.

```

5 \newif\ifshiftedverses
6 \shiftedversesfalse
7 \DeclareOption{shiftedverses}{\shiftedversestrue}
8 \ProcessOptions
```

As noted above, much of the code is a duplication of the original `ledmac` code to handle the extra box(es) for the right hand side text, and sometimes for the left hand side as well. In order to distinguish I use ‘R’ or ‘L’ in the names of macros for the right and left code. The specifics of ‘L’ and ‘R’ are normally hidden from the user by letting the `Leftside` and `Rightside` environments set things up appropriately.

```

\ifl@dpairing \ifl@dpairing is set TRUE if we are processing parallel texts and \ifl@dpaging
\ifl@dpaging is also set TRUE if we are doing parallel pages. \ifledRcol is set TRUE if we
\ifledRcol are doing the right hand text. \ifl@dpairing is defined in ledmac.
9 \l@dpairingfalse
10 \newif\ifl@dpaging
11 \l@dpagingfalse
12 \newif\ifledRcol
13 \ledRcolfalse

\lcolwidth The widths of the left and right parallel columns (or pages).
\Rcolwidth
14 \newdimen\lcolwidth
15 \lcolwidth=0.45\textwidth
16 \newdimen\Rcolwidth
17 \Rcolwidth=0.45\textwidth
18

```

9.1 Messages

All the error and warning messages are collected here as macros.

```

\led@err@TooManyPstarts
19 \newcommand*{\led@err@TooManyPstarts}{%
20 \ledmac@error{Too many \string\pstart\space without printing.
21 \quad Some text will be lost}{\@ehc}}

\led@err@BadLeftRightPstarts
22 \newcommand*{\led@err@BadLeftRightPstarts}[2]{%
23 \ledmac@error{The numbers of left (#1) and right (#2)
24 \string\pstart s do not match}{\@ehc}}

\led@err@LeftOnRightPage
\led@err@RightOnLeftPage
25 \newcommand*{\led@err@LeftOnRightPage}{%
26 \ledmac@error{The left page has ended on a right page}{\@ehc}}
27 \newcommand*{\led@err@RightOnLeftPage}{%
28 \ledmac@error{The right page has ended on a left page}{\@ehc}}

```

10 Sectioning commands

```

\section@numR This is the right side equivalent of \section@num.
Each section will read and write an associated ‘line-list file’, containing information used to do the numbering. Normally the file will be called <jobname>.nn, where nn is the section number. However, for right side texts the file is called <jobname>.nnR. The \extensionchars applies to the right side files just as it does to the normal files.
29 \newcount\section@numR
30 \section@numR=\z@

```

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

```

31 \newif\ifnumberingR

```

`\ifpst@rtedL` `\ifpst@rtedL` is set `FALSE` at the start of left side numbering, and similarly for `\ifpst@rtedR`. `\ifpst@rtedL` is defined in `ledmac`.

```

32 \pst@rtedLfalse
33 \newif\ifpst@rtedR
34 \pst@rtedRfalse
35

```

`\beginnumbering` For parallel processing the original `\beginnumbering` is extended to zero `\l@dnumpstartsL` — the number of chunks to be processed. It also sets `\ifpst@rtedL` to `FALSE`.

```

36 \providecommand*{\beginnumbering}{%
37   \ifnumbering
38     \led@err@NumberingStarted
39   \endnumbering
40 \fi
41 \global\l@dnumpstartsL \z@
42 \global\pst@rtedLfalse
43 \global\numberingtrue
44 \global\advance\section@num \@ne
45 \initnumbering@reg
46 \message{Section \the\section@num}%
47 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
48 \l@dend@stuff}

```

`\beginnumberingR` This is the right text equivalent of `\beginnumbering`, and begins a section of numbered text.

```

49 \newcommand*{\beginnumberingR}{%
50   \ifnumberingR
51     \led@err@NumberingStarted
52   \endnumberingR
53 \fi
54 \global\l@dnumpstartsR \z@
55 \global\pst@rtedRfalse
56 \global\numberingRtrue
57 \global\advance\section@numR \@ne
58 \global\absline@numR \z@
59 \global\line@numR \z@
60 \global\@lockR \z@
61 \global\sub@lockR \z@
62 \global\sublines@false
63 \global\let\next@page@numR\relax
64 \global\let\sub@change\relax
65 \message{Section \the\section@numR R}%
66 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
67 \l@dend@stuff}
68

```

- `\endnumbering` This is the left text version of the regular `\endnumbering` and must follow the last text for a left text numbered section. It sets `\ifpst@rtedL` to FALSE. It is fully defined in `ledmac`.
- `\endnumberingR` This is the right text equivalent of `\endnumbering` and must follow the last text for a right text numbered section.
- ```

69 \def\endnumberingR{%
70 \ifnumberingR
71 \global\numberingRfalse
72 \normal@pars
73 \ifl@dpairing
74 \global\pst@rtedRfalse
75 \else
76 \ifx\insertlines@listR\empty\else
77 \global\noteschanged@true
78 \fi
79 \ifx\line@listR\empty\else
80 \global\noteschanged@true
81 \fi
82 \fi
83 \ifnoteschanged@
84 \led@mess@NotesChanged
85 \fi
86 \else
87 \led@err@NumberingNotStarted
88 \fi}
89

```
- `\pausenumberingR` These are the right text equivalents of `\pausenumbering` and `\resumenumbering`.
- `\resumenumberingR`
- ```

90 \newcommand*{\pausenumberingR}{%
91   \endnumberingR\global\numberingRtrue}
92 \newcommand*{\resumenumberingR}{%
93   \ifnumberingR
94     \global\pst@rtedRtrue
95     \global\advance\section@numR \@ne
96     \led@mess@SectionContinued{\the\section@numR R}%
97     \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
98     \l@dend@stuff
99   \else
100     \led@err@numberingShouldHaveStarted
101     \endnumberingR
102     \beginnumberingR
103   \fi}
104

```
- `\memorydumpL` `\memorydump` is a shorthand for `\pausenumbering\resumenumbering`. This will clear the memorised stuff for the previous chunks while keeping the numbering going.
- `\memorydumpR`
- ```

105 \newcommand*{\memorydumpL}{%

```

```

106 \endnumbering
107 \numberingtrue
108 \global\pst@rtedLtrue
109 \global\advance\section@num \@ne
110 \led@mess@SectionContinued{\the\section@num}%
111 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
112 \l@dend@stuff}
113 \newcommand*{\memorydumpR}{%
114 \endnumberingR
115 \numberingRtrue
116 \global\pst@rtedRtrue
117 \global\advance\section@numR \@ne
118 \led@mess@SectionContinued{\the\section@numR R}%
119 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
120 \l@dend@stuff}
121

```

## 11 Line counting

### 11.1 Choosing the system of lineation

Sometimes you want line numbers that start at 1 at the top of each page; other times you want line numbers that start at 1 at the start of each section and increase regardless of page breaks. `ledpar` lets you choose different schemes for the left and right texts.

`\ifbypage@R` The `\ifbypage@R` flag specifies the current lineation system for right texts: `false` for line-of-section, `true` for line-of-page. `ledpar` will use the line-of-section system unless instructed otherwise.

```

122 \newif\ifbypage@R
123 \bypage@Rfalse

```

`\lineationR` `\lineationR{<word>}` is the macro used to select the lineation system for right texts. Its argument is a string: either `page` or `section`.

```

124 \newcommand*{\lineationR}[1]{%
125 \ifnumberingR
126 \led@err@LineationInNumbered
127 \else
128 \def\@tempa{#1}\def\@tempb{page}%
129 \ifx\@tempa\@tempb
130 \global\bypage@Rtrue
131 \else
132 \def\@tempb{section}%
133 \ifx\@tempa\@tempb
134 \global\bypage@Rfalse
135 \else
136 \led@warn@BadLineation
137 \fi

```

```

138 \fi
139 \fi}}
140

```

`\linenummargin` You call `\linenummargin{<word>}` to specify which margin you want your right text's 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. 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.

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

```

141 \newcount\line@marginR
142 \renewcommand*{\linenummargin}[1]{%
143 \l@getline@margin{#1}%
144 \ifnum\l@dttempcntb>\m@ne
145 \ifledRcol
146 \global\line@marginR=\l@dttempcntb
147 \else
148 \global\line@margin=\l@dttempcntb
149 \fi
150 \fi}}

```

By default put right text numbers at the right.

```

151 \line@marginR=\@ne
152

```

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

```

153 \newcounter{firstlinenumR}
154 \setcounter{firstlinenumR}{5}
155 \newcounter{linenumincrementR}
156 \setcounter{linenumincrementR}{5}

```

`\c@firstsublinenumR` The following parameters are just like `firstlinenumR` and `linenumincrementR`, but for sub-line numbers. `sublinenumincrementR` must be at least 1.

```

157 \newcounter{firstsublinenumR}
158 \setcounter{firstsublinenumR}{5}
159 \newcounter{sublinenumincrementR}
160 \setcounter{sublinenumincrementR}{5}
161

```

`\firstlinenum` These are the user's macros for changing (sub) line numbers. They are defined in `ledmac v0.7`, but just in case I have started by `\provide`ing them.

```

\firstlinenum
\linenumincrement
\firstsublinenum
\sublinenumincrement
162 \providecommand*{\firstlinenum}{ }

```



```

163 \providecommand*\linenumincrement{}
164 \providecommand*\firstsublinenum{}
165 \providecommand*\sublinenumincrement{}
166 \renewcommand*\firstlinenum[1]{%
167 \ifledRcol \setcounter{firstlinenumR}{#1}%
168 \else \setcounter{firstlinenum}{#1}%
169 \fi}
170 \renewcommand*\linenumincrement[1]{%
171 \ifledRcol \setcounter{linenumincrementR}{#1}%
172 \else \setcounter{linenumincrement}{#1}%
173 \fi}
174 \renewcommand*\firstsublinenum[1]{%
175 \ifledRcol \setcounter{firstsublinenumR}{#1}%
176 \else \setcounter{firstsublinenum}{#1}%
177 \fi}
178 \renewcommand*\sublinenumincrement[1]{%
179 \ifledRcol \setcounter{sublinenumincrementR}{#1}%
180 \else \setcounter{sublinenumincrement}{#1}%
181 \fi}
182

```

`\Rlineflag` This is appended to the line numbers of right text.

```

183 \newcommand*\Rlineflag{R}
184

```

`\linenumrepR` `\linenumrepR{<ctr>}` typesets the right line number `<ctr>`, and similarly `\sublinenumrepR` for subline numbers.

```

185 \newcommand*\linenumrepR[1]{\@arabic{#1}}
186 \newcommand*\sublinenumrepR[1]{\@arabic{#1}}
187

```

`\leftlinenumR` `\leftlinenumR` and `\rightlinenumR` are the macros that are called to print the right text's marginal line numbers. Much of the code for these is common and is maintained in `\l@dlinenumR`.

```

188 \newcommand*\leftlinenumR{%
189 \l@dlinenumR
190 \kern\linenumsep}
191 \newcommand*\rightlinenumR{%
192 \kern\linenumsep
193 \l@dlinenumR}
194 \newcommand*\l@dlinenumR{%
195 \numlabfont\linenumrepR{\line@numR}\Rlineflag%
196 \ifsublines@
197 \ifnum\subline@num>\z@
198 \unskip\fullstop\sublinenumrepR{\subline@numR}%
199 \fi
200 \fi}
201

```

## 11.2 Line-number counters and lists

We need another set of counters and lists for the right text, corresponding to those in `ledmac` for regular or left text.

`\line@numR` The count `\line@numR` stores the line number that's used in the right text's marginal line numbering and in notes. The count `\subline@numR` stores a sub-line number that qualifies `\line@numR`. The count `\absline@numR` stores the absolute number of lines since the start of the right text section: that is, the number we've actually printed, no matter what numbers we attached to them.

```
202 \newcount\line@numR
203 \newcount\subline@numR
204 \newcount\absline@numR
205
```

`\line@listR` Now we can define the list macros that will be created from the line-list file. They are directly analogous to the left text ones. The full list of action codes and their meanings is given in the `ledmac` manual.

`\actions@listR` Here are the commands to create these lists:

```
206 \list@create{\line@listR}
207 \list@create{\insertlines@listR}
208 \list@create{\actionlines@listR}
209 \list@create{\actions@listR}
210
```

`\linesinpar@listL` In order to synchronise left and right chunks in parallel processing we need to know how many lines are in each left and right text chunk, and the maximum of these for each pair of chunks.

```
211 \list@create{\linesinpar@listL}
212 \list@create{\linesinpar@listR}
213 \list@create{\maxlinesinpar@list}
214
```

`\page@numR` The right text page number.

```
215 \newcount\page@numR
216
```

## 11.3 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.

```
217 \renewcommand*{\read@linelist}[1]{%
```

We do different things depending whether or not we are processing right text

```
218 \ifledRcol
219 \list@clear{\line@listR}%
220 \list@clear{\insertlines@listR}%
```

```

221 \list@clear{\actionlines@listR}%
222 \list@clear{\actions@listR}%
223 \list@clear{\linesinpar@listR}%
224 \list@clear{\linesonpage@listR}
225 \else
226 \list@clearing@reg
227 \list@clear{\linesinpar@listL}%
228 \list@clear{\linesonpage@listL}%
229 \fi

```

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

```
230 \list@clear{\maxlinesinpar@list}
```

Now get the file and interpret it.

```

231 \get@linelistfile{#1}%
232 \endgroup

```

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.

```

233 \ifledRcol
234 \global\page@numR=\m@ne
235 \ifx\actionlines@listR\empty
236 \gdef\next@actionlineR{1000000}%
237 \else
238 \gl@p\actionlines@listR\to\next@actionlineR
239 \gl@p\actions@listR\to\next@actionR
240 \fi
241 \else
242 \global\page@num=\m@ne
243 \ifx\actionlines@list\empty
244 \gdef\next@actionline{1000000}%
245 \else
246 \gl@p\actionlines@list\to\next@actionline
247 \gl@p\actions@list\to\next@action
248 \fi
249 \fi}
250

```

This version of `\read@linelist` creates list macros containing data for the entire section, so they could get rather large. The `\memorydump` macro is available if you run into macro memory limitations.

## 11.4 Commands within the line-list file

This section defines the commands that can appear within a line-list file, except for `\@lab` which is in a later section among the cross-referencing commands it is associated with.

The macros with `action` in their names contain all the code that modifies the action-code list.

`\@l@regR` `\@l` does everything related to the start of a new line of numbered text. Exactly `\@l` what it does depends on whether right text is being processed.

```

251 \newcommand{\@l@regR}{%
252 \ifx\l@dchset@num\relax \else
253 \advance\absline@numR \@ne
254 \set@line@action
255 \let\l@dchset@num\relax
256 \advance\absline@numR \m@ne
257 \advance\line@numR \m@ne% % do we need this?
258 \fi
259 \advance\absline@numR \@ne
260 \ifx\next@page@numR\relax \else
261 \page@action
262 \let\next@page@numR\relax
263 \fi
264 \ifx\sub@change\relax \else
265 \ifnum\sub@change>\z@
266 \sublines@true
267 \else
268 \sublines@false
269 \fi
270 \sub@action
271 \let\sub@change\relax
272 \fi
273 \ifcase\@lockR
274 \or
275 \@lockR \tw@
276 \or\or
277 \@lockR \z@
278 \fi
279 \ifcase\sub@lockR
280 \or
281 \sub@lockR \tw@
282 \or\or
283 \sub@lockR \z@
284 \fi
285 \ifsublines@
286 \ifnum\sub@lockR<\tw@
287 \advance\subline@numR \@ne
288 \fi
289 \else
290 \ifnum\@lockR<\tw@
291 \advance\line@numR \@ne \subline@numR \z@
292 \fi
293 \fi}
294
295 \renewcommand*{\@l}[2]{%
```

```

296 \fix@page{#1}%
297 \ifledRcol
298 \@l@regR
299 \else
300 \@l@reg
301 \fi}
302

```

\last@page@numR We have to adjust \fix@page to handle parallel texts.

```

\fix@page
303 \newcount\last@page@numR
304 \last@page@numR=-10000
305 \renewcommand*{\fix@page}[1]{%
306 \ifledRcol
307 \ifnum #1=\last@page@numR
308 \else
309 \ifbypage@R
310 \line@numR \z@ \subline@numR \z@
311 \fi
312 \page@numR=#1\relax
313 \last@page@numR=#1\relax
314 \def\next@page@numR{#1}%
315 \fi
316 \else
317 \ifnum #1=\last@page@num
318 \else
319 \ifbypage@
320 \line@num \z@ \subline@num \z@
321 \fi
322 \page@num=#1\relax
323 \last@page@num=#1\relax
324 \def\next@page@num{#1}%
325 \fi
326 \fi}
327

```

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

```

328 \renewcommand*{\@adv}[1]{%
329 \ifsublines@
330 \ifledRcol
331 \advance\subline@numR by #1\relax
332 \ifnum\subline@numR<\z@
333 \led@warn@BadAdvancelineSubline
334 \subline@numR \z@
335 \fi
336 \else
337 \advance\subline@num by #1\relax
338 \ifnum\subline@num<\z@
339 \led@warn@BadAdvancelineSubline

```

```

340 \subline@num \z@
341 \fi
342 \fi
343 \else
344 \ifledRcol
345 \advance\line@numR by #1\relax
346 \ifnum\line@numR<\z@
347 \led@warn@BadAdvancelineLine
348 \line@numR \z@
349 \fi
350 \else
351 \advance\line@num by #1\relax
352 \ifnum\line@num<\z@
353 \led@warn@BadAdvancelineLine
354 \line@num \z@
355 \fi
356 \fi
357 \fi
358 \set@line@action}
359

```

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

```

360 \renewcommand*{\@set}[1]{%
361 \ifledRcol
362 \ifsublines@
363 \subline@numR=#1\relax
364 \else
365 \line@numR=#1\relax
366 \fi
367 \set@line@action
368 \else
369 \ifsublines@
370 \subline@num=#1\relax
371 \else
372 \line@num=#1\relax
373 \fi
374 \set@line@action
375 \fi}
376

```

`\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.

```

377 \renewcommand*{\l@d@set}[1]{%
378 \ifledRcol
379 \line@numR=#1\relax
380 \advance\line@numR \@ne

```

```

381 \def\l@dchset@num{#1}
382 \else
383 \line@num=#1\relax
384 \advance\line@num \@ne
385 \def\l@dchset@num{#1}
386 \fi}
387 \let\l@dchset@num\relax
388

```

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

```

389 \renewcommand*{\page@action}{%
390 \ifledRcol
391 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
392 \xright@appenditem{\next@page@numR}\to\actions@listR
393 \else
394 \xright@appenditem{\the\absline@num}\to\actionlines@list
395 \xright@appenditem{\next@page@num}\to\actions@list
396 \fi}

```

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

```

397 \renewcommand*{\set@line@action}{%
398 \ifledRcol
399 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
400 \ifsublines@
401 \@l@dttempcnta=-\subline@numR
402 \else
403 \@l@dttempcnta=-\line@numR
404 \fi
405 \advance\@l@dttempcnta by -5000\relax
406 \xright@appenditem{\the\@l@dttempcnta}\to\actions@listR
407 \else
408 \xright@appenditem{\the\absline@num}\to\actionlines@list
409 \ifsublines@
410 \@l@dttempcnta=-\subline@num
411 \else
412 \@l@dttempcnta=-\line@num
413 \fi
414 \advance\@l@dttempcnta by -5000\relax
415 \xright@appenditem{\the\@l@dttempcnta}\to\actions@list
416 \fi}
417

```

`\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.

```

418 \renewcommand*{\sub@action}{%
419 \ifledRcol
420 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
421 \ifsublines@

```

```

422 \xright@appenditem{-1001}\to\actions@listR
423 \else
424 \xright@appenditem{-1002}\to\actions@listR
425 \fi
426 \else
427 \xright@appenditem{\the\absline@num}\to\actionlines@list
428 \ifsublines@
429 \xright@appenditem{-1001}\to\actions@list
430 \else
431 \xright@appenditem{-1002}\to\actions@list
432 \fi
433 \fi}
434

```

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

```

435 \newcount\@lockR
436 \newcount\sub@lockR
437
438 \newcommand*{\do@lockonR}{%
439 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
440 \ifsublines@
441 \xright@appenditem{-1005}\to\actions@listR
442 \ifnum\sub@lockR=\z@
443 \sub@lockR \@ne
444 \else
445 \ifnum\sub@lockR=\thr@@
446 \sub@lockR \@ne
447 \fi
448 \fi
449 \else
450 \xright@appenditem{-1003}\to\actions@listR
451 \ifnum\@lockR=\z@
452 \@lockR \@ne
453 \else
454 \ifnum\@lockR=\thr@@
455 \@lockR \@ne
456 \fi
457 \fi
458 \fi}
459
460 \renewcommand*{\do@lockon}{%
461 \ifx\next\lock@off
462 \global\let\lock@off=\skip@lockoff
463 \else
464 \ifledRcol
465 \do@lockonR
466 \else
467 \do@lockonL

```



```

468 \fi
469 \fi}

```

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

```

\do@lockoff 470
\do@lockoffR 471
\skip@lockoff 472 \newcommand{\do@lockoffR}{%
473 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
474 \ifsublines@
475 \xright@appenditem{-1006}\to\actions@listR
476 \ifnum\sub@lockR=\tw@
477 \sub@lockR \thr@@
478 \else
479 \sub@lockR \z@
480 \fi
481 \else
482 \xright@appenditem{-1004}\to\actions@listR
483 \ifnum\@lockR=\tw@
484 \@lockR \thr@@
485 \else
486 \@lockR \z@
487 \fi
488 \fi}
489
490 \renewcommand*{\do@lockoff}{%
491 \ifledRcol
492 \do@lockoffR
493 \else
494 \do@lockoffL
495 \fi}
496 \global\let\lock@off=\do@lockoff
497

```

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

```

498 \providecommand*\n@num{}
499 \renewcommand*\n@num{%
500 \ifledRcol
501 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
502 \xright@appenditem{-1007}\to\actions@listR
503 \else
504 \n@num@reg
505 \fi}
506

```

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

- #1, the number of entries to add to `\insertlines@list` for this reference. This value for right text, here and within `\edtext`, which computes it and

writes it to the line-list file, will be stored in the count `\insert@countR`.

```
507 \newcount\insert@countR
```

- #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`.)

The first thing `\@ref` itself does is to add the specified number of items to the `\insertlines@list` list.

```
508 \renewcommand*{\@ref}[2]{%
509 \ifledRcol
510 \global\insert@countR=#1\relax
511 \loop\ifnum\insert@countR>\z@
512 \xright@appenditem{\the\absline@numR}\to\insertlines@listR
513 \global\advance\insert@countR \m@ne
514 \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.

```
515 \begingroup
516 \let\@ref=\dummy@ref
517 \let\page@action=\relax
518 \let\sub@action=\relax
519 \let\set@line@action=\relax
520 \let\@lab=\relax
521 #2
522 \global\endpage@num=\page@numR
523 \global\endline@num=\line@numR
524 \global\endsubline@num=\subline@numR
525 \endgroup
```

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

```
526 \xright@appenditem%
527 {\the\page@numR|\the\line@numR|}%
528 \ifsublines@ \the\subline@numR \else 0\fi|}%
529 \the\endpage@num|\the\endline@num|}%
530 \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@listR
```

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

```
531 #2
532 \else
```

And when not in right text

```
533 \@ref@reg{#1}{#2}%
534 \fi}
```

`\@pend` `\@pend{<num>}` adds its argument to the `\linesinpar@listL` list, and analogously `\@pendR` for `\@pendR`. We start off with a `\providecommand` just in case an older version of `ledmac` is being used which does not define these macros.

```
535 \providecommand*\@pend}[1]{}
536 \renewcommand*\@pend}[1]{%
537 \xright@appenditem{#1}\to\linesinpar@listL}
538 \providecommand*\@pendR}[1]{}
539 \renewcommand*\@pendR}[1]{%
540 \xright@appenditem{#1}\to\linesinpar@listR}
541
```

`\@lopL` `\@lopL{<num>}` adds its argument to the `\linesonpage@listL` list, and analogously `\@lopR` for `\@lopR`. We start off with a `\providecommand` just in case an older version of `ledmac` is being used which does not define these macros.

```
542 \providecommand*\@lopL}[1]{}
543 \renewcommand*\@lopL}[1]{%
544 \xright@appenditem{#1}\to\linesonpage@listL}
545 \providecommand*\@lopR}[1]{}
546 \renewcommand*\@lopR}[1]{%
547 \xright@appenditem{#1}\to\linesonpage@listR}
548
```

## 11.5 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@outR` The file for right texts will be opened on output stream `\linenum@outR`.

```
549 \newwrite\linenum@outR
```

`\iffirst@linenum@outR` Once any file is opened on this stream, we keep it open forever, or else switch to another file that we keep open.

```
\first@linenum@out@Rtrue
\first@linenum@out@Rfalse 550 \newif\iffirst@linenum@out@R
551 \first@linenum@out@Rtrue
```

`\line@list@stuffR` This is the right text version of the `\line@list@stuff{<file>}` macro. It is called by `\beginnumberingR` and performs all the line-list operations needed at the start of a section. Its argument is the name of the line-list file.

```
552 \newcommand*\line@list@stuffR}[1]{%
553 \read@linelist{#1}%
554 \iffirst@linenum@out@R
555 \immediate\closeout\linenum@outR
556 \global\first@linenum@out@Rfalse
557 \immediate\openout\linenum@outR=#1
558 \else
559 \closeout\linenum@outR
560 \openout\linenum@outR=#1
```

```
561 \fi}
562
```

`\new@lineR` The `\new@lineR` macro sends the `\@l` command to the right text line-list file, to mark the start of a new text line.

```
563 \newcommand*{\new@lineR}{%
564 \write\linenum@outR{\string\@l[\the\c@page][\thepage]}}
```

`\flag@start` We enclose a lemma marked by `\edtext` in `\flag@start` and `\flag@end`: these `\flag@end` send the `\@ref` command to the line-list file.

```
565 \renewcommand*{\flag@start}{%
566 \ifledRcol
567 \edef\next{\write\linenum@outR{%
568 \string\@ref[\the\insert@countR] []}%
569 \next
570 \else
571 \edef\next{\write\linenum@out{%
572 \string\@ref[\the\insert@count] []}%
573 \next
574 \fi}
575 \renewcommand*{\flag@end}{%
576 \ifledRcol
577 \write\linenum@outR{[]}%
578 \else
579 \write\linenum@out{[]}%
580 \fi}
```

`\startsub` `\startsub` and `\endsub` turn sub-lineation on and off, by writing appropriate instructions to the line-list file.

```
581 \renewcommand*{\startsub}{\dimen0\lastskip
582 \ifdim\dimen0>0pt \unskip \fi
583 \ifledRcol \write\linenum@outR{\string\sub@on}%
584 \else \write\linenum@out{\string\sub@on}%
585 \fi
586 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
587 \def\endsub{\dimen0\lastskip
588 \ifdim\dimen0>0pt \unskip \fi
589 \ifledRcol \write\linenum@outR{\string\sub@off}%
590 \else \write\linenum@out{\string\sub@off}%
591 \fi
592 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
593
```

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

```
594 \renewcommand*{\advanceline}[1]{%
595 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
596 \else \write\linenum@out{\string\@adv[#1]}%
597 \fi}
```

`\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.

```
598 \renewcommand*{\setline}[1]{%
599 \ifnum#1<\z@
600 \led@warn@BadSetline
601 \else
602 \ifledRcol \write\linenum@outR{\string\@set[#1]}%
603 \else \write\linenum@out{\string\@set[#1]}%
604 \fi
605 }
```

`\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.

```
606 \renewcommand*{\setlinenum}[1]{%
607 \ifnum#1<\z@
608 \led@warn@BadSetlinenum
609 \else
610 \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
611 \else \write\linenum@out{\string\l@d@set[#1]} \fi
612 \fi}
613
```

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

```
\endlock
614 \renewcommand*{\startlock}{%
615 \ifledRcol \write\linenum@outR{\string\lock@on}%
616 \else \write\linenum@out{\string\lock@on}%
617 \fi}
618 \def\endlock{%
619 \ifledRcol \write\linenum@outR{\string\lock@off}%
620 \else \write\linenum@out{\string\lock@off}%
621 \fi}
622
```

`\skipnumbering` In numbered text, `\skipnumbering` in a line will suspend the numbering for that particular line. That is, line numbers are unchanged and no line number will be printed.

```
623 \renewcommand*{\skipnumbering}{%
624 \ifledRcol \write\linenum@outR{\string\n@num}%
625 \advanceline{-1}%
626 \else
627 \skipnumbering@reg
628 \fi}
629
```

## 12 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.

`\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}
```

`\critext` Now we begin `\critext` itself.

We slightly modify the original to make accomodation for when right text is being processed.

```
630 \long\def\critext#1#2/{\leavevmode
631 \begingroup
632 \no@expands
633 \xdef\@tag{#1}%
634 \set@line
635 \ifledRcol \global\insert@countR \z@
636 \else \global\insert@count \z@ \fi
637 \ignorespaces #2\relax
638 \flag@start
639 \endgroup
640 \showlemma{#1}%
641 \ifx\end@lemmas\empty \else
642 \glp\end@lemmas\to\x@lemma
643 \x@lemma
644 \global\let\x@lemma=\relax
645 \fi
646 \flag@end}
```

`\edtext` And similarly for `\edtext`.

```
647 \renewcommand{\edtext}[2]{\leavevmode
648 \begingroup
649 \no@expands
650 \xdef\@tag{#1}%
651 \set@line
652 \ifledRcol \global\insert@countR \z@
653 \else \global\insert@count \z@ \fi
654 \ignorespaces #2\relax
655 \flag@start
656 \endgroup
657 \showlemma{#1}%
```

```

658 \ifx\end@lemmas\empty \else
659 \glp\end@lemmas\to\x@lemma
660 \x@lemma
661 \global\let\x@lemma=\relax
662 \fi
663 \flag@end}
664

```

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

```

665 \renewcommand*{\set@line}{%
666 \ifledRcol
667 \ifx\line@listR\empty
668 \global\noteschanged@true
669 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
670 \else
671 \glp\line@listR\to\@tempb
672 \xdef\l@d@nums{\@tempb|\edfont@info}%
673 \global\let\@tempb=\undefined
674 \fi
675 \else
676 \ifx\line@list\empty
677 \global\noteschanged@true
678 \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
679 \else
680 \glp\line@list\to\@tempb
681 \xdef\l@d@nums{\@tempb|\edfont@info}%
682 \global\let\@tempb=\undefined
683 \fi
684 \fi}
685

```

## 13 Parallel environments

The initial set up for parallel processing is deceptively simple.

`pairs` The `pairs` environment is for parallel columns and the `pages` environment for parallel pages.

```

chapterinpages 686 \newenvironment{pairs}{%
687 \l@dpairingtrue
688 \l@dpagingfalse
689 }{%
690 \l@dpairingfalse
691 }

```

The `pages` environment additionally sets the ‘column’ widths to the `\textwidth` (as known at the time the package is called). As in this environnement there is two text in parallels in 2 pages, chapter have not to start in a left page. So the `\chapter` command is redefined to make no test about clearing page.

```

692 \newenvironment{pages}{%
693 \let\oldchapter\chapter
694 \let\chapter\chapterinpages
695 \l@dpairingtrue
696 \l@dpagingtrue
697 \setlength{\Lcolwidth}{\textwidth}%
698 \setlength{\Rcolwidth}{\textwidth}%
699 }{%
700 \l@dpairingfalse
701 \l@dpagingfalse
702 \let\chapter\oldchapter
703 }
704 \newcommand{\chapterinpages}{\thispagestyle{plain}%
705 \global\@topnum\z@
706 \@afterindentfalse
707 \secdef\@chapter\@schapter}
708

```

**Leftside** Within the `pairs` and `pages` environments the left and right hand texts are within `Leftside` and `Rightside` environments, respectively. The `Leftside` environment is simple, indicating that right text is not within its purview and using some particular macros.

```

709 \newenvironment{Leftside}{%
710 \ledRcolfalse
711 \let\pstart\pstartL
712 \let\pend\pendL
713 \let\memorydump\memorydumpL
714 \Leftsidehook
715 }{\Leftsidehookend}

```

`\Leftsidehook` Hooks into the start and end of the `Leftside` and `Rightside` environments. These  
`\Leftsidehookend` are initially empty.

```

\Rightsidehook 716 \newcommand*{\Leftsidehook}{}
\Rightsidehookend 717 \newcommand*{\Leftsidehookend}{}
718 \newcommand*{\Rightsidehook}{}
719 \newcommand*{\Rightsidehookend}{}
720

```

**Rightside** The `Rightside` environment is only slightly more complicated than the `Leftside`. Apart from indicating that right text is being provided it ensures that the right right text code will be used.

```

721 \newenvironment{Rightside}{%
722 \ledRcoltrue
723 \let\beginnumbering\beginnumberingR
724 \let\endnumbering\endnumberingR
725 \let\pausenumbering\pausenumberingR
726 \let\resumenumbering\resumenumberingR
727 \let\memorydump\memorydumpR
728 \let\pstart\pstartR

```



```

729 \let\pend\pendR
730 \let\lineation\lineationR
731 \Rightsidehook
732 }-{}
733 \ledRcolfalse
734 \Rightsidehookend
735 }
736

```

## 14 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.

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

**\num@linesR** Here are numbers and flags that are used internally in the course of the paragraph decomposition.

**\one@lineR** When we first form the paragraph, it goes into a box register, **\l@dLcolrawbox** or **\l@dRcolrawbox** for right text, instead of onto the current vertical list. The **\ifnumberedpar@** flag will be **true** while a paragraph is being processed in that way. **\num@lines(R)** will store the number of lines in the paragraph when it's complete. When we chop it up into lines, each line in turn goes into the **\one@line** or **\one@lineR** register, and **\par@line(R)** will be the number of that line within the paragraph.

```

737 \newcount\num@linesR
738 \newbox\one@lineR
739 \newcount\par@lineR

```

**\pstartL** **\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 appropriate box. **\pstart** needs to appear at the start of every paragraph that's to be numbered.

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.

We have to have specific left and right **\pstart** when parallel processing; among other things because of potential changes in the linewidth.

```

740 \newcommand*{\pstartL}{
741 \if@nbreak
742 \let\@oldnbreak\@nbreaktrue
743 \else

```

```

744 \let\@oldnobreak\@nobreakfalse
745 \fi
746 \@nobreaktrue
747 \ifnumbering \else
748 \led@err@PstartNotNumbered
749 \beginnumbering
750 \fi
751 \ifnumberedpar@
752 \led@err@PstartInPstart
753 \pend
754 \fi

```

If this is the first `\pstart` in a numbered section, clear any inserts and set `\ifpstart@rtedL` to FALSE.

```

755 \ifpstart@rtedL\else
756 \list@clear{\inserts@list}%
757 \global\let\next@insert=\empty
758 \global\pstart@rtedLtrue
759 \fi
760 \begingroup\normal@pars

```

When parallel processing we check that we haven't exceeded the maximum number of chunks. In any event we grab a box for the forthcoming text.

```

761 \global\advance\l@dnumpstartsL \@ne
762 \ifnum\l@dnumpstartsL>\l@dc@maxchunks
763 \led@err@TooManyPstarts
764 \global\l@dnumpstartsL=\l@dc@maxchunks
765 \fi
766 \global\setnamebox{\l@dLcolrawbox\the\l@dnumpstartsL}=\vbox\bgroup%
767 \hsize=\lcolwidth
768 \numberedpar@true}

769 \newcommand*{\pstartR}{
770 \if@nobreak
771 \let\@oldnobreak\@nobreaktrue
772 \else
773 \let\@oldnobreak\@nobreakfalse
774 \fi
775 \@nobreaktrue
776 \ifnumberingR \else
777 \led@err@PstartNotNumbered
778 \beginnumberingR
779 \fi
780 \ifnumberedpar@
781 \led@err@PstartInPstart
782 \pendR
783 \fi
784 \ifpstart@rtedR\else
785 \list@clear{\inserts@listR}%
786 \global\let\next@insertR=\empty
787 \global\pstart@rtedRtrue

```

```

788 \fi
789 \begingroup\normal@pars
790 \global\advance\l@dnumpstartsR \@ne
791 \ifnum\l@dnumpstartsR>\l@dc@maxchunks
792 \led@err@TooManyPstarts
793 \global\l@dnumpstartsR=\l@dc@maxchunks
794 \fi
795 \global\setnamebox{\l@dRcolrawbox\the\l@dnumpstartsR}=\vbox\bgroup%
796 \hsize=\Rcolwidth
797 \numberedpar@true}

```

`\pendL` `\pend` must be used to end a numbered paragraph. Again we need a version that knows about left parallel texts.

```

798 \newcommand*{\pendL}{\ifnumbering \else
799 \led@err@PendNotNumbered
800 \fi
801 \ifnumberedpar@ \else
802 \led@err@PendNoPstart
803 \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.

```

804 \l@dzeropenalties
805 \endgraf\global\num@lines=\prevgraf\egroup
806 \global\par@line=0

```

End the group that was begun in the `\pstart`.

```

807 \endgroup
808 \ignorespaces
809 \@oldnobreak}
810

```

`\pendR` The version of `\pend` needed for right texts.

```

811 \newcommand*{\pendR}{\ifnumberingR \else
812 \led@err@PendNotNumbered
813 \fi
814 \ifnumberedpar@ \else
815 \led@err@PendNoPstart
816 \fi
817 \l@dzeropenalties
818 \endgraf\global\num@linesR=\prevgraf\egroup
819 \global\par@lineR=0
820 \endgroup
821 \ignorespaces
822 \@oldnobreak}
823

```

## 14.2 Processing one line

For parallel texts we have to be able to process left and right lines independently. For sequential text we happily use the original `\do@line`. Otherwise ...

```

\l@dleftbox A line of left text will be put in the box \l@dleftbox, and analagously for a line
\l@drightbox of right text.
824 \newbox\l@dleftbox
825 \newbox\l@drightbox
826

\countLline We need to know the number of lines processed.
\countRline 827 \newcount\countLline
828 \countLline \z@
829 \newcount\countRline
830 \countRline \z@
831

\@donereallinesL We need to know the number of ‘real’ lines output (i.e., those that have been input
\@donetotallinesL by the user), and the total lines output (which includes any blank lines output for
\@donereallinesR synchronisation).
\@donetotallinesR 832 \newcount\@donereallinesL
833 \newcount\@donetotallinesL
834 \newcount\@donereallinesR
835 \newcount\@donetotallinesR
836

\do@lineL The \do@lineL macro is called to do all the processing for a single line of left text.

837 \newcommand*{\do@lineL}{%
838 \advance\countLline \@ne
839 \ifvbox\namebox{\l@dLcolrawbox\the\l@dpscl}%
840 {\vbadness=10000
841 \splittopskip=\z@
842 \do@lineLhook
843 \l@emptyd@ta
844 \global\setbox\one@line=\vsplit\namebox{\l@dLcolrawbox\the\l@dpscl}
845 to\baselineskip}%
846 \unvbox\one@line \global\setbox\one@line=\lastbox
847 \getline@num
848 \setbox\l@dleftbox
849 \hb@xt@ \Lcolwidth{%
850 \affixline@num
851 \l@dld@ta
852 \add@inserts
853 \affixside@note
854 \l@dlsn@te
855 {\ledllfill\hb@xt@ \wd\one@line{\new@line\l@dunhbox@line{\one@line}}\ledrlfill\l@drd@
856 \l@drsn@te

```

```

857 }}%
858 \add@penaltiesL
859 \global\advance\@donereallinesL\@ne
860 \global\advance\@donetotallinesL\@ne
861 \else
862 \setbox\l@leftbox \hb@xt@ \Lcolwidth{\hspace*\Lcolwidth}}%
863 \global\advance\@donetotallinesL\@ne
864 \fi}
865
866

```

`\do@lineLhook` Hooks, initially empty, into the respective `\do@line(L/R)` macros.

```

\do@lineRhook 867 \newcommand*{\do@lineLhook}{}
868 \newcommand*{\do@lineRhook}{}
869

```

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

```

870 \newcommand*{\do@lineR}{%
871 \advance\countRline \@ne
872 \ifvbox\namebox{\l@dRcolrawbox\the\l@dpscR}%
873 {\vbadness=10000
874 \splittopskip=\z@
875 \do@lineRhook
876 \l@demptyd@ta
877 \global\setbox\one@lineR=\vsplit\namebox{\l@dRcolrawbox\the\l@dpscR}
878 to\baselineskip}%
879 \unvbox\one@lineR \global\setbox\one@lineR=\lastbox
880 \getline@numR
881 \setbox\l@drightbox
882 \hb@xt@ \Rcolwidth{%
883 \affixline@numR
884 \l@dld@ta
885 \add@insertsR
886 \affixside@noteR
887 \l@dlsn@te
888 {\ledllfill\hb@xt@ \wd\one@lineR{\new@lineR\l@dunhbox@line{\one@lineR}}\ledrlfill\l@d@rd@ta%
889 \l@d@rsn@te
890 }}%
891 \add@penaltiesR
892 \global\advance\@donereallinesR\@ne
893 \global\advance\@donetotallinesR\@ne
894 \else
895 \setbox\l@drightbox \hb@xt@ \Rcolwidth{\hspace*\Rcolwidth}}
896 \global\advance\@donetotallinesR\@ne
897 \fi}
898
899

```

### 14.3 Line and page number computation

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

```

900 \newcommand*{\getline@numR}{%
901 \global\advance\absline@numR \@ne
902 \do@actionsR
903 \do@ballastR
904 \ifsublines@
905 \ifnum\sub@lockR<\tw@
906 \global\advance\subline@numR \@ne
907 \fi
908 \else
909 \ifnum\@lockR<\tw@
910 \global\advance\line@numR \@ne
911 \global\subline@numR \z@
912 \fi
913 \fi}
914
915
```

`\do@ballastR` The real work in the line macros above is done in `\do@actions`, but before we plunge into that, let's get `\do@ballastR` out of the way.

```

916 \newcommand*{\do@ballastR}{\global\ballast@count=\z@
917 \begingroup
918 \advance\absline@numR \@ne
919 \ifnum\next@actionlineR=\absline@numR
920 \ifnum\next@actionlineR>-1001
921 \global\advance\ballast@count by -\c@ballast
922 \fi
923 \fi
924 \endgroup}

```

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

`\do@actions@fixedcodeR`

`\do@actions@nextR`

It may call itself recursively and we use tail recursion, via `\do@actions@nextR` for this.

```

925 \newcommand*{\do@actions@fixedcodeR}{%
926 \ifcase\@l@tempcnta%
927 \or% % 1001
928 \global\sublines@true
929 \or% % 1002
930 \global\sublines@false
931 \or% % 1003
932 \global\@lockR=\@ne
933 \or% % 1004
934 \ifnum\@lockR=\tw@
935 \global\@lockR=\thr@@

```

```

936 \else
937 \global\@lockR=\z@
938 \fi
939 \or% % 1005
940 \global\sub@lockR=\@ne
941 \or% % 1006
942 \ifnum\sub@lockR=\tw@
943 \global\sub@lockR=\thr@@
944 \else
945 \global\sub@lockR=\z@
946 \fi
947 \or% % 1007
948 \l@dskipnumbertrue
949 \else
950 \led@warn@BadAction
951 \fi}
952
953
954 \newcommand*{\do@actionsR}{%
955 \global\let\do@actions@nextR=\relax
956 \@l@dttempcntb=\absline@numR
957 \ifnum\@l@dttempcntb<\next@actionlineR\else
958 \ifnum\next@actionR>-1001\relax
959 \global\page@numR=\next@actionR
960 \ifbypage@R
961 \global\line@numR \z@ \global\subline@numR \z@
962 \fi
963 \else
964 \ifnum\next@actionR<-4999\relax % 9/05 added relax here
965 \@l@dttempcnta=-\next@actionR
966 \advance\@l@dttempcnta by -5001\relax
967 \ifsublines@
968 \global\subline@numR=\@l@dttempcnta
969 \else
970 \global\line@numR=\@l@dttempcnta
971 \fi
972 \else
973 \@l@dttempcnta=-\next@actionR
974 \advance\@l@dttempcnta by -1000\relax
975 \do@actions@fixedcodeR
976 \fi
977 \fi
978 \ifx\actionlines@listR\empty
979 \gdef\next@actionlineR{1000000}%
980 \else
981 \gl@p\actionlines@listR\to\next@actionlineR
982 \gl@p\actions@listR\to\next@actionR
983 \global\let\do@actions@nextR=\do@actionsR
984 \fi
985 \fi

```

```

986 \do@actions@nextR}
987

```

#### 14.4 Line number printing

```

\l@dcalnum \affixline@numR is the right text version of the \affixline@num macro.
\ch@cksub@l@ckR 988
\ch@ck@l@ckR 989 \providecommand*\l@dcalnum}[3]{%
\fx@l@cksR 990 \ifnum #1 > #2\relax
\affixline@numR 991 \@l@tempcnta = #1\relax
992 \advance\@l@tempcnta by -#2\relax
993 \divide\@l@tempcnta by #3\relax
994 \multiply\@l@tempcnta by #3\relax
995 \advance\@l@tempcnta by #2\relax
996 \else
997 \@l@tempcnta=#2\relax
998 \fi}
999
1000 \newcommand*\ch@cksub@l@ckR}{%
1001 \ifcase\sub@lockR
1002 \or
1003 \ifnum\sublock@disp=\@ne
1004 \@l@tempcntb \z@ \@l@tempcnta \@ne
1005 \fi
1006 \or
1007 \ifnum\sublock@disp=\tw@
1008 \else
1009 \@l@tempcntb \z@ \@l@tempcnta \@ne
1010 \fi
1011 \or
1012 \ifnum\sublock@disp=\z@
1013 \@l@tempcntb \z@ \@l@tempcnta \@ne
1014 \fi
1015 \fi}
1016
1017 \newcommand*\ch@ck@l@ckR}{%
1018 \ifcase\@lockR
1019 \or
1020 \ifnum\lock@disp=\@ne
1021 \@l@tempcntb \z@ \@l@tempcnta \@ne
1022 \fi
1023 \or
1024 \ifnum\lock@disp=\tw@
1025 \else
1026 \@l@tempcntb \z@ \@l@tempcnta \@ne
1027 \fi
1028 \or
1029 \ifnum\lock@disp=\z@
1030 \@l@tempcntb \z@ \@l@tempcnta \@ne

```



```

1031 \fi
1032 \fi}
1033
1034 \newcommand*{\f@x@l@cksR}{%
1035 \ifcase\@lockR
1036 \or
1037 \global\@lockR \tw@
1038 \or \or
1039 \global\@lockR \z@
1040 \fi
1041 \ifcase\sub@lockR
1042 \or
1043 \global\sub@lockR \tw@
1044 \or \or
1045 \global\sub@lockR \z@
1046 \fi}
1047
1048
1049 \newcommand*{\affixline@numR}{%
1050 \ifl@dskipnumber
1051 \global\l@dskipnumberfalse
1052 \else
1053 \ifsublines@
1054 \l@dttempcntb=\subline@numR
1055 \l@dcalcnum{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR}%
1056 \ch@cksub@lockR
1057 \else
1058 \l@dttempcntb=\line@numR
1059 \ifx\linenumberlist\empty
1060 \l@dcalcnum{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%
1061 \else
1062 \l@dttempcnta=\line@numR
1063 \edef\rem@inder{\linenumberlist,\number\line@numR,%
1064 \edef\sc@n@list{\def\noexpand\sc@n@list
1065 ###1,\number\l@dttempcnta,###2|{\def\noexpand\rem@inder{###2}}}%
1066 \sc@n@list\expandafter\sc@n@list\rem@inder|%
1067 \ifx\rem@inder\empty\advance\l@dttempcnta\@ne\fi
1068 \fi
1069 \ch@ck@l@ckR
1070 \fi
1071 \ifnum\l@dttempcnta=\l@dttempcntb
1072 \if@twocolumn
1073 \if@firstcolumn
1074 \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1075 \else
1076 \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1077 \fi
1078 \else
1079 \l@dttempcntb=\line@marginR
1080 \ifnum\l@dttempcntb>\@ne

```

```

1081 \advance\l@dttempcntb by\page@numR
1082 \fi
1083 \ifodd\l@dttempcntb
1084 \gdef\l@drd@ta{\rlap{{\rightlinenumR}}}%
1085 \else
1086 \gdef\l@dld@ta{\llap{{\leftlinenumR}}}%
1087 \fi
1088 \fi
1089 \fi
1090 \f@x@l@cksR
1091 \fi}
1092

```

## 14.5 Add insertions to the vertical list

`\inserts@listR` `\inserts@listR` is the list macro that contains the inserts that we save up for one right text paragraph.

```

1093 \list@create{\inserts@listR}

```

`\add@insertsR` The right text version.

```

\add@inserts@nextR 1094 \newcommand*{\add@insertsR}{%
1095 \global\let\add@inserts@nextR=\relax
1096 \ifx\inserts@listR\empty \else
1097 \ifx\next@insertR\empty
1098 \ifx\insertlines@listR\empty
1099 \global\noteschanged@true
1100 \gdef\next@insertR{100000}%
1101 \else
1102 \gl@p\insertlines@listR\to\next@insertR
1103 \fi
1104 \fi
1105 \ifnum\next@insertR=\absline@numR
1106 \gl@p\inserts@listR\to\@insertR
1107 \@insertR
1108 \global\let\@insertR=\undefined
1109 \global\let\next@insertR=\empty
1110 \global\let\add@inserts@nextR=\add@insertsR
1111 \fi
1112 \fi
1113 \add@inserts@nextR}
1114

```

## 14.6 Penalties

`\add@penaltiesL` `\add@penaltiesL` is the last macro used by `\do@lineL`. 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 the code below, which is a virtual copy of the original `\add@penalties`, `\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`. Finally, the penalty is checked to see that it doesn't go below  $-10000$ .

```
\newcommand*{\add@penaltiesR}{\@l@tempcnta=\ballast@count
\ifnum\num@linesR>\@ne
\global\advance\par@lineR \@ne
\ifnum\par@lineR=\@ne
\advance\@l@tempcnta by \clubpenalty
\fi
\@l@tempcntb=\par@lineR \advance\@l@tempcntb \@ne
\ifnum\@l@tempcntb=\num@linesR
\advance\@l@tempcnta by \widowpenalty
\fi
\ifnum\par@lineR<\num@linesR
\advance\@l@tempcnta by \interlinepenalty
\fi
\fi
\ifnum\@l@tempcnta=\z@
\relax
\else
\ifnum\@l@tempcnta>-10000
\penalty\@l@tempcnta
\else
\penalty -10000
\fi
\fi}
```

This is for a single chunk. However, as we are probably dealing with several chunks at a time, the above is not really relevant. I think that it is likely with parallel text that there is no real need to add back any penalties; even if there was, they would have to match across the left and right lines. So, I end up with the following.

```
1115 \newcommand*{\add@penaltiesL}{\}
1116 \newcommand*{\add@penaltiesR}{\}
1117
```

## 14.7 Printing leftover notes

`\flush@notesR` The `\flush@notesR` macro is called after the entire right text has been sliced up and sent on to the vertical list.

```
1118 \newcommand*{\flush@notesR}{\%
1119 \xloop
1120 \ifx\inserts@listR\empty \else
1121 \glp\inserts@listR\to\@insertR
1122 \@insertR
```



```

1154 \renewcommand*{\Cfootnote}[1]{%
1155 \ifnumberedpar@
1156 \ifledRcol
1157 \xright@appenditem{\noexpand\Cfootnote{C}%
1158 {\l@d@nums}{\@tag}{#1}}\to\inserts@listR
1159 \global\advance\insert@countR \@ne
1160 \else
1161 \xright@appenditem{\noexpand\Cfootnote{C}%
1162 {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1163 \global\advance\insert@count \@ne
1164 \fi
1165 \else
1166 \vCfootnote{C}{\l@d@nums}{\@tag}{#1}}%
1167 \fi\ignorespaces}

1168 \renewcommand*{\Dfootnote}[1]{%
1169 \ifnumberedpar@
1170 \ifledRcol
1171 \xright@appenditem{\noexpand\Dfootnote{D}%
1172 {\l@d@nums}{\@tag}{#1}}\to\inserts@listR
1173 \global\advance\insert@countR \@ne
1174 \else
1175 \xright@appenditem{\noexpand\Dfootnote{D}%
1176 {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1177 \global\advance\insert@count \@ne
1178 \fi
1179 \else
1180 \vDfootnote{D}{\l@d@nums}{\@tag}{#1}}%
1181 \fi\ignorespaces}

1182 \renewcommand*{\Efootnote}[1]{%
1183 \ifnumberedpar@
1184 \ifledRcol
1185 \xright@appenditem{\noexpand\Efootnote{E}%
1186 {\l@d@nums}{\@tag}{#1}}\to\inserts@listR
1187 \global\advance\insert@countR \@ne
1188 \else
1189 \xright@appenditem{\noexpand\Efootnote{E}%
1190 {\l@d@nums}{\@tag}{#1}}\to\inserts@list
1191 \global\advance\insert@count \@ne
1192 \fi
1193 \else
1194 \vEfootnote{E}{\l@d@nums}{\@tag}{#1}}%
1195 \fi\ignorespaces}
1196

```

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

```

\mpBfootnote 1197 \renewcommand*{\mpAfootnote}[1]{%
\mpCfootnote 1198 \ifnumberedpar@
\mpDfootnote 1199 \ifledRcol
\mpEfootnote 1200 \xright@appenditem{\noexpand\mpvAfootnote{A}%

```

```

1201 {{\l@d@nums}{\@tag}{#1}}\to\inserts@listR
1202 \global\advance\insert@countR \@ne
1203 \else
1204 \xright@appenditem{\noexpand\mpvAfootnote{A}%
1205 {{\l@d@nums}{\@tag}{#1}}\to\inserts@list
1206 \global\advance\insert@count \@ne
1207 \fi
1208 \else
1209 \mpvAfootnote{A}{{0|0|0|0|0|0|0|0}}{#1}}%
1210 \fi\ignorespaces}
1211 \renewcommand*{\mpBfootnote}[1]{%
1212 \ifnumberedpar@
1213 \iflabeledRcol
1214 \xright@appenditem{\noexpand\mpvBfootnote{B}%
1215 {{\l@d@nums}{\@tag}{#1}}\to\inserts@listR
1216 \global\advance\insert@countR \@ne
1217 \else
1218 \xright@appenditem{\noexpand\mpvBfootnote{B}%
1219 {{\l@d@nums}{\@tag}{#1}}\to\inserts@list
1220 \global\advance\insert@count \@ne
1221 \fi
1222 \else
1223 \mpvBfootnote{B}{{0|0|0|0|0|0|0|0}}{#1}}%
1224 \fi\ignorespaces}
1225 \renewcommand*{\mpCfootnote}[1]{%
1226 \ifnumberedpar@
1227 \iflabeledRcol
1228 \xright@appenditem{\noexpand\mpvCfootnote{C}%
1229 {{\l@d@nums}{\@tag}{#1}}\to\inserts@listR
1230 \global\advance\insert@countR \@ne
1231 \else
1232 \xright@appenditem{\noexpand\mpvCfootnote{C}%
1233 {{\l@d@nums}{\@tag}{#1}}\to\inserts@list
1234 \global\advance\insert@count \@ne
1235 \fi
1236 \else
1237 \mpvCfootnote{C}{{0|0|0|0|0|0|0|0}}{#1}}%
1238 \fi\ignorespaces}
1239 \renewcommand*{\mpDfootnote}[1]{%
1240 \ifnumberedpar@
1241 \iflabeledRcol
1242 \xright@appenditem{\noexpand\mpvDfootnote{D}%
1243 {{\l@d@nums}{\@tag}{#1}}\to\inserts@listR
1244 \global\advance\insert@countR \@ne
1245 \else
1246 \xright@appenditem{\noexpand\mpvDfootnote{D}%
1247 {{\l@d@nums}{\@tag}{#1}}\to\inserts@list
1248 \global\advance\insert@count \@ne
1249 \fi

```

## 15.2 Normal footnote formatting

```

\printlinesR This is the right text version of \printlines and takes account of \Rlineflag.
\ledsavedprintlines Just in case, \ledsavedprintlines is a copy of the original \printlines.

```

|                           |                         |                   |                      |                       |                   |                      |                   |
|---------------------------|-------------------------|-------------------|----------------------|-----------------------|-------------------|----------------------|-------------------|
| <code>\printlinesR</code> | <code>#1</code>         | <code>#2</code>   | <code>#3</code>      | <code>#4</code>       | <code>#5</code>   | <code>#6</code>      | <code>#7</code>   |
|                           | <code>start-page</code> | <code>line</code> | <code>subline</code> | <code>end-page</code> | <code>line</code> | <code>subline</code> | <code>font</code> |

```

1267 \def\printlinesR#1|#2|#3|#4|#5|#6|#7|{\begingroup
1268 \setprintlines{#1}-{#2}-{#3}-{#4}-{#5}-{#6}%
1269 \ifl@d@pnum #1\fullstop\fi
1270 \ifledplinenum \linenumr@p{#2}\Rlineflag\else \symlinenum\fi
1271 \ifl@d@ssub \fullstop \sublinenumr@p{#3}\fi
1272 \ifl@d@dash \endashchar\fi
1273 \ifl@d@pnum #4\fullstop\fi
1274 \ifl@d@elin \linenumr@p{#5}\Rlineflag\fi
1275 \ifl@d@esl \ifl@d@elin \fullstop\fi \sublinenumr@p{#6}\fi
1276 \endgroup}
1277
1278 \let\ledsavedprintlines\printlines
1279

```

## 16 Cross referencing

`\labelref@listR` Set up a new list, `\labelref@listR`, to hold the page, line and sub-line numbers for each label in right text.

```
1280 \list@create{\labelref@listR}
1281
```

`\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.

```
1282 \renewcommand*{\edlabel}[1]{\@bsphack
1283 \ifledRcol
1284 \write\linenum@outR{\string\@lab}%
1285 \ifx\labelref@listR\empty
1286 \xdef\label@refs{\zz@@@}%
1287 \else
1288 \gl@p\labelref@listR\to\label@refs
1289 \fi
1290 \protected@write\@auxout{%
1291 {\string\l@dmake@labelsR\space\thepage|\label@refs|{#1}}%
1292 \else
1293 \write\linenum@out{\string\@lab}%
1294 \ifx\labelref@list\empty
1295 \xdef\label@refs{\zz@@@}%
1296 \else
1297 \gl@p\labelref@list\to\label@refs
1298 \fi
1299 \fi
1300 \protected@write\@auxout{%
1301 {\string\l@dmake@labels\space\thepage|\label@refs|{#1}}%
1302 \@esphack}
1303
```

`\l@dmake@labelsR` This is the right text version of `\l@dmake@labels`, taking account of `\Rlineflag`.

```
1304 \def\l@dmake@labelsR#1|#2|#3|#4{%
1305 \expandafter\ifx\csname the@label#4\endcsname \relax\else
1306 \led@warn@DuplicateLabel{#4}%
1307 \fi
1308 \expandafter\gdef\csname the@label#4\endcsname{#1|#2\Rlineflag|#3}%
1309 \ignorespaces}
1310 \AtBeginDocument{%
1311 \def\l@dmake@labelsR#1|#2|#3|#4{%
1312 }
1313
```

`\@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.

```

1314 \renewcommand*{\@lab}{%
1315 \ifledRcol
1316 \xright@appenditem{\linenumr@p{\line@numR}}{|%
1317 \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi}%
1318 \to\labelref@listR
1319 \else
1320 \xright@appenditem{\linenumr@p{\line@num}}{|%
1321 \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi}%
1322 \to\labelref@list
1323 \fi}
1324
```

## 17 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.

`\sidenote@marginR` Specifies which margin sidenotes can be in.

```

\sidenotemargin 1325 \newcount\sidenote@marginR
1326 \renewcommand*{\sidenotemargin}[1]{%
1327 \l@dgetsidenote@margin{#1}%
1328 \ifnum\@l@dttempcntb>\m@ne
1329 \ifledRcol
1330 \global\sidenote@marginR=\@l@dttempcntb
1331 \else
1332 \global\sidenote@margin=\@l@dttempcntb
1333 \fi
1334 \fi}}
1335 \sidenotemargin{right}
1336 \global\sidenote@margin=\@ne
1337
```

`\l@dlsnote` The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.

```

\l@dcsnote 1338 \renewcommand*{\l@dlsnote}[1]{%
1339 \ifnumberedpar@
1340 \ifledRcol
1341 \xright@appenditem{\noexpand\l@dlsnote{#1}}{|%
1342 \to\inserts@listR
1343 \global\advance\insert@countR \@ne
1344 \else
1345 \xright@appenditem{\noexpand\l@dlsnote{#1}}{|%
1346 \to\inserts@list
1347 \global\advance\insert@count \@ne
1348 \fi
1349 \fi\ignorespaces}

```

```

1350 \renewcommand*{\l@drsnote}[1]{%
1351 \ifnumberedpar@
1352 \ifledRcol
1353 \xright@appenditem{\noexpand\vl@drsnote{#1}}%
1354 \to\inserts@listR
1355 \global\advance\insert@countR \@ne
1356 \else
1357 \xright@appenditem{\noexpand\vl@drsnote{#1}}%
1358 \to\inserts@list
1359 \global\advance\insert@count \@ne
1360 \fi
1361 \fi\ignorespaces}
1362 \renewcommand*{\l@dcnote}[1]{%
1363 \ifnumberedpar@
1364 \ifledRcol
1365 \xright@appenditem{\noexpand\vl@dcnote{#1}}%
1366 \to\inserts@listR
1367 \global\advance\insert@countR \@ne
1368 \else
1369 \xright@appenditem{\noexpand\vl@dcnote{#1}}%
1370 \to\inserts@list
1371 \global\advance\insert@count \@ne
1372 \fi
1373 \fi\ignorespaces}
1374

```

\affixside@noteR The right text version of \affixside@note.

```

1375 \newcommand*{\affixside@noteR}{%
1376 \gdef\@templ@d{}%
1377 \ifx\@templ@d\l@dcnotetext \else
1378 \if@twocolumn
1379 \if@firstcolumn
1380 \setl@dlp@rbox{\l@dcnotetext}%
1381 \else
1382 \setl@drp@rbox{\l@dcnotetext}%
1383 \fi
1384 \else
1385 \@l@dttempcntb=\sidenote@marginR
1386 \ifnum\@l@dttempcntb>\@ne
1387 \advance\@l@dttempcntb by\page@num
1388 \fi
1389 \ifodd\@l@dttempcntb
1390 \setl@drp@rbox{\l@dcnotetext}%
1391 \else
1392 \setl@dlp@rbox{\l@dcnotetext}%
1393 \fi
1394 \fi
1395 \fi}
1396

```

## 18 Familiar footnotes

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

```

1397 \renewcommand{\l@dbfnote}[1]{%
1398 \ifnumberedpar@
1399 \ifledRcol
1400 \xright@appenditem{\noexpand\vl@dbfnote{#1}}{\@thefnmark}}%
1401 \to\inserts@listR
1402 \global\advance\insert@countR \@ne
1403 \else
1404 \xright@appenditem{\noexpand\vl@dbfnote{#1}}{\@thefnmark}}%
1405 \to\inserts@list
1406 \global\advance\insert@count \@ne
1407 \fi
1408 \fi\ignorespaces}
1409
```

`\normalbfnoteX`

```

1410 \renewcommand{\normalbfnoteX}[2]{%
1411 \ifnumberedpar@
1412 \ifledRcol
1413 \xright@appenditem{\noexpand\vbfnoteX{#1}{#2}}{\@nameuse{thefootnote#1}}}%
1414 \to\inserts@listR
1415 \global\advance\insert@countR \@ne
1416 \else
1417 \xright@appenditem{\noexpand\vbfnoteX{#1}{#2}}{\@nameuse{thefootnote#1}}}%
1418 \to\inserts@list
1419 \global\advance\insert@count \@ne
1420 \fi
1421 \fi\ignorespaces}
1422
```

## 19 Verse

Before we can define the main stanza macros we need to be able to save and reset the category code for `&`. To save the current value we use `\next` from the `\loop` macro.

```

1423 \chardef\next=\catcode'\&
1424 \catcode'\&=\active
1425
```

`astanza` This is roughly an environmental form of `\stanza`, which treats its stanza-like contents as a single chunk.

```

1426 \newenvironment{astanza}{%
1427 \startstanzahook
1428 \catcode'\&\active
```

```

1429 \global\stanza@count\@ne
1430 \ifnum\usernamecount{sza@00}=\z@
1431 \let\stanza@hang\relax
1432 \let\endlock\relax
1433 \else
1434 %% \interlinepenalty\@M % this screws things up, but I don't know why
1435 \rightskip\z@ plus 1fil\relax
1436 \fi
1437 \ifnum\usernamecount{szp@00}=\z@
1438 \let\sza@penalty\relax
1439 \fi
1440 \def&{%
1441 \endlock\mbox{}%
1442 \sza@penalty
1443 \global\advance\stanza@count\@ne
1444 \@astanza@line}%
1445 \def\&{%
1446 \endlock\mbox{}}
1447 \pend
1448 \endstanzaextra}%
1449 \pstart
1450 \@astanza@line
1451 }{}
1452

```

`\@astanza@line` This gets put at the start of each line in the environment. It sets up the paragraph style — each line is treated as a paragraph.

```

1453 \newcommand*{\@astanza@line}{%
1454 \parindent=\csname sza@\number\stanza@count @\endcsname\stanzaindentbase
1455 \par
1456 \stanza@hang%\mbox{}%
1457 \ignorespaces}
1458

```

Lastly reset the modified category codes.

```

1459 \catcode'\&=\next
1460

```

## 20 Naming macros

The LaTeX kernel provides `\@namedef` and `\@namuse` for defining and using macros that may have non-letters in their names. We need something similar here as we are going to need and use some numbered boxes and counters.

```

\newnamebox A set of macros for creating and using ‘named’boxes; the macros are called after
\setnamebox the regular box macros, but including the string ‘name’.
\unhnamebox 1461 \providecommand*{\newnamebox}[1]{%
\unvnamebox 1462 \expandafter\newbox\csname #1\endcsname}
\namebox

```

```

1463 \providecommand*\setnamebox}[1]{%
1464 \expandafter\setbox\csname #1\endcsname}
1465 \providecommand*\unhnamebox}[1]{%
1466 \expandafter\unhbox\csname #1\endcsname}
1467 \providecommand*\unvnamebox}[1]{%
1468 \expandafter\unvbox\csname #1\endcsname}
1469 \providecommand*\namebox}[1]{%
1470 \csname #1\endcsname}
1471

```

`\newnamecount` Macros for creating and using ‘named’ counts.

```

\usenamecount 1472 \providecommand*\newnamecount}[1]{%
1473 \expandafter\newcount\csname #1\endcsname}
1474 \providecommand*\usenamecount}[1]{%
1475 \csname #1\endcsname}
1476

```

## 21 Counts and boxes for parallel texts

In sequential text, each chunk (that enclosed by `\pstart ... \pend`) is put into a box called `\raw@text` and then immediately printed, resulting in the box being emptied and ready for the next chunk. For parallel processing multiple boxes are needed as printing is delayed. We also need extra counters for various things.

`\maxchunks` The maximum number of chunk pairs before printing has to be called for. The  
`\l@dc@maxchunks` default is 10 chunk pairs.

```

1477 \newcount\l@dc@maxchunks
1478 \newcommand*\maxchunks}[1]{\l@dc@maxchunks=#1}
1479 \maxchunks{10}
1480

```

`\l@dnumpstartsL` The numbers of left and right chunks. `\l@dnumpstartsL` is defined in `ledmac`.

```

\l@dnumpstartsR 1481 \newcount\l@dnumpstartsR
1482

```

`\l@pscL` A couple of scratch counts for use in left and right texts, respectively.

```

\l@pscR 1483 \newcount\l@dpscL
1484 \newcount\l@dpscR
1485

```

`\l@dsetuprawboxes` This macro creates `\maxchunks` pairs of boxes for left and right chunks. The boxes are called `\l@dLcolrawbox1`, `\l@dLcolrawbox2`, etc.

```

1486 \newcommand*\l@dsetuprawboxes){%
1487 \@l@dttempcntb=\l@dc@maxchunks
1488 \loop\ifnum\@l@dttempcntb>\z@
1489 \newnamebox{\l@dLcolrawbox\the\@l@dttempcntb}
1490 \newnamebox{\l@dRcolrawbox\the\@l@dttempcntb}
1491 \advance\@l@dttempcntb \m@ne

```

```

1492 \repeat}
1493
\l@dssetupmaxlinecounts To be able to synchronise left and right texts we need to know the maximum num-
\l@dzeromaxlinecounts ber of text lines there are in each pair of chunks. \l@dssetupmaxlinecounts creates
\maxchunks new counts called \l@dmaxlinesinpar1, etc., and \l@dzeromaxlinecounts zeroes all of them.
1494 \newcommand*{\l@dssetupmaxlinecounts}{%
1495 \l@ldtempcntb=\l@dc@maxchunks
1496 \loop\ifnum\l@ldtempcntb>\z@
1497 \newnamecount{\l@dmaxlinesinpar\the\l@ldtempcntb}
1498 \advance\l@ldtempcntb \m@ne
1499 \repeat}
1500 \newcommand*{\l@dzeromaxlinecounts}{%
1501 \begin{group}
1502 \l@ldtempcntb=\l@dc@maxchunks
1503 \loop\ifnum\l@ldtempcntb>\z@
1504 \global\usenamecount{\l@dmaxlinesinpar\the\l@ldtempcntb}=\z@
1505 \advance\l@ldtempcntb \m@ne
1506 \repeat
1507 \end{group}
1508

```

Make sure that all these are set up. This has to be done after the user has had an opportunity to change `\maxchunks`.

```

1509 \AtBeginDocument{%
1510 \l@dssetuppraxboxes
1511 \l@dssetupmaxlinecounts
1512 \l@dzeromaxlinecounts
1513 \l@dnumstartsl=\z@
1514 \l@dnumstartsr=\z@
1515 \l@dpscl=\z@
1516 \l@dpscr=\z@}
1517

```

## 22 Fixing babel

With parallel texts there is the possibility that the two sides might use different languages via `babel`. On the other hand, `babel` might not be called at all (even though it might be already built into the format).

With the normal sequential text each line is initially typeset in the current language environment, and then it is output at which time its attachments are typeset (in the same language environment). In the parallel case lines are typeset in their current language but an attachment might be typeset outside the language environment of its line if the left and right side languages are different. To counter this, we have to make sure that the correct language is used at the proper times.

```

\ifl@dusedbabel A flag for checking if babel has been used as a package.
\l@dusedbabelfalse 1518 \newif\ifl@dusedbabel
\l@dusedbabeltrue 1519 \l@dusedbabelfalse

\ifl@dsamelang A flag for checking if the same babel language has been used for both the left and
\l@dsamelangfalse right texts.
\l@dsamelangtrue 1520 \newif\ifl@dsamelang
1521 \l@dsamelangtrue

\l@dchecklang I'm going to use \theledlanguageL and \theledlanguageR to hold the names of
the languages used for the left and right texts. This macro sets \ifl@dsamelang
TRUE if they are the same, otherwise it sets it FALSE.
1522 \newcommand*{\l@dchecklang}{%
1523 \l@dsamelangfalse
1524 \edef\@tempa{\theledlanguageL}\edef\@tempb{\theledlanguageR}%
1525 \ifx\@tempa\@tempb
1526 \l@dsamelangtrue
1527 \fi}
1528

\l@dbbl@set@language In babel the macro \bbl@set@language{<lang>} does the work when the language
<lang> is changed via \selectlanguage. Unfortunately for me, if it is given an
argument in the form of a control sequence it strips off the \ character rather than
expanding the command. I need a version that accepts an argument in the form
\lang without it stripping the \.
1529 \newcommand*{\l@dbbl@set@language}[1]{%
1530 \edef\languageName{#1}%
1531 \select@language{\languageName}%
1532 \if@filesw
1533 \protected@write\auxout{}\string\select@language{\languageName}}%
1534 \addtocontents{toc}{\string\select@language{\languageName}}%
1535 \addtocontents{lof}{\string\select@language{\languageName}}%
1536 \addtocontents{lot}{\string\select@language{\languageName}}%
1537 \fi}
1538

The rest of the setup has to be postponed until the end of the preamble when
we know if babel has been used or not. However, for now assume that it has not
been used.

\selectlanguage \selectlanguage is a babel command. \theledlanguageL and \theledlanguageR
\l@duselanguage are the names of the languages of the left and right texts. \l@duselanguage is
\theledlanguageL similar to \selectlanguage.
\theledlanguageR 1539 \providecommand{\selectlanguage}[1]{%
1540 \newcommand*{\l@duselanguage}[1]{%
1541 \gdef\theledlanguageL{
1542 \gdef\theledlanguageR{
1543

```

Now do the `babel` fix or `polyglossia`, if necessary.

```
1544 \AtBeginDocument{%
1545 \ifundefined{xpg@main@language}{%
1546 \ifundefined{bbl@main@language}{%
```

Either `babel` has not been used or it has been used with no specified language.

```
1547 \l@dusedbabelfalse
1548 \renewcommand*{\selectlanguage}[1]{}%
```

Here we deal with the case where `babel` has been used. `\selectlanguage` has to be redefined to use our version of `\bbl@set@language` and to store the left or right language.

```
1549 \l@dusedbabeltrue
1550 \let\l@doldselectlanguage\selectlanguage
1551 \let\l@doldbbl@set@language\bbl@set@language
1552 \let\bbl@set@language\l@doldbbl@set@language
1553 \renewcommand{\selectlanguage}[1]{%
1554 \l@doldselectlanguage{#1}%
1555 \ifledRcol \gdef\theledlanguageR{#1}%
1556 \else \gdef\theledlanguageL{#1}%
1557 \fi}
```

`\l@duselanguage` simply calls the original `\selectlanguage` so that `\theledlanguageL` and `\theledlanguageR` are unaltered.

```
1558 \renewcommand*{\l@duselanguage}[1]{%
1559 \l@doldselectlanguage{#1}}
```

Lastly, initialise the left and right languages to the current `babel` one.

```
1560 \gdef\theledlanguageL{\bbl@main@language}%
1561 \gdef\theledlanguageR{\bbl@main@language}%
1562 }%
1563 }
```

If on `Polyglossia`

```
1564 { \apptocmd{\xpg@set@language}{%
1565 \ifledRcol \gdef\theledlanguageR{#1}%
1566 \else \gdef\theledlanguageL{#1}%
1567 \fi}%
1568 \let\l@duselanguage\xpg@set@language
1569 \gdef\theledlanguageL{\xpg@main@language}%
1570 \gdef\theledlanguageR{\xpg@main@language}%
1571 % \end{macrocode}
1572 % That's it.
1573 % \begin{macrocode}
1574 }}
```

## 23 Parallel columns

`\Columns` The `\Columns` command results in the previous Left and Right texts being typeset in matching columns. There should be equal numbers of chunks in the left and



right texts.

```

1575 \newcommand*{\Columns}{%
1576 \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else
1577 \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
1578 \fi

```

Start a group and zero counters, etc.

```

1579 \begingroup
1580 \l@dzeropenalties
1581 \endgraf\global\num@lines=\prevgraf
1582 \global\num@linesR=\prevgraf
1583 \global\par@line=\z@
1584 \global\par@lineR=\z@
1585 \global\l@dpscL=\z@
1586 \global\l@dpscR=\z@

```

Check if there are chunks to be processed, and process them two by two (left and right pairs).

```

1587 \check@pstarts
1588 \loop\if@pstarts

```

Increment `\l@dpscL` and `\l@dpscR` which here count the numbers of left and right chunks.

```

1589 \global\advance\l@dpscL \@ne
1590 \global\advance\l@dpscR \@ne

```

Check if there is text yet to be processed in at least one of the two current chunks, and also whether the left and right languages are the same

```

1591 \checkraw@text
1592 \l@dchecklang
1593 { \loop\ifaraw@text

```

Grab the next pair of left and right text lines and output them, swapping languages if they differ

```

1594 \ifl@dsamelang
1595 \do@lineL
1596 \do@lineR
1597 \else
1598 \l@duselanguage{\theledlanguageL}%
1599 \do@lineL
1600 \l@duselanguage{\theledlanguageR}%
1601 \do@lineR
1602 \fi
1603 \hb@xt@ \hsize{%
1604 \unhbox\l@dleftbox
1605 \hfill \columnseparator \hfill
1606 \unhbox\l@drightbox
1607 }%
1608 \checkraw@text
1609 \repeat}

```

Having completed a pair of chunks, write the number of lines in each chunk to the respective section files.

```
1610 \@writelinesinparL
1611 \@writelinesinparR
1612 \check@pstarts
1613 \repeat
```

Having output all chunks, make sure all notes have been output, then zero counts ready for the next set of texts.

```
1614 \flush@notes
1615 \flush@notesR
1616 \endgroup

1617 \global\l@dpscL=\z@
1618 \global\l@dpscR=\z@
1619 \global\l@dnpstartsL=\z@
1620 \global\l@dnpstartsR=\z@
1621 \ignorespaces}
1622
```

`\columnseparator` The separator between line pairs in parallel columns is in the form of a vertical rule extending a little below the baseline and with a height slightly greater than the `\baselineskip`. The width of the rule is `\columnrulewidth` (initially 0pt so the rule is invisible).

```
1623 \newcommand*{\columnseparator}{%
1624 \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}}
1625 \newdimen\columnrulewidth
1626 \columnrulewidth=\z@
1627
```

`\if@pstarts` `\check@pstarts` returns `\@pstartstrue` if there are any unprocessed chunks.

```
\@pstartstrue 1628 \newif\if@pstarts
\@pstartsfalse 1629 \newcommand*{\check@pstarts}{%
\check@pstarts 1630 \@pstartsfalse
1631 \ifnum\l@dnpstartsL>\l@dpscL
1632 \@pstartstrue
1633 \else
1634 \ifnum\l@dnpstartsR>\l@dpscR
1635 \@pstartstrue
1636 \fi
1637 \fi}
1638
```

`\ifaraw@text` `\checkraw@text` checks whether the current Left or Right box is void or not. If `\araw@texttrue` one or other is not void it sets `\araw@texttrue`, otherwise both are void and it sets `\araw@textfalse`.

```
\checkraw@text 1639 \newif\ifaraw@text
1640 \araw@textfalse
1641 \newcommand*{\checkraw@text}{%
```

```

1642 \araw@textfalse
1643 \ifvbox\namebox{l@dLcolrawbox\the\l@dpscL}
1644 \araw@texttrue
1645 \else
1646 \ifvbox\namebox{l@dRcolrawbox\the\l@dpscR}
1647 \araw@texttrue
1648 \fi
1649 \fi}
1650

```

`\@writelinesinparL` These write the number of text lines in a chunk to the section files, and then  
`\@writelinesinparR` afterwards zero the counter.

```

1651 \newcommand*{\@writelinesinparL}{%
1652 \edef\next{%
1653 \write\linenum@out{\string\@pend[\the\@donereallinesL]}}%
1654 \next
1655 \global\@donereallinesL \z@}
1656 \newcommand*{\@writelinesinparR}{%
1657 \edef\next{%
1658 \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}%
1659 \next
1660 \global\@donereallinesR \z@}
1661

```

## 24 Parallel pages

This is considerably more complicated than parallel columns.

`\numpagelinesL` Counts for the number of lines on a left or right page, and the smaller of the  
`\numpagelinesR` number of lines on a pair of facing pages.  
`\l@dminpagelines` 1662 \newcount\numpagelinesL  
1663 \newcount\numpagelinesR  
1664 \newcount\l@dminpagelines  
1665

`\Pages` The `\Pages` command results in the previous Left and Right texts being typeset  
on matching facing pages. There should be equal numbers of chunks in the left  
and right texts.

```

1666 \newcommand*{\Pages}{%
1667 \typeout{}
1668 \typeout{***** PAGES *****}
1669 \ifnum\l@dnumstartsL=\l@dnumstartsR\else
1670 \led@err@BadLeftRightPstarts{\the\l@dnumstartsL}{\the\l@dnumstartsR}%
1671 \fi

```

Get onto an empty even (left) page, then initialise counters, etc.

```

1672 \cleartol@evenpage
1673 \begingroup

```

```

1674 \l@dzeroopenalties
1675 \endgraf\global\num@lines=\prevgraf
1676 \global\num@linesR=\prevgraf
1677 \global\par@line=\z@
1678 \global\par@lineR=\z@
1679 \global\l@dpscL=\z@
1680 \global\l@dpscR=\z@
1681 \writtenlinesLfalse
1682 \writtenlinesRfalse

```

Check if there are chunks to be processed.

```

1683 \check@pstarts
1684 \loop\if@pstarts

```

Loop over the number of chunks, incrementing the chunk counts (\l@dpscL and \l@dpscR are chunk (box) counts.)

```

1685 \global\advance\l@dpscL \@ne
1686 \global\advance\l@dpscR \@ne

```

Calculate the maximum number of real text lines in the chunk pair, storing the result in the relevant \l@dmxlinesinpar.

```

1687 \getlinesfromparlistL
1688 \getlinesfromparlistR
1689 \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}%
1690 {\usernamecount{\l@dmxlinesinpar\the\l@dpscL}}%
1691 \check@pstarts
1692 \repeat

```

Zero the counts again, ready for the next bit.

```

1693 \global\l@dpscL=\z@
1694 \global\l@dpscR=\z@

```

Get the number of lines on the first pair of pages and store the minimum in \l@dminpagelines.

```

1695 \getlinesfrompagelistL
1696 \getlinesfrompagelistR
1697 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
1698 {\l@dminpagelines}%

```

Now we start processing the left and right chunks (\l@dpscL and \l@dpscR count the left and right chunks), starting with the first pair.

```

1699 \check@pstarts
1700 \if@pstarts

```

Increment the chunk counts to get the first pair.

```

1701 \global\advance\l@dpscL \@ne
1702 \global\advance\l@dpscR \@ne

```

We haven't processed any lines from these chunks yet, so zero the respective line counts.

```

1703 \global\@donereallinesL=\z@
1704 \global\@donetotallinesL=\z@

```

```

1705 \global\@donereallinesR=\z@
1706 \global\@donetotallinesR=\z@

```

Start a loop over the boxes (chunks).

```

1707 \checkraw@text
1708 % \begingroup
1709 { \loop\ifaraw@text

```

See if there is more that can be done for the left page and set up the left language.

```

1710 \checkpageL
1711 \l@duselanguage{\theledlanguageL}%
1712 %%% \begingroup
1713 { \loop\ifl@dsamepage

```

Process the next (left) text line, adding it to the page.

```

1714 \do@lineL
1715 \advance\numpagelinesL \@ne
1716 \ifshiftedverses
1717 \ifdim\ht\l@dleftbox>0pt\hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}\fi%
1718 \else
1719 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
1720 \fi

```

Perhaps we have to move to the next (left) box. Check if we have got all we can onto the page. If not, repeat for the next line.

```

1721 \get@nextboxL
1722 \checkpageL
1723 \repeat

```

That (left) page has been filled. Output the number of real lines on the page — if the page break is because the page has been filled with lines, use the actual number, otherwise the page has been ended early in order to synchronise with the facing page so use an impossibly large number.

```

1724 \ifl@dpagfull
1725 \@writelinesonpageL{\the\numpagelinesL}%
1726 \else
1727 \@writelinesonpageL{1000}%
1728 \fi

```

Zero the left page lines count and clear the page to get onto the facing (odd, right) page.

```

1729 \numpagelinesL \z@
1730 \clearl@dleftpage }%

```

Now do the same for the right text.

```

1731 \checkpageR
1732 \l@duselanguage{\theledlanguageR}%
1733 { \loop\ifl@dsamepage
1734 \do@lineR
1735 \advance\numpagelinesR \@ne

```

```

1736 \ifshiftedverses
1737 \ifdim\ht\l@drightbox>0pt\hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}\fi%
1738 \else
1739 \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
1740 \fi
1741 \get@nextboxR
1742 \checkpageR
1743 \repeat
1744 \ifl@dpagfull
1745 \@writelinesonpageR{\the\numpagelinesR}%
1746 \else
1747 \@writelinesonpageR{1000}%
1748 \fi
1749 \numpagelinesR=\z@

```

The page is full, so move onto the next (left, odd) page and repeat left text processing.

```

1750 \clearl@drightpage}

```

More to do? If there is we have to get the number of lines for the next pair of pages before starting to output them.

```

1751 \checkraw@text
1752 \ifaraw@text
1753 \getlinesfrompagelistL
1754 \getlinesfrompagelistR
1755 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
1756 {\l@dminpagelines}%
1757 \fi
1758 \repeat}

```

We have now output the text from all the chunks.

```

1759 \fi

```

Make sure that there are no inserts hanging around.

```

1760 \flush@notes
1761 \flush@notesR
1762 \endgroup

```

Zero counts ready for the next set of left/right text chunks.

```

1763 \global\l@dpscL=\z@
1764 \global\l@dpscR=\z@
1765 \global\l@dnumstartsL=\z@
1766 \global\l@dnumstartsR=\z@
1767 \ignorespaces}
1768

```

`\ledstrutL` Struts inserted into left and right text lines.

```

\ledstrutR 1769 \newcommand*{\ledstrutL}{\strut}
1770 \newcommand*{\ledstrutR}{\strut}
1771

```

`\cleartoevenpage` `\cleartoevenpage`, which is defined in the memoir class, is like `\clear(double)page`  
`\cleartol@devenpage` except that we end up on an even page. `\cleartol@devenpage` is similar except  
`\clearl@dleftpage` that it first checks to see if it is already on an empty page. `\clearl@dleftpage`  
`\clearl@drighpage` and `\clearl@drighpage` get us onto an odd and even page, respectively, checking  
that we end up on the immediately next page.

```

1772 \providecommand{\cleartoevenpage}[1][\@empty]{%
1773 \clearpage
1774 \ifodd\c@page\hbox{ }#1\clearpage\fi}
1775 \newcommand*{\cleartol@devenpage}{%
1776 \ifdim\pagetotal<\topskip% on an empty page
1777 \else
1778 \clearpage
1779 \fi
1780 \ifodd\c@page\hbox{ }\clearpage\fi}
1781 \newcommand*{\clearl@dleftpage}{%
1782 \clearpage
1783 \ifodd\c@page\else
1784 \led@err@LeftOnRightPage
1785 \hbox{ }%
1786 \cleardoublepage
1787 \fi}
1788 \newcommand*{\clearl@drighpage}{%
1789 \clearpage
1790 \ifodd\c@page
1791 \led@err@RightOnLeftPage
1792 \hbox{ }%
1793 \cleartoevenpage
1794 \fi}
1795
\getlinesfromparlistL \getlinesfromparlistL gets the next entry from the \linesinpar@listL and
\cs@linesinparL puts it into \cs@linesinparL; if the list is empty, it sets \cs@linesinparL to
\getlinesfromparlistR 0. Similarly for \getlinesfromparlistR.
\cs@linesinparL 1796 \newcommand*{\getlinesfromparlistL}{%
1797 \ifx\linesinpar@listL\empty
1798 \gdef\cs@linesinparL{0}%
1799 \else
1800 \gl@p\linesinpar@listL\to\cs@linesinparL
1801 \fi}
1802 \newcommand*{\getlinesfromparlistR}{%
1803 \ifx\linesinpar@listR\empty
1804 \gdef\cs@linesinparR{0}%
1805 \else
1806 \gl@p\linesinpar@listR\to\cs@linesinparR
1807 \fi}
1808
\getlinesfrompagelistL \getlinesfrompagelistL gets the next entry from the \linesonpage@listL and
\cs@linesonpageL puts it into \cs@linesonpageL; if the list is empty, it sets \cs@linesonpageL
\getlinesfrompagelistR
\cs@linesonpageR

```

```

1809 \newcommand*{\getlinesfrompagelistL}{%
1810 \ifx\linesonpage@listL\empty
1811 \gdef\cs@linesonpageL{1000}%
1812 \else
1813 \glp\linesonpage@listL\to\cs@linesonpageL
1814 \fi}
1815 \newcommand*{\getlinesfrompagelistR}{%
1816 \ifx\linesonpage@listR\empty
1817 \gdef\cs@linesonpageR{1000}%
1818 \else
1819 \glp\linesonpage@listR\to\cs@linesonpageR
1820 \fi}
1821

```

```

1822 \newcommand*{\@writelinesonpageL}[1]{%
1823 \edef\next{\write\linenum@out{\string\@lopL{#1}}}%
1824 \next}
1825 \newcommand*{\@writelinesonpageR}[1]{%
1826 \edef\next{\write\linenum@outR{\string\@lopR{#1}}}%
1827 \next}
1828

```

Similarly `\l@dcalc@minoftwo{⟨num⟩}{⟨num⟩}{⟨count⟩}` sets  $\langle count \rangle$  to the minimum of the two  $\langle num \rangle$ .

```

1829 \newcommand*{\l@dcalc@maxoftwo}[3]{%
1830 \ifnum #2>#1\relax
1831 #3=#2\relax
1832 \else
1833 #3=#1\relax
1834 \fi}
1835 \newcommand*{\l@dcalc@minoftwo}[3]{%
1836 \ifnum #2<#1\relax
1837 #3=#2\relax
1838 \else
1839 #3=#1\relax
1840 \fi}
1841

```

|                                |                                                                                                         |
|--------------------------------|---------------------------------------------------------------------------------------------------------|
| <code>\ifl@dsamepage</code>    | <code>\checkpageL</code> tests if the space and lines already taken on the page by text and foot-       |
| <code>\l@dsamepagetrue</code>  | notes is less than the constraints. If so, then <code>\ifl@dpagefull</code> is set FALSE and            |
| <code>\l@dsamepagefalse</code> | <code>\ifl@dsamepage</code> is set TRUE. If the page is spatially full then <code>\ifl@dpagefull</code> |
| <code>\ifl@dpagefull</code>    | is set TRUE and <code>\ifl@dsamepage</code> is set FALSE. If it is not spatially full but               |
| <code>\l@dpagefulltrue</code>  | the maximum number of lines have been output then both <code>\ifl@dpagefull</code> and                  |
| <code>\l@dpagefullfalse</code> | <code>\ifl@dsamepage</code> are set FALSE.                                                              |
| <code>\checkpageL</code>       |                                                                                                         |
| <code>\checkpageR</code>       |                                                                                                         |



```

1842 \newif\ifl@dsamepage
1843 \l@dsamepagetrue
1844 \newif\ifl@dpagfull
1845 \newcommand*{\checkpageL}{%
1846 \l@dpagfulltrue
1847 \l@dsamepagetrue
1848 \check@goal
1849 \ifdim\pagetotal<\ledthegoal
1850 \ifnum\numpagelinesL<\l@dminpagelines
1851 \else
1852 \l@dsamepagefalse
1853 \l@dpagfullfalse
1854 \fi
1855 \else
1856 \l@dsamepagefalse
1857 \l@dpagfulltrue
1858 \fi}
1859 \newcommand*{\checkpageR}{%
1860 \l@dpagfulltrue
1861 \l@dsamepagetrue
1862 \check@goal
1863 \ifdim\pagetotal<\ledthegoal
1864 \ifnum\numpagelinesR<\l@dminpagelines
1865 \else
1866 \l@dsamepagefalse
1867 \l@dpagfullfalse
1868 \fi
1869 \else
1870 \l@dsamepagefalse
1871 \l@dpagfulltrue
1872 \fi}
1873

```

`\ledthegoal` `\ledthegoal` is the amount of space allowed to taken by text and footnotes on  
`\goalfraction` a page before a forced pagebreak. This can be controlled via `\goalfraction`.  
`\check@goal` `\ledthegoal` is calculated via `\check@goal`.

```

1874 \newdimen\ledthegoal
1875 \ifshiftedverses
1876 \newcommand*{\goalfraction}{0.95}
1877 \else
1878 \newcommand*{\goalfraction}{0.9}
1879 \fi
1880
1881 \newcommand*{\check@goal}{%
1882 \ledthegoal=\goalfraction\pagegoal}
1883

```

`\ifwrittenlinesL` Booleans for whether line data has been written to the section file.  
`\ifwrittenlinesL` 1884 \newif\ifwrittenlinesL

```

1885 \newif\ifwrittenlinesR
1886

\get@nextboxL If the current box is not empty (i.e., still contains some lines) nothing is done.
\get@nextboxR Otherwise if and only if a synchronisation point is reached the next box is started.

1887 \newcommand*{\get@nextboxL}{%
1888 \ifvbox\namebox{\l@dLcolrawbox\the\l@dpscL}% box is not empty

 The current box is not empty; do nothing.

1889 \else% box is empty

 The box is empty; check if enough lines (real and blank) have been output.
1890 \ifnum\usernamecount{\l@dmaxlinesinpar\the\l@dpscL}>\@donetotallinesL
1891 \else

 Sufficient lines have been output.
1892 \ifwrittenlinesL
1893 \else

 Write out the number of lines done, and set the boolean so this is only done once.
1894 \@writelinesinparL
1895 \writtenlinesLtrue
1896 \fi
1897 \ifnum\l@dnumstartsL>\l@dpscL

 There are still unprocessed boxes. Recalculate the maximum number of lines
 needed, and move onto the next box (by incrementing \l@dpscL).
1898 \writtenlinesLfalse
1899 \l@dcalc@maxoftwo{\the\usernamecount{\l@dmaxlinesinpar\the\l@dpscL}}%
1900 {\the\@donetotallinesL}%
1901 {\usernamecount{\l@dmaxlinesinpar\the\l@dpscL}}%
1902 \global\@donetotallinesL \z@
1903 \global\advance\l@dpscL \@ne
1904 \fi
1905 \fi
1906 \fi}

1907 \newcommand*{\get@nextboxR}{%
1908 \ifvbox\namebox{\l@dRcolrawbox\the\l@dpscR}% box is not empty
1909 \else% box is empty

 \ifnum\usernamecount{\l@dmaxlinesinpar\the\l@dpscR}>\@donetotallinesR
1910 \else
1911 \else
1912 \ifwrittenlinesR
1913 \else
1914 \@writelinesinparR
1915 \writtenlinesRtrue
1916 \fi
1917 \ifnum\l@dnumstartsR>\l@dpscR
1918 \writtenlinesRfalse
1919 \l@dcalc@maxoftwo{\the\usernamecount{\l@dmaxlinesinpar\the\l@dpscR}}%
1920 {\the\@donetotallinesR}%
1921 {\usernamecount{\l@dmaxlinesinpar\the\l@dpscR}}%

```

```
1922 \global\@donetotallinesR \z@
1923 \global\advance\l@dpscr \@ne
1924 \fi
1925 \fi
1926 \fi}
1927
```

## 25 The End

i/code<sub>i</sub>

## A Examples

This section presents some sample documents.

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 villon
> latex villon
> latex villon
> dvips -E -o villon.eps villon % produces villon.eps
> epstopdf villon.eps % produces villon.pdf
```

For a multipage example, DVIPS has an option to output a range of pages (-p for the first and -l (letter l) for the last). For instance, to output a single page, say page 2:

```
> latex djd17nov
> latex djd17nov
> latex djd17nov
> dvips -E -p2 -l2 -o djd17novL.eps djd17nov % produces djd17novL.eps
> epstopdf djd17novL.eps % produces djd17novL.pdf
```

For those who aren't fascinated by LaTeX code, I show the all the typeset results first, then the code that produced them.

I thought that limericks were peculiarly English, but this appears not to be the case. As with most limericks this one is by Anonymous.

|   |                                     |                                       |   |
|---|-------------------------------------|---------------------------------------|---|
|   | Il y avait un jeune homme de Dijon, | There was a young man of Dijon,       | 1 |
| 2 | Qui n'avait que peu de religion.    | Who had only a little religion,       |   |
|   | Il dit: 'Quant à moi,               | He said: 'As for me,                  | 3 |
| 4 | Je déteste tous les trois,          | I detest all the three,               |   |
|   | Le Père, et le Fils, et le Pigeon.' | The Father, the Son, and the Pigeon.' | 5 |

The following is verse LXXIII of François Villon's *Le Testament* (The Testament), composed in 1461.

|   |                                        |                                         |    |
|---|----------------------------------------|-----------------------------------------|----|
|   | Dieu mercy et Tacque Thibault,         | Thanks to God — and to Tacque Thibaud   |    |
| 2 | Qui tant d'eaue froid m'a fait boire,  | Who made me drink so much cold water,   | 2r |
|   | Mis en bas lieu, non pas en hault,     | Put me underground instead of higher up |    |
| 4 | Mengier d'angoisse maints poire,       | And made me eat such bitter fruit,      | 4r |
|   | Enferré ... Quant j'en ay memoire,     | In chains ... When I think of this,     |    |
| 6 | Je Prie pour luy <i>et reliqua</i> ,   | I pray for him— <i>et reliqua</i> ;     | 6r |
|   | Que Dieu luy doint, et voire, voire!   | May God grant him (yes, by God)         |    |
| 8 | Ce que je pense ... <i>et cetera</i> . | What I think ... <i>et cetera</i> .     | 8r |

The translation and notes are by Anthony Bonner, *The Complete Works of François Villon*, published by Bantam Books in 1960.

---

4 poire d'angoisse] This has a triple meaning: literally it is the fruit of the choke pear, figuratively it means 'bitter fruit', and it also refers to a torture instrument.  
6 *et reliqua*] and so on

---

1r Tacque Thibaud] A favourite of Jean, Duc de Berry and loathed for his exactions and debauchery. Villon uses his name as an insulting nickname for Thibaud d'Auxigny, the Bishop of Orléans.

2r cold water] Can either refer to the normal prison diet of bread and water or to a common medieval torture which involved forced drinking of cold water.

Figure 1: Output from `villon.tex`.

## 1 De ecclesia S. Stephani Novimagensi

Nobilis itaque comes Otto imperio et dominio Novimagensi sibi, ut praefer-  
tur, impignoratis et commissis proinde praeesse cupiens, anno LIII superius  
descripto, mense Iunio, una cum iudice, scabinis ceterisque civibus civitatis  
Novimagensis, pro ipsius et inhabitantium in ea necessitate, commodo et utili-  
tate, ut ecclesia eius parochialis extra civitatem sita destrueretur et infra muros  
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  
gravamen iurium et bonorum eorundem, impetravit.

Et exinde liberum locum eiusdem civitatis qui dicitur Hundisburg, de praeli-  
bati 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  
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 con-  
tiguam, ipsi plebano darent et assignarent. Et desuper apud dictam ecclesiam  
sanctorum Apostolorum est littera sigillis ipsorum Ottonis comitis et civitatis  
Novimagensis sigillata.

// One additional line to show synchronization. //

---

3 p. 227 R 4 p. 97 N 6 p. 129 D 12 f. 72v M 13 p. 228 R 20 p. 130 D

2 proinde] primum D 5 ecclesia eius] ecclesia D: eius eius H extra civitatem om. H  
infra] intra D 6 transferretur] transferreretur NH 7 Hofsteden] Hoffstede D: Hoffsteden  
H Coloniensi] Colononiensi H dominis] viris H 8 Coloniensi] Coloniae H 10 iurium]  
virium D 11 liberum] librum H qui] quae D Hundisburg] Hundisburch D: Hundisbrug  
HMN: 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. Apostles’  
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 indulgemus...”  
11–19 Cf. Sloet, *Oorkondenboek* nr. 762 (June 1254)

# 1 St. Stephen's Church in Nijmegen

After the noble count Otto had taken in pledge the power over Nijmegen,<sup>1</sup> like  
 I have written above, he wanted to protect the town. So in June 1254 he and 1254  
 the judge, the sheriffs and other citizens of Nijmegen obtained permission to  
 demolish the parish church that lay outside the town walls,<sup>2</sup> to move it inside  
 5 the walls and to rebuild it new. This operation was necessary and useful both for  
 Otto himself and for the inhabitants of the town. The reverend father Conrad of  
 Hochstaden, archbishop of Cologne,<sup>3</sup> gave his permission. So did the reverend  
 dean and canons of the chapter of St. Apostles' in Cologne, who had long<sup>4</sup> been  
 the true and benevolent patrons of the church—but they did not allow Otto to  
 10 do anything without their knowledge, nor to infringe their rights, nor to damage  
 their property.

And so the count and the town voluntarily gave an open space in town called  
 Hundisburg, which was owned by the aforementioned king William, to the dean  
 and chapter of St. Apostles' in order to build and consecrate a church and grave-  
 15 yard. King William approved and the town of Nijmegen explicitly expressed its  
 assent. A new ditch was dug on property of the church near the castle and the  
 harbour,<sup>5</sup> causing the demolition of the presbytery. In compensation, the count  
 and citizens committed themselves to giving the parish priest another suitable  
 space close enough to the new church that was about to be built. A letter about  
 20 these transactions, with the seals of count Otto and the town of Nijmegen, is  
 kept at St. Apostles' church.<sup>6</sup>

// One additional line to show synchronization. //

---

<sup>1</sup>In 1247 William II (1227–1256) count of Holland needed money to fight his way to Aachen to be crowned King of the Holy Roman Empire. He gave the town of Nijmegen in pledge to Otto II (1229–1271) count of Guelders.

<sup>2</sup>Since the early seventh century old St. Stephen's church had been located close to the castle, at today's Kelfkensbos square. Traces of the church and the presbytery were found during excavations in 1998–1999.

<sup>3</sup>Conrad of Hochstaden († 1261) was archbishop of Cologne in 1238–1261. Nijmegen belonged to the archdiocese of Cologne until 1559.

<sup>4</sup>They probably became the patrons when the chapter was established in the early eleventh century. About the church and the chapter, see Gottfried Stracke, *Köln: St. Aposteln*, Stadtspuren – Denkmäler in Köln, vol. 19, Köln: J. P. Bachem, 1992.

<sup>5</sup>Nowadays, the exact location of the medieval ditch—and of two Roman ones—can be seen in the pavement of Kelfkensbos square.

<sup>6</sup>The original letter is lost. A 15th century transcription of it is kept at the Historisches Archiv der Stadt Köln (HASTK).

1 Arma gravi numero violentaque bella parabam  
 2 edere, materiā conveniente modis.  
 3 Par erat inferior versus—rissime Cupido  
 4 dicitur atque unum surripuisse pedem.  
 -----  
 5 “Quis tibi, saeve puer, dedit hoc in carmina iuris?  
 6 Pieridum vates, non tua turba sumus.  
 7 Quid si praecripiat flavae Vēnus arma Minervae,  
 8 ventilet accensas flava Minerva faces?  
 -----  
 9 Quis probet in silvis Cererem regnare iugosis,  
 10 lege pharetratae Virginis arva coli?  
 11 Crinibus insignem quis acuta cuspide Phoebum  
 -----  
 12 instruat, Aoniam Marte movente lyram?  
 -----

---

6 sumus] note lost 11 acuta] acutā (abl. abs.)

Figure 4: First left page output from `djdpoe.ms.tex`.



<sup>1</sup>I.e., the even lines, which were hexameters (with six feet) became pentameters (with five feet).

<sup>2</sup>Muses

<sup>3</sup>Ceres was the Roman goddess of the harvest.

<sup>4</sup>By '*Virgo*' ('Virgin') Ovid means Diana, the Roman goddess of the hunt.

<sup>5</sup>Lines 7R-12R show some paradoxical situations that would occur if the gods didn't stay with their own business.

12R Aonian] Mount Parnassus, where the Muses live, is located in Aonia.

1 Arma gravi numero violentaque bella parabam  
 2 edere, materiā conveniente modis.  
 3 Par erat inferior versus—risisse Cupido  
 4 dicitur atque unum surripuisse pedem.  
 —————  
 5 “Quis tibi, saeve puer, dedit hoc in carmina iuris?  
 6 Pieridum vates, non tua turba sumus.  
 7 Quid si praeripiat flavae Vēnus arma Minervae,  
 8 ventilet accensas flava Minerva faces?  
 —————  
 9 Quis probet in silvis Cererem regnare iugosis,  
 10 lege pharetratae Virginis arva coli?  
 11 Crinibus insignem quis acuta cuspidē Phoebum  
 12 instruat, Aoniam Marte movente lyram?  
 —————

---

6 sumus] note lost 11 acuta] acutā (abl. abs.)

Figure 6: Second left page output from `djdpoe.ms.tex`.

<sup>6</sup>I.e., the even lines, which were hexameters (with six feet) became pentameters (with five feet).

<sup>7</sup>Muses

<sup>8</sup>Ceres was the Roman goddess of the harvest.

<sup>9</sup>By '*Virgo*' ('Virgin') Ovid means Diana, the Roman goddess of the hunt.

<sup>10</sup>Lines 7R-12R show some paradoxical situations that would occur if the gods didn't stay with their own business.

12R Aonian] Mount Parnassus, where the Muses live, is located in Aonia.

## A.1 Parallel column example

This made-up example, `villon.tex`, is included to show parallel columns and how they can be interspersed in regular text. The verses are set using the `\stanza` construct, where each verse line is a chunk. The code is given below and the result is shown in Figure 1.

---

```

1928 (*villon)
1929 %% villon.tex Example parallel columns
1930 \documentclass{article}
1931 \addtolength{\textheight}{-10\baselineskip}
1932 \usepackage{ledmac,ledpar}
1933 %% Use r instead of R to flag right text line numbers
1934 \renewcommand{\Rlineflag}{r}
1935 %% Use the flag in the notes
1936 \let\oldBfootfmt\Bfootfmt
1937 \renewcommand{\Bfootfmt}[3]{%
1938 \let\printlines\printlinesR
1939 \oldBfootfmt{#1}{#2}{#3}}
1940 \begin{document}
1941
1942 I thought that limericks were peculiarly English, but this appears not
1943 to be the case. As with most limericks this one is by Anonymous.
1944
1945 \vspace*{\baselineskip}
1946
1947 \begin{pairs}
1948 %% no indentation
1949 \setstanzaindents{0,0,0,0,0,0,0,0,0}
1950 %% no number flag
1951 \renewcommand{\Rlineflag}{}
1952 %% draw a rule and widen the columns
1953 \setlength{\columnrulewidth}{0.4pt}
1954 \setlength{\Lcolwidth}{0.46\textwidth}
1955 \setlength{\Rcolwidth}{\Lcolwidth}
1956
1957 \begin{Leftside}
1958 %% set left text line numbering sequence
1959 \firstlinenum{2}
1960 \linenumincrement{2}
1961 \linenummargin{left}
1962 \beginnumbering
1963 \stanza
1964 Il y avait un jeune homme de Dijon, &
1965 Qui n'avait que peu de religion. &
1966 Il dit: 'Quant \'{a} moi, &
1967 Je d'\{e\}teste tous les trois, &
1968 Le P\{e\}re, et le Fils, et le Pigeon.' \&
1969 \endnumbering
1970 \end{Leftside}

```

```

1971
1972 \begin{Rightside}
1973 %% different right text line numbering sequence
1974 \firstlinenum{1}
1975 \linenumincrement{2}
1976 \linenummargin{right}
1977 \beginnumbering
1978 \stanza
1979 There was a young man of Dijon, &
1980 Who had only a little religion, &
1981 He said: 'As for me, &
1982 I detest all the three, &
1983 The Father, the Son, and the Pigeon.' \&
1984 \endnumbering
1985 \end{Rightside}
1986
1987 \Columns
1988 \end{pairs}
1989
1990 \vspace*{\baselineskip}
1991
1992 The following is verse \textsc{lxixiii} of Fran\c{c}ois Villon's
1993 \textit{Le Testament} (The Testament), composed in 1461.
1994
1995 %% Allow for hanging indentation for long lines
1996 \setstanzaindents{1,0,0,0,0,0,0,0}
1997 %% Columns wider than the default
1998 \setlength{\Lcolwidth}{0.46\textwidth}
1999 \setlength{\Rcolwidth}{\Lcolwidth}
2000 \vspace*{\baselineskip}
2001
2002 \begin{pairs}
2003 \begin{Leftside}
2004 \firstlinenum{2}
2005 \linenumincrement{2}
2006 \linenummargin{left}
2007 \beginnumbering
2008 \stanza
2009 Dieu mercy et Tacque Thibault, &
2010 Qui tant d'eaue froid m'a fait boire, &
2011 Mis en bas lieu, non pas en hault, &
2012 Mengier d'angoisse maints \edtext{poire}{\lemma{poire d'angoisse}}%
2013 \Afootnote{This has a triple meaning: literally it is the fruit of the
2014 choke pear,
2015 figuratively it means 'bitter fruit', and it also refers to a torture
2016 instrument.}}, &
2017 Enferr'\{e} \ldots Quant j'en ay memoire, &
2018 Je Prie pour luy \edtext{\textit{et reliqua}}{\Afootnote{and so on}}, &
2019 Que Dieu luy doint, et voire, voire! &
2020 Ce que je pense \ldots \textit{et cetera}. \&

```

```

2021 \endnumbering
2022 \end{Leftside}
2023
2024 \begin{Rightside}
2025 \firstlinenum{2}
2026 \linenumincrement{2}
2027 \linenummargin{right}
2028 \beginnumbering
2029 \stanza
2030 Thanks to God --- and to \edtext{Tacque Thibaud}{%
2031 \Bfootnote{A favourite of Jean, Duc de Berry and loathed for his exactions
2032 and debauchery. Villon uses his name as an insulting nickname for
2033 Thibaud d'Auxigny, the Bishop of Orl\{'e}ans.}} &
2034 Who made me drink so much \edtext{cold water}{%
2035 \Bfootnote{Can either refer to the normal prison diet of bread and
2036 water or to a common medieval torture which involved forced drinking
2037 of cold water.}}, &
2038 Put me underground instead of higher up &
2039 And made me eat such bitter fruit, &
2040 In chains \ldots When I think of this, &
2041 I pray for him---\textit{et reliqua;} &
2042 May God grant him (yes, by God) &
2043 What I think \ldots \textit{et cetera}. \&
2044 \endnumbering
2045 \end{Rightside}
2046
2047 \Columns
2048 \end{pairs}
2049
2050 \vspace*{\baselineskip}
2051
2052 The translation and notes are by Anthony Bonner,
2053 \textit{The Complete Works of Fran\c{c}ois Villon}, published by
2054 Bantam Books in 1960.
2055
2056 \end{document}
2057
2058 \</villon>

```

## A.2 Example parallel facing pages

This example, illustrated in Figures 2 and 3, was provided in November 2004 by Dirk-Jan Dekker of the Department of Medieval History at Radboud University, Nijmegen.

---

```

2059 (*djd17nov)
2060 %%% This is djd17nov.tex, a sample critical text edition
2061 %%% written in LaTeX2e with the ledmac and ledpar packages.
2062 %%% (c) 2003--2004 by Dr. Dirk-Jan Dekker,

```

```

2063 %% Radboud University, Nijmegen (The Netherlands)
2064 %% (PRW) Modified slightly by PRW to fit the ledpar manual
2065
2066 \documentclass[10pt, letterpaper, twoside]{article}
2067 \usepackage[latin,english]{babel}
2068 \usepackage{makeidx}
2069 \usepackage{ledmac,ledpar}
2070 \lineation{section}
2071 \linenummargin{inner}
2072 \sidenotemargin{outer}
2073
2074 \makeindex
2075
2076 \renewcommand{\notenumfont}{\footnotesize}
2077 \newcommand{\notetextfont}{\footnotesize}
2078
2079 %\let\Afootnoterule=\relax
2080 \let\Bfootnoterule=\relax
2081 \let\Cfootnoterule=\relax
2082
2083 \addtolength{\skip\Afootins}{1.5mm}
2084 %\addtolength{\skip\Bfootins}{1.5mm}
2085 %\addtolength{\skip\Cfootins}{1.5mm}
2086
2087 \makeatletter
2088
2089 \renewcommand*{\para@vfootnote}[2]{%
2090 \insert\csname #1footins\endcsname
2091 \bgroup
2092 \notefontsetup
2093 \interlinepenalty=\interfootnotelinepenalty
2094 \floatingpenalty=\@MM
2095 \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2096 \leftskip=\z@skip \rightskip=\z@skip
2097 \l@dparssefootsec #2\ledplinenumtrue% new from here
2098 \ifnum\@nameuse{previous@#1@number}=\l@dparsestartline\relax
2099 \ledplinenumfalse
2100 \fi
2101 \ifnum\previous@page=\l@dparsestartpage\relax
2102 \else \ledplinenumtrue \fi
2103 \ifnum\l@dparsestartline=\l@dparseendline\relax
2104 \else \ledplinenumtrue \fi
2105 \expandafter\xdef\csname previous@#1@number\endcsname{\l@dparsestartline}%
2106 \xdef\previous@page{\l@dparsestartpage}% to here
2107 \setbox0=\vbox{\hsize=\maxdimen
2108 \noindent\csname #1footfmt\endcsname#2}%
2109 \setbox0=\hbox{\unvvh0}%
2110 \dp0=0pt
2111 \ht0=\csname #1footfudgefactor\endcsname\wd0
2112 \box0

```

```

2113 \penalty0
2114 \egroup
2115 }
2116
2117 \newcommand*{\previous@A@number}{-1}
2118 \newcommand*{\previous@B@number}{-1}
2119 \newcommand*{\previous@C@number}{-1}
2120 \newcommand*{\previous@page}{-1}
2121
2122 \newcommand{\abb}[1]{#1%
2123 \let\rbracket\nobrak\relax}
2124 \newcommand{\nobrak}{\textnormal{}}
2125 \newcommand{\morenoexpands}{%
2126 \let\abb=0%
2127 }
2128
2129 \newcommand{\Aparafootfmt}[3]{%
2130 \ledsetnormalparstuff
2131 \scriptsize
2132 \notenumfont\printlines#1\enspace
2133 % \lemmafont#1/#2\enskip
2134 \notetextfont
2135 #3\penalty-10\hskip 1em plus 4em minus.4em\relax}
2136
2137 \newcommand{\Bparafootfmt}[3]{%
2138 \ledsetnormalparstuff
2139 \scriptsize
2140 \notenumfont\printlines#1/%
2141 \ifledplinenum
2142 \enspace
2143 \else
2144 {\hskip 0em plus 0em minus .3em}%
2145 \fi
2146 \select@lemmafont#1/#2\rbracket\enskip
2147 \notetextfont
2148 #3\penalty-10\hskip 1em plus 4em minus.4em\relax }
2149
2150 \newcommand{\Cparafootfmt}[3]{%
2151 \ledsetnormalparstuff
2152 \scriptsize
2153 \notenumfont\printlines#1\enspace
2154 % \lemmafont#1/#2\enskip
2155 \notetextfont
2156 #3\penalty-10\hskip 1em plus 4em minus.4em\relax}
2157
2158 \makeatother
2159
2160 \footparagraph{A}
2161 \footparagraph{B}
2162 \footparagraph{C}

```



```

2163
2164 \let\Afootfmt=\Aparafootfmt
2165 \let\Bfootfmt=\Bparafootfmt
2166 \let\Cfootfmt=\Cparafootfmt
2167
2168 \renewcommand*{\Rlineflag}{}
2169
2170 \emergencystretch40pt
2171
2172 \author{Guillelmus de Berchen}
2173 \title{Chronicon Geldriae}
2174 \date{}
2175 \hyphenation{archi-epi-sco-po Huns-dis-brug li-be-ra No-vi-ma-gen-si}
2176 \begin{document}
2177 \begin{pages}
2178 \begin{Leftside}
2179 \beginnumbering\pstart
2180 \selectlanguage{latin}
2181 \section{De ecclesia S. Stephani Novimagensi}
2182
2183 \noindent\setline{1}
2184 Nobilis itaque comes Otto\protect\edindex{Otto II of Guelders}
2185 imperio et dominio Novimagensi sibi, ut praefertur, impignoratis
2186 et commissis
2187 \edtext{proinde}{\Bfootnote{primum D}} praeesse cupiens, anno
2188 \textsc{liiii} superius descripto, mense
2189 Iu\edtext{}{\Afootnote{p.\ 227~R}}nio, una cum iudice, scabinis ceterisque
2190 civibus civitatis Novimagensis, pro ipsius et inhabitantium in ea
2191 necessitate,\edtext{}{\Afootnote{p.\ 97~N}} commodo et utilitate,
2192 ut \edtext{ecclesia eius}{\Bfootnote{ecclesia D: eius eius H}} parochialis
2193 \edtext{\abb{extra civitatem}}{\Bfootnote{\textit{om.}~H}} sita
2194 destrueretur et \edtext{infra}{\Bfootnote{intra D}} muros
2195 \edtext{transfer}\edtext{}{\Afootnote{p.\ 129~D}}retur}%
2196 {\Bfootnote{transferreretur NH}}
2197 ac de novo construeretur,
2198 \edtext{a reverendo patre domino
2199 Conrado\protect\edindex{Conrad of Hochstaden} de
2200 \edtext{Hofsteden}{\Bfootnote{Hoffstede D: Hoffsteden H}}, archiepiscopo
2201 \edtext{Coloniensi}{\Bfootnote{Colononiensi H}}, licentiam}%
2202 {\Cfootnote{William is confusing two charters that are five years
2203 apart. Permission from St.\ Apostles' Church in Cologne had been
2204 obtained as early as 1249. Cf.\
2205 Sloet\protect\index{Sloet van de Beele, L.A.J.W.},
2206 \textit{Oorkondenboek} nr.\ 707 (14 November 1249):
2207 ‘‘\ldots{nos devotionis tue precibus annuentes, ut ipsam ecclesiam
2208 faciens demoliri transferas in locum alium competentem, tibi
2209 auctoritate presentium indulgemus\ldots’’}}, et a venerabilibus
2210 \edtext{dominis}{\Bfootnote{viris H}} decano et capitulo sanctorum
2211 Apostolorum\protect\edindex{St. Apostles' (Cologne)}
2212 \edtext{Coloniensi}{\Bfootnote{Coloniae H}}, ipsius ecclesiae ab

```

```

2213 antiquo veris et pacificis patronis, consensum, citra tamen
2214 praeiudicium, damnum aut gravamen \edtext{iurium}{\Bfootnote{virium D}}
2215 et bonorum eorundem, impetravit.
2216 \pend
2217
2218 \pstart
2219 \edtext{Et exinde \edtext{liberum}{\Bfootnote{librum H}}}
2220 locum eiusdem civitatis
2221 \edtext{qui}{\Bfootnote{quae D}} dicitur
2222 \edtext{Hundisburg}{\Bfootnote{Hundisburch D: Hundisbrug HMN:
2223 Hunsdisbrug R}}\protect\edindex{Hundisburg},
2224 de praelibati Wilhelmi\protect\edindex{William II of Holland} Romanorum
2225 \edtext{regis}{\Bfootnote{imperatoris D}}, ipsius fundi
2226 do\edtext{}{\Afootnote{f.\ 72v~M}}mini, consensu, ad aedificandum
2227 \edtext{\abb{et consecrandum}}{\Bfootnote{\textit{om.}\ H}}
2228 ecclesi\edtext{}{\Afootnote{p.\ 228~R}}am et coemeterium,
2229 \edtext{eisdem}{\Bfootnote{eiusdem D}} decano et capitulo de expresso
2230 eiusdem civitatis assensu libera contradiderunt voluntate, obligantes
2231 se ipsi \edtext{comes}{\Bfootnote{comites D}} et civitas
2232 \edtext{\abb{dictis}}{\Bfootnote{\textit{om.}\ H}} decano et capitulo,
2233 quod in recompensationem illius areae infra castrum et portam, quae
2234 fuit dos ecclesiae, in qua plebanus habitare solebat---quae
2235 \edtext{tunc}{\Bfootnote{nunc H}} per novum fossatum civitatis est
2236 destructa---aliam aream competentem et ecclesiae novae,
2237 \edtext{ut praeferitur, aedificandae}{\%
2238 \lemma{\abb{ut\ldots aedificandae}}{\Bfootnote{\textit{om.}\ H}} satis
2239 \edtext{contiguam}{\Bfootnote{contiguum M}}, ipsi plebano darent et
2240 assignarent.}{\Cfootnote{Cf.\ Sloet, \textit{Oorkondenboek} nr.\ 762
2241 (June 1254)}} Et desuper
2242 \edtext{\abb{apud}}{\Bfootnote{\textit{om.}\ H}} dictam ecclesiam
2243 sanctorum Apostolorum \edtext{est}{\Bfootnote{et H}}
2244 \edtext{littera}{\Bfootnote{litteram H}} sigillis ipsorum
2245 Ottonis\edtext{}{\Afootnote{p.\ 130~D}} comitis et civitatis
2246 \edtext{Novimagensis}{\Bfootnote{Novimagii D}}
2247 \edtext{sigillata}{\Bfootnote{sigillis communita H}}.
2248 \pend
2249
2250 \pstart
2251 // One additional line to show synchronization. //
2252 \pend
2253 \endnumbering
2254 \end{Leftside}
2255
2256 \begin{Rightside}
2257 \sidenotemargin{right}\selectlanguage{english}
2258 \beginnumbering
2259 \pstart
2260 \addtocounter{section}{-1}%
2261 \leavevmode\section{St.\ Stephen's Church in Nijmegen}
2262

```

2263 \noindent\setline{1}%  
 2264 After the noble count Otto had taken in pledge the power over  
 2265 Nijmegen,\footnote{In 1247 William II\protect\index{William II of Holland}  
 2266 (1227--1256) count of Holland needed money to fight his way to  
 2267 Aachen\protect\index{Aachen} to be crowned King of the Holy Roman  
 2268 Empire. He gave the town of Nijmegen in pledge to Otto  
 2269 II\protect\index{Otto II of Guelders} (1229--1271) count of Guelders.}  
 2270 like I have written above, he wanted to protect the town. So in June  
 2271 1254\ledsidenote{1254} he and the judge, the sheriffs and other  
 2272 citizens of Nijmegen obtained permission to demolish the parish  
 2273 church that lay outside the town walls,\footnote{Since the early  
 2274 seventh century old St.\ Stephen's church had been located close  
 2275 to the castle, at today's  
 2276 Kelfkensbos\protect\index{Kelfkensbos (Nijmegen)}} square.  
 2277 Traces of the church and the presbytery were found during excavations  
 2278 in 1998--1999.} to move it inside the walls and to rebuild it new.  
 2279 This operation was necessary and useful both for Otto himself and  
 2280 for the inhabitants of the town. The reverend father Conrad of  
 2281 Hochstaden, archbishop of  
 2282 Cologne,\footnote{Conrad of Hochstaden ({\textdag} 1261) was  
 2283 archbishop of Cologne in 1238--1261. Nijmegen belonged to the  
 2284 archdiocese of Cologne until 1559.} gave his permission. So did the  
 2285 reverend dean and canons of the chapter of St.\  
 2286 Apostles'\protect\index{St. Apostles' (Cologne)} in Cologne, who had  
 2287 long\footnote{They probably became the patrons when the chapter was  
 2288 established in the early eleventh century. About the church and the  
 2289 chapter, see Gottfried Stracke\protect\index{Stracke, G.},  
 2290 \textit{K}\{"o}ln: St.\ Aposteln}, Stadspuren -- Denkm\{"a}ler in  
 2291 K\{"o}ln, vol.\ 19, K\{"o}ln: J.\,P.\ Bachem, 1992.} been the true  
 2292 and benevolent patrons of the church---but they did not allow Otto  
 2293 to do anything without their knowledge, nor to infringe their rights,  
 2294 nor to damage their property.  
 2295 \pend  
 2296  
 2297 \pstart  
 2298 And so the count and the town voluntarily gave an open space in town  
 2299 called Hundisburg, which was owned by the aforementioned king William,  
 2300 to the dean and chapter of St.\ Apostles' in order to build and  
 2301 consecrate a church and graveyard. King William approved and the  
 2302 town of Nijmegen explicitly expressed its assent. A new ditch was dug  
 2303 on property of the church near the castle and the  
 2304 harbour,\footnote{Nowadays, the exact location of the medieval  
 2305 ditch---and of two Roman ones---can be seen in the pavement of  
 2306 Kelfkensbos\protect\index{Kelfkensbos (Nijmegen)}} square.} causing  
 2307 the demolition of the presbytery. In compensation, the count and  
 2308 citizens committed themselves to giving the parish priest another  
 2309 suitable space close enough to the new church that was about to be  
 2310 built. A letter about these transactions, with the seals of count  
 2311 Otto and the town of Nijmegen, is kept at St.\ Apostles'  
 2312 church.\footnote{The original letter is lost. A 15th century

```

2313 transcription of it is kept at the Historisches Archiv der
2314 Stadt K\{"o}ln (HAsTK).}
2315 \pend
2316
2317 \pstart
2318 // One additional line to show synchronization. //
2319 \pend
2320 \endnumbering
2321 \end{Rightside}
2322 \Pages
2323 \end{pages}
2324
2325 %%%%%%%%%%%
2326 \printindex
2327 \end{document}
2328 %%%%%%%%%%%
2329
2330 </djd17nov>

```

---

### A.3 Example poetry on parallel facing pages

This example, illustrated in Figures 4 to 7, was originally provided in November 2004 by Dirk-Jan Dekker for an earlier version of `ledpar`. I have updated it, and also extended it to show the difference between the `\stanza` command and the `astanza` environment. `\stanza` is used for the first pair of pages and `astanza` for the second pair. Note the definition of `\endstanzaextra` to give a short line after each stanza.

---

```

2331 (*djdpoems)
2332 %% djdpoems.tex example parallel verses on facing pages
2333 \documentclass{article}
2334 \usepackage{ledmac,ledpar}
2335 \addtolength{\textheight}{-15\baselineskip}
2336
2337 \maxchunks{24} % default value = 10
2338 \setstanzaindents{6,0,1,0,1}
2339
2340 \newcommand{\longdash}{-----}
2341
2342 \footparagraph{A} % for left pages
2343 \footparagraph{B} % for right pages
2344 \firstlinenum{1}
2345 \linenumincrement{1}
2346
2347 \let\oldBfootfmt\Bfootfmt
2348 \renewcommand{\Bfootfmt}[3]{%
2349 \let\printlines\printlinesR
2350 \oldBfootfmt{#1}{#2}{#3}}

```

```

2351
2352 \begin{document}
2353
2354 \newcommand{\interstanza}{\pstart\centering\longdash\skipnumbering\pend}
2355
2356 \begin{pages}
2357 \begin{Leftside}
2358 \def\endstanzaextra{\interstanza}
2359 \beginnumbering
2360
2361 \stanza
2362 Arma gravi numero violentaque bella parabam &
2363 edere, materi\={a} conveniente modis. &
2364 Par erat inferior versus---risisse Cupido &
2365 dicitur atque unum surripuisse pedem. \&
2366
2367 \stanza
2368 ‘Quis tibi, saeve puer, dedit hoc in carmina iuris? &
2369 Pieridum vates, non tua turba \edtext{sumus}{\Afootnote{note lost}}. &
2370 Quid si praeripiat flavae V\ue}nus arma Minervae, &
2371 ventilet accensas flava Minerva faces? \&
2372
2373 \stanza
2374 Quis probet in silvis Cererem regnare iugosis, &
2375 lege pharetratae Virginis arva coli? &
2376 Crinibus insignem quis \edtext{acuta}{\Afootnote{acut\={a} (abl.\ abs.)}} &
2377 cuspidе Phoebum &
2378 instruat, Aoniam Marte movente lyram? \&
2379 \endnumbering
2380 \end{Leftside}
2381
2382 \begin{Rightside}
2383 \def\endstanzaextra{\interstanza}
2384 \beginnumbering
2385 \firstlinenum{1}
2386 \linenumincrement{1}
2387 \setstanzaindents{6,0,1,0,1,0}
2388
2389 \stanza
2390 I was preparing to sing of weapons and violent wars, &
2391 in heavy numbers, with the subject matter suited to the verse measure. &
2392 The even lines were as long as the odd ones, but Cupid laughed, &
2393 they said, and he stole away one foot.\footnote{I.e., the even lines,
2394 which were hexameters (with six feet) became pentameters
2395 (with five feet).} \&
2396
2397 \stanza
2398 ‘O cruel boy, who gave you the right over poetry? &
2399 We poets belong to the Pierides,\footnote{Muses} we are not your folk. &
2400 \edlabel{beginparadox}What if Venus should seize away the arms of

```

```

2401 Minerva with the golden hair, &
2402 if Minerva with the golden hair should fan alight the kindled torch
2403 of love? \&
2404
2405 \stanza
2406 Who would approve of Ceres\footnote{Ceres was the Roman goddess of
2407 the harvest.} reigning on the woodland ridges, &
2408 and of land tilled under the law of the Maid with the
2409 quiver\footnote{By '\textit{Virgo}' ('Virgin') Ovid means Diana, the
2410 Roman goddess of the hunt.}? &
2411 Who would provide Phoebus with his beautiful hair with a sharp-pointed
2412 spear, &
2413 while Mars stirs the \edtext{Aonian}{\Bfootnote{Mount Parnassus,
2414 where the Muses live, is located in Aonia.}}
2415 lyre?\edlabel{endparadox}\footnote{Lines
2416 \xlineref{beginparadox}--\xlineref{endparadox} show some paradoxical
2417 situations that would occur if the gods didn't stay with their own
2418 business.} \&
2419 \endnumbering
2420 \end{Rightside}
2421
2422 \Pages
2423 \end{pages}
2424
2425 \begin{pages}
2426 \begin{Leftside}
2427 \def\endstanzaextra{\interstanza}
2428 \beginnumbering
2429
2430 \begin{astanza}
2431 Arma gravi numero violentaque bella parabam &
2432 edere, materi\={a} conveniente modis. &
2433 Par erat inferior versus---risisse Cupido &
2434 dicitur atque unum surripuisse pedem. \&
2435 \end{astanza}
2436
2437 \begin{astanza}
2438 'Quis tibi, saeve puer, dedit hoc in carmina iuris? &
2439 Pieridum vates, non tua turba \edtext{sumus}{\Afootnote{note lost}}. &
2440 Quid si praeripiat flavae V\ufe}nus arma Minervae, &
2441 ventilet accensas flava Minerva faces? \&
2442 \end{astanza}
2443
2444 \begin{astanza}
2445 Quis probet in silvis Cererem regnare iugosis, &
2446 lege pharetratae Virginis arva coli? &
2447 Crinibus insignem quis \edtext{acuta}{\Afootnote{acut\={a} (abl.\ abs.)}}
2448 cuspide Phoebum &
2449 instruat, Aoniam Marte movente lyram? \&
2450 \end{astanza}

```

```

2451
2452 \endnumbering
2453 \end{Leftside}
2454
2455 \begin{Rightside}
2456 \def\endstanzaextra{\interstanza}
2457 \beginnumbering
2458 \firstlinenum{1}
2459 \linenumincrement{1}
2460 \setstanzaindents{6,0,1,0,1,0}
2461
2462 \begin{astanza}
2463 I was preparing to sing of weapons and violent wars, &
2464 in heavy numbers, with the subject matter suited to the verse measure. &
2465 The even lines were as long as the odd ones, but Cupid laughed, &
2466 they said, and he stole away one foot.\footnote{I.e., the even lines,
2467 which were hexameters (with six feet) became pentameters
2468 (with five feet).} \&
2469 \end{astanza}
2470
2471 \begin{astanza}
2472 ‘‘O cruel boy, who gave you the right over poetry? &
2473 We poets belong to the Pierides,\footnote{Muses} we are not your folk. &
2474 \edlabel{beginparadox}What if Venus should seize away the arms of
2475 Minerva with the golden hair, &
2476 if Minerva with the golden hair should fan alight the kindled torch
2477 of love? \&
2478 \end{astanza}
2479
2480 \begin{astanza}
2481 Who would approve of Ceres\footnote{Ceres was the Roman goddess of the
2482 harvest.} reigning on the woodland ridges, &
2483 and of land tilled under the law of the Maid with the
2484 quiver\footnote{By ‘\textit{Virgo}’ (‘Virgin’) Ovid means Diana,
2485 the Roman goddess of the hunt.}? &
2486 Who would provide Phoebus with his beautiful hair with a sharp-pointed
2487 spear, &
2488 while Mars stirs the \edtext{Aonian}{\Bfootnote{Mount Parnassus, where
2489 the Muses live, is located in Aonia.}}
2490 lyre?\edlabel{endparadox}\footnote{Lines
2491 \xlineref{beginparadox}--\xlineref{endparadox} show some paradoxical
2492 situations that would occur if the gods didn’t stay with their
2493 own business.} \&
2494 \end{astanza}
2495
2496 \endnumbering
2497 \end{Rightside}
2498
2499 \Pages
2500 \end{pages}

```

```
2501
2502 \end{document}
2503
2504 </djdpoems>
```

---



## References

- [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`)
- [Wil02] Peter Wilson. *The memoir class for configurable typesetting*. November 2002. (Available from CTAN in `macros/latex/contrib/memoir`)
- [Wil04] Peter Wilson. *ledmac A presumptuous attempt to port EDMAC, TABMAC and EDSTANZA to LaTeX*. December 2004. (Available from CTAN in `macros/latex/contrib/ledmac`)

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

| Symbols                                                                                                                                                         |                                                                                                                                                                                                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\&amp;</code> . . . 1423, 1424, 1428, 1445, 1459,<br>1968, 1983, 2020, 2043, 2365,<br>2371, 2378, 2395, 2403, 2418,<br>2434, 2441, 2449, 2468, 2477, 2493 | <code>\@donetotallinesR</code> . . . . . 832,<br>893, 896, 1706, 1910, 1920, 1922                                                                                                                                       |
| <code>\@M</code> . . . . . 1434                                                                                                                                 | <code>\@insertR</code> . . . . . 1106–1108, 1121–1123                                                                                                                                                                   |
| <code>\@MM</code> . . . . . 2094                                                                                                                                | <code>\@l</code> . . . . . 251, 564                                                                                                                                                                                     |
| <code>\@adv</code> . . . . . 328, 595, 596                                                                                                                      | <code>\@l@dttempcnta</code> 401, 403, 405, 406, 410,<br>412, 414, 415, 926, 965, 966,<br>968, 970, 973, 974, 991–995,<br>997, 1004, 1009, 1013, 1021,<br>1026, 1030, 1062, 1065, 1067, 1071                             |
| <code>\@afterindentfalse</code> . . . . . 706                                                                                                                   | <code>\@l@dttempcntb</code> . . . . . 144, 146,<br>148, 956, 957, 1004, 1009, 1013,<br>1021, 1026, 1030, 1054, 1058,<br>1071, 1079–1081, 1083, 1328,<br>1330, 1332, 1385–1387, 1389,<br>1487–1491, 1495–1498, 1502–1505 |
| <code>\@arabic</code> . . . . . 185, 186                                                                                                                        | <code>\@l@reg</code> . . . . . 300                                                                                                                                                                                      |
| <code>\@astanza@line</code> . . . . . 1444, 1450, 1453                                                                                                          | <code>\@l@regR</code> . . . . . 251                                                                                                                                                                                     |
| <code>\@auxout</code> . . . . . 1290, 1300, 1533                                                                                                                | <code>\@lab</code> . . . . . 520, 1284, 1293, 1314                                                                                                                                                                      |
| <code>\@chapter</code> . . . . . 707                                                                                                                            | <code>\@lockR</code> . . . . . 60, 273, 275, 277,<br>290, 435, 451, 452, 454, 455,<br>483, 484, 486, 909, 932, 934,<br>935, 937, 1018, 1035, 1037, 1039                                                                 |
| <code>\@cs@linesinparL</code> . . . . . 1689, 1796                                                                                                              | <code>\@lopL</code> . . . . . 542, 1823                                                                                                                                                                                 |
| <code>\@cs@linesinparR</code> . . . . . 1689, 1796                                                                                                              | <code>\@lopR</code> . . . . . 542, 1826                                                                                                                                                                                 |
| <code>\@cs@linesonpageL</code> . . 1697, 1755, 1809                                                                                                             |                                                                                                                                                                                                                         |
| <code>\@cs@linesonpageR</code> . . 1697, 1755, 1809                                                                                                             |                                                                                                                                                                                                                         |
| <code>\@donereallinesL</code> . . . . .<br>. . . . . 832, 859, 1653, 1655, 1703                                                                                 |                                                                                                                                                                                                                         |
| <code>\@donereallinesR</code> . . . . .<br>. . . . . 832, 892, 1658, 1660, 1705                                                                                 |                                                                                                                                                                                                                         |
| <code>\@donetotallinesL</code> . . . . . 832,<br>860, 863, 1704, 1890, 1900, 1902                                                                               |                                                                                                                                                                                                                         |

- `\@nameuse` ..... 1413, 1417, 2098  
`\@nobreakfalse` ..... 744, 773  
`\@nobreaktrue` ..... 742, 746, 771, 775  
`\@oldnobreak` 742, 744, 771, 773, 809, 822  
`\@pend` ..... 535, 1653  
`\@pendR` ..... 535, 1658  
`\@pstartsfalse` ..... 1628  
`\@pstartstrue` ..... 1628  
`\@ref` ..... 507, 568, 572  
`\@ref@reg` ..... 533  
`\@schapter` ..... 707  
`\@set` ..... 360, 602, 603  
`\@tag` 633, 650, 1130, 1134, 1144, 1148,  
1158, 1162, 1172, 1176, 1186,  
1190, 1201, 1205, 1215, 1219,  
1229, 1233, 1243, 1247, 1257, 1261  
`\@temp` ..... 1524  
`\@templ@d` ..... 1376, 1377  
`\@writelinesinparL` .. 1610, 1651, 1894  
`\@writelinesinparR` .. 1611, 1651, 1914  
`\@writelinesonpageL` . 1725, 1727, 1822  
`\@writelinesonpageR` . 1745, 1747, 1822  
`\@xloop` ..... 1119  
`\add@penaltiesR` ..... 891, 1115  
`\addtocontents` ..... 1534–1536  
`\addtocounter` ..... 2260  
`\addtolength` ... 1931, 2083–2085, 2335  
`\advanceline` ..... 594, 625  
`\affixline@num` ..... 850  
`\affixline@numR` ..... 883, 988  
`\affixside@note` ..... 853  
`\affixside@noteR` ..... 886, 1375  
`\Afootfmt` ..... 2164  
`\Afootins` ..... 2083  
`\Afootnote` ..... 1126, 2013,  
2018, 2189, 2191, 2195, 2226,  
2228, 2245, 2369, 2376, 2439, 2447  
`\Afootnoterule` ..... 2079  
`\Aparafootfmt` ..... 2129, 2164  
`\apptocmd` ..... 1564  
`\araw@textfalse` ..... 1639  
`\araw@texttrue` ..... 1639  
`astanza` (environment) ..... 7, 1426  
`\AtBeginDocument` ... 1310, 1509, 1544  
`\author` ..... 2172
- ## B
- `\ballast@count` ..... 916, 921  
`\bbl@main@language` ..... 1560, 1561  
`\bbl@set@language` ..... 1551, 1552  
`\beginnumbering` ..... 6, 36, 723,  
749, 1962, 1977, 2007, 2028,  
2179, 2258, 2359, 2384, 2428, 2457  
`\beginnumberingR` ... 49, 102, 723, 778  
`\Bfootfmt` 1936, 1937, 2165, 2347, 2348  
`\Bfootins` ..... 2084  
`\Bfootnote` ..... 1140, 2031, 2035,  
2187, 2192–2194, 2196, 2200,  
2201, 2210, 2212, 2214, 2219,  
2221, 2222, 2225, 2227, 2229,  
2231, 2232, 2235, 2238, 2239,  
2242–2244, 2246, 2247, 2413, 2488  
`\Bfootnoterule` ..... 2080  
`\box` ..... 2112  
`\Bparafootfmt` ..... 2137, 2165  
`\bypage@Rfalse` ..... 122, 134  
`\bypage@Rtrue` ..... 122, 130
- ## C
- `\c@ballast` ..... 921  
`\c@firstlinenumR` ..... 153, 1060  
`\c@firstsublinenumR` ..... 157, 1055  
`\c@linenumincrementR` ..... 153, 1060
- ## A
- `\abb` ..... 2122,  
2126, 2193, 2227, 2232, 2238, 2242  
`\absline@num` ..... 394, 408, 427  
`\absline@numR` ... 58, 202, 253, 256,  
259, 391, 399, 420, 439, 473,  
501, 512, 901, 918, 919, 956, 1105  
`\actionlines@list` .....  
..... 243, 246, 394, 408, 427  
`\actionlines@listR` .....  
..... 206, 221, 235, 238, 391,  
399, 420, 439, 473, 501, 978, 981  
`\actions@list` . 247, 395, 415, 429, 431  
`\actions@listR` .....  
. 206, 222, 239, 392, 406, 422,  
424, 441, 450, 475, 482, 502, 982  
`\add@inserts` ..... 852  
`\add@inserts@nextR` ..... 1094  
`\add@insertsR` ..... 885, 1094  
`\add@penaltiesL` ..... 858, 1115

- `\c@page` ... 564, 1774, 1780, 1783, 1790  
`\c@sublinenumincrementR` .. 157, 1055  
`\centering` ..... 2354  
`\Cfootfmt` ..... 2166  
`\Cfootins` ..... 2085  
`\Cfootnote` ..... 1140, 2202, 2240  
`\Cfootnoterule` ..... 2081  
`\ch@ck@l@ckR` ..... 988  
`\ch@cksub@l@ckR` ..... 988  
`\ch@cksub@lockR` ..... 1056  
`\chapter` ..... 693, 694, 702  
`\chapterinpages` ..... 686, 694, 704  
`\chardef` ..... 1423  
`\check@goal` ..... 1848, 1862, 1874  
`\check@pstarts` .....  
1587, 1612, 1628, 1683, 1691, 1699  
`\checkpageL` ..... 1710, 1722, 1842  
`\checkpageR` ..... 1731, 1742, 1842  
`\checkraw@text` .....  
.... 1591, 1608, 1639, 1707, 1751  
`\cleardoublepage` ..... 1786  
`\clearl@dleftpage` ..... 1730, 1772  
`\clearl@drighpage` ..... 1750, 1772  
`\cleartoevenpage` ..... 1772  
`\cleartol@devenpage` ..... 1672, 1772  
`\closeout` ..... 555, 559  
`\columnrulewidth` ..... 4, 1623, 1953  
`\Columns` ..... 3, 1575, 1987, 2047  
`\columnseparator` ..... 4, 1605, 1623  
`\countLline` ..... 827, 838  
`\countRline` ..... 827, 871  
`\Cparafootfmt` ..... 2150, 2166  
`\critext` ..... 630
- D**
- `\date` ..... 2174  
`\DeclareOption` ..... 7  
Dekker, Dirk-Jan ..... 78, 84  
`\Dfootnote` ..... 1140  
`\dimen` ..... 581, 582, 586–588, 592  
`\divide` ..... 993  
`\do@actions@fixedcodeR` ..... 925  
`\do@actions@nextR` ..... 925  
`\do@actionsR` ..... 902, 925  
`\do@ballastR` ..... 903, 916  
`\do@lineL` ..... 837, 1595, 1599, 1714  
`\do@lineLhook` ..... 842, 867  
`\do@lineR` ..... 870, 1596, 1601, 1734  
`\do@lineRhook` ..... 867, 875  
`\do@lockoff` ..... 470  
`\do@lockoffL` ..... 494  
`\do@lockoffR` ..... 470  
`\do@lockon` ..... 435  
`\do@lockonL` ..... 467  
`\do@lockonR` ..... 435  
`\documentclass` ..... 1930, 2066, 2333  
`\dp` ..... 2095, 2110  
`\dummy@ref` ..... 516
- E**
- `\edfont@info` ..... 669, 672, 678, 681  
`\edindex` .. 2184, 2199, 2211, 2223, 2224  
`\edlabel` .. 1282, 2400, 2415, 2474, 2490  
`\edtext` ..... 647, 2012,  
2018, 2030, 2034, 2187, 2189,  
2191–2195, 2198, 2200, 2201,  
2210, 2212, 2214, 2219, 2221,  
2222, 2225–2229, 2231, 2232,  
2235, 2237, 2239, 2242–2247,  
2369, 2376, 2413, 2439, 2447, 2488  
`\Efootnote` ..... 1140  
`\emergencystretch` ..... 2170  
`\empty` .. 76, 79, 235, 243, 641, 658,  
667, 676, 757, 786, 978, 1059,  
1067, 1096–1098, 1109, 1120,  
1285, 1294, 1797, 1803, 1810, 1816  
`\end@lemmas` ..... 641, 642, 658, 659  
`\endashchar` ..... 1272  
`\endgraf` ..... 805, 818, 1581, 1675  
`\endline@num` ..... 523, 529  
`\endlock` ..... 614, 1432, 1441, 1446  
`\endnumbering` ..... 6, 39, 69, 106,  
724, 1969, 1984, 2021, 2044,  
2253, 2320, 2379, 2419, 2452, 2496  
`\endnumberingR` 52, 69, 91, 101, 114, 724  
`\endpage@num` ..... 522, 529  
`\endstanzaextra` .....  
.... 1448, 2358, 2383, 2427, 2456  
`\endsub` ..... 581  
`\endsubline@num` ..... 524, 530  
`\enskip` ..... 2133, 2146, 2154  
`\enspace` ..... 2132, 2142, 2153  
environments:  
`astanza` ..... 7, 1426  
`Leftside` ..... 5, 709  
`pages` ..... 4, 686  
`pairs` ..... 3, 686  
`Rightside` ..... 5, 721  
`\extensionchars` .. 47, 66, 97, 111, 119

**F**

`\f@x@l@cksR` ..... 988  
`\first@linenum@out@Rfalse` .. 550, 556  
`\first@linenum@out@Rtrue` ..... 550  
`\firstlinenum` ..... 5, 162, 1959,  
 1974, 2004, 2025, 2344, 2385, 2458  
`\firstsublinenum` ..... 5, 162  
`\fix@page` ..... 296, 303  
`\flag@end` ..... 565, 646, 663  
`\flag@start` ..... 565, 638, 655  
`\floatingpenalty` ..... 2094  
`\flush@notes` ..... 1614, 1760  
`\flush@notesR` ..... 1118, 1615, 1761  
`\footnote` .....  
 . 2265, 2273, 2282, 2287, 2304,  
 2312, 2393, 2399, 2406, 2409,  
 2415, 2466, 2473, 2481, 2484, 2490  
`\footnotesize` ..... 2076, 2077  
`\footparagraph` . 2160–2162, 2342, 2343  
`\fullstop` . 198, 1269, 1271, 1273, 1275

**G**

`\get@linelistfile` ..... 231  
`\get@nextboxL` ..... 1721, 1887  
`\get@nextboxR` ..... 1741, 1887  
`\getline@num` ..... 847  
`\getline@numR` ..... 880, 900  
`\getlinesfrompagelistL` .....  
 ..... 1695, 1753, 1809  
`\getlinesfrompagelistR` .....  
 ..... 1696, 1754, 1809  
`\getlinesfromparlistL` ... 1687, 1796  
`\getlinesfromparlistR` ... 1688, 1796  
`\gl@p` 238, 239, 246, 247, 642, 659, 671,  
 680, 981, 982, 1102, 1106, 1121,  
 1288, 1297, 1800, 1806, 1813, 1819  
`\goalfraction` ..... 4, 1874

**H**

`\hb@xt@` ... 849, 855, 862, 882, 888,  
 895, 1603, 1717, 1719, 1737, 1739  
`\hsize` ..... 767, 796,  
 1603, 1717, 1719, 1737, 1739, 2107  
`\hyphenation` ..... 2175

**I**

`\if@files` ..... 1532  
`\if@firstcolumn` ..... 1073, 1379  
`\if@nobreak` ..... 741, 770  
`\if@pstarts` .... 1588, 1628, 1684, 1700

`\ifaraw@text` ... 1593, 1639, 1709, 1752  
`\ifbypage@` ..... 319  
`\ifbypage@R` ..... 122, 309, 960  
`\ifdim` ..... 582, 586, 588,  
 592, 1717, 1737, 1776, 1849, 1863  
`\iffirst@linenum@out@R` .... 550, 554  
`\ifl@d@dash` ..... 1272  
`\ifl@d@elin` ..... 1274, 1275  
`\ifl@d@esl` ..... 1275  
`\ifl@d@pnum` ..... 1269, 1273  
`\ifl@d@ssub` ..... 1271  
`\ifl@d@pagefull` ..... 1724, 1744, 1842  
`\ifl@d@pacing` ..... 9  
`\ifl@d@pairing` ..... 9, 73  
`\ifl@d@samelang` ..... 1520, 1594  
`\ifl@d@samepage` ..... 1713, 1733, 1842  
`\ifl@d@skipnumber` ..... 1050  
`\ifl@dusedbabel` ..... 1518  
`\ifled@plinum` ..... 1270, 2141  
`\ifled@Rcol` ..... 9, 145, 167,  
 171, 175, 179, 218, 233, 297,  
 306, 330, 344, 361, 378, 390,  
 398, 419, 464, 491, 500, 509,  
 566, 576, 583, 589, 595, 602,  
 610, 615, 619, 624, 635, 652,  
 666, 1128, 1142, 1156, 1170,  
 1184, 1199, 1213, 1227, 1241,  
 1255, 1283, 1315, 1329, 1340,  
 1352, 1364, 1399, 1412, 1555, 1565  
`\ifnoteschanged@` ..... 83  
`\ifnumberedpar@` ... 751, 780, 801,  
 814, 1127, 1141, 1155, 1169,  
 1183, 1198, 1212, 1226, 1240,  
 1254, 1339, 1351, 1363, 1398, 1411  
`\ifnumbering` ..... 37, 747, 798  
`\ifnumberingR` .....  
 .... 31, 50, 70, 93, 125, 776, 811  
`\ifodd` 1083, 1389, 1774, 1780, 1783, 1790  
`\ifpst@rtedL` ..... 32, 755  
`\ifpst@rtedR` ..... 32, 784  
`\ifshiftedverses` . 5, 1716, 1736, 1875  
`\ifsublines@` 196, 285, 329, 362, 369,  
 400, 409, 421, 428, 440, 474,  
 528, 530, 904, 967, 1053, 1317, 1321  
`\ifvbox` 839, 872, 1643, 1646, 1888, 1908  
`\ifwrittenlinesL` ..... 1884, 1892  
`\ifwrittenlinesR` ..... 1885, 1912  
`\initnumbering@reg` ..... 45  
`\insert` ..... 2090

- \insert@count . . . . . 506, 572, 636,  
653, 1135, 1149, 1163, 1177,  
1191, 1206, 1220, 1234, 1248,  
1262, 1347, 1359, 1371, 1406, 1419
- \insert@countR . . . . . 507, 568, 635,  
652, 1131, 1145, 1159, 1173,  
1187, 1202, 1216, 1230, 1244,  
1258, 1343, 1355, 1367, 1402, 1415
- \insertlines@listR . . . . .  
. . . . . 76, 206, 220, 512, 1098, 1102
- \inserts@list . . . . .  
. . . . . 756, 1134, 1148, 1162, 1176,  
1190, 1205, 1219, 1233, 1247,  
1261, 1346, 1358, 1370, 1405, 1418
- \inserts@listR . . . . .  
. . . . . 785, 1093, 1096, 1106, 1120,  
1121, 1130, 1144, 1158, 1172,  
1186, 1201, 1215, 1229, 1243,  
1257, 1342, 1354, 1366, 1401, 1414
- \interfootnotelinepenalty . . . . . 2093
- \interlinepenalty . . . . . 1434, 2093
- \interstanza . . . . .  
. . . . . 2354, 2358, 2383, 2427, 2456
- L**
- \l@d@nums . . . . . 669, 672, 678,  
681, 1130, 1134, 1144, 1148,  
1158, 1162, 1172, 1176, 1186,  
1190, 1201, 1205, 1215, 1219,  
1229, 1233, 1243, 1247, 1257, 1261
- \l@d@set . . . . . 377, 610, 611
- \l@dbbl@set@language . . . . . 1529, 1552
- \l@dbfnote . . . . . 1397
- \l@dc@maxchunks . . . . . 762, 764,  
791, 793, 1477, 1487, 1495, 1502
- \l@dcalc@maxoftwo . . . . .  
. . . . . 1689, 1829, 1899, 1919
- \l@dcalc@minoftwo . . . . . 1697, 1755, 1829
- \l@dcalcnun . . . . . 988
- \l@dchecklang . . . . . 1522, 1592
- \l@dchset@num . . . . . 252, 255, 377
- \l@dcsnote . . . . . 1338
- \l@dcsnotetext . . . . .  
. . . . . 1377, 1380, 1382, 1390, 1392
- \l@demptyd@ta . . . . . 843, 876
- \l@dend@stuff . . . . . 48, 67, 98, 112, 120
- \l@dgetline@margin . . . . . 143
- \l@dgetsidenote@margin . . . . . 1327
- \l@dld@ta . . . . . 851, 884, 1074, 1086
- \l@dleftbox . . . . .  
. . . . . 824, 848, 862, 1604, 1717, 1719
- \l@dlinenumR . . . . . 188
- \l@dlsn@te . . . . . 854, 887
- \l@dlsnote . . . . . 1338
- \l@dmake@labels . . . . . 1301
- \l@dmake@labelsR . . . . . 1291, 1304
- \l@dminpagelines . . . . .  
. . . . . 1662, 1698, 1756, 1850, 1864
- \l@dnumpstartsL . . . . . 41, 761, 762, 764,  
766, 1481, 1513, 1576, 1577,  
1619, 1631, 1669, 1670, 1765, 1897
- \l@dnumpstartsR . . . . . 54, 790, 791, 793,  
795, 1481, 1514, 1576, 1577,  
1620, 1634, 1669, 1670, 1766, 1917
- \l@doldbbl@set@language . . . . . 1551
- \l@doldselectlanguage . . . . . 1550, 1554, 1559
- \l@dpagefullfalse . . . . . 1842
- \l@dpagefulltrue . . . . . 1842
- \l@dpagingfalse . . . . . 11, 688, 701
- \l@dpagingtrue . . . . . 696
- \l@dpairingfalse . . . . . 9, 690, 700
- \l@dpairingtrue . . . . . 687, 695
- \l@dparsedendline . . . . . 2103
- \l@dparsedstartline . . . . . 2098, 2103, 2105
- \l@dparsedstartpage . . . . . 2101, 2106
- \l@dparsefootspec . . . . . 2097
- \l@dpscl . . . . . 839, 844, 1483, 1515, 1585,  
1589, 1617, 1631, 1643, 1679,  
1685, 1690, 1693, 1701, 1763,  
1888, 1890, 1897, 1899, 1901, 1903
- \l@dpscr . . . . . 872, 877, 1484, 1516,  
1586, 1590, 1618, 1634, 1646,  
1680, 1686, 1694, 1702, 1764,  
1908, 1910, 1917, 1919, 1921, 1923
- \l@drd@ta . . . . . 855, 888, 1076, 1084
- \l@drightbox . . . . .  
. . . . . 824, 881, 895, 1606, 1737, 1739
- \l@drsn@te . . . . . 856, 889
- \l@drsnote . . . . . 1338
- \l@dsamelangfalse . . . . . 1520, 1523
- \l@dsamelangtrue . . . . . 1520, 1526
- \l@dsamepagefalse . . . . . 1842
- \l@dsamepagetrue . . . . . 1842
- \l@dsetupmaxlinecounts . . . . . 1494, 1511
- \l@dsetupprawboxes . . . . . 1486, 1510
- \l@dskipnumberfalse . . . . . 1051
- \l@dskipnumbertrue . . . . . 948
- \l@dunhbox@line . . . . . 855, 888
- \l@dusedbabelfalse . . . . . 1518, 1547

- \l@dusedbabeltrue . . . . . 1518, 1549
- \l@duselanguage . . . . .
- . . . . . 1539, 1598, 1600, 1711, 1732
- \l@dzeromaxlinecounts . . . 1494, 1512
- \l@dzeropenalties 804, 817, 1580, 1674
- \l@pscL . . . . . 1483
- \l@pscR . . . . . 1483
- \label@refs . . . . .
- 1286, 1288, 1291, 1295, 1297, 1301
- \labelref@list . . . . . 1294, 1297, 1322
- \labelref@listR 1280, 1285, 1288, 1318
- \languagename . . 1530, 1531, 1533–1536
- \last@page@num . . . . . 317, 323
- \last@page@numR . . . . . 303
- \lastbox . . . . . 846, 879
- \lastskip . . . . . 581, 587
- \Lcolwidth . . . . . 3, 4, 14, 697, 767,
- 849, 862, 1954, 1955, 1998, 1999
- \ldots . . . . . 2017,
- 2020, 2040, 2043, 2207, 2209, 2238
- \led@err@BadLeftRightPstarts . . .
- . . . . . 22, 1577, 1670
- \led@err@LeftOnRightPage . . . 25, 1784
- \led@err@LineationInNumbered . . . 126
- \led@err@NumberingNotStarted . . . 87
- \led@err@NumberingShouldHaveStarted
- . . . . . 100
- \led@err@NumberingStarted . . . . 38, 51
- \led@err@PendNoPstart . . . . . 802, 815
- \led@err@PendNotNumbered . . . 799, 812
- \led@err@PstartInPstart . . . 752, 781
- \led@err@PstartNotNumbered . 748, 777
- \led@err@RightOnLeftPage . . . 25, 1791
- \led@err@TooManyPstarts 19, 763, 792
- \led@mess@NotesChanged . . . . . 84
- \led@mess@SectionContinued . . . .
- . . . . . 96, 110, 118
- \led@warn@BadAction . . . . . 950
- \led@warn@BadAdvancelineLine 347, 353
- \led@warn@BadAdvancelineSubline .
- . . . . . 333, 339
- \led@warn@BadLineation . . . . . 136
- \led@warn@BadSetline . . . . . 600
- \led@warn@BadSetlinenum . . . . . 608
- \led@warn@DuplicateLabel . . . . . 1306
- \ledllfill . . . . . 855, 888
- \ledmac@error . . . . . 20, 23, 26, 28
- \ledplinenumfalse . . . . . 2099
- \ledplinenumtrue . . . 2097, 2102, 2104
- \ledRcolfalse . . . . . 13, 710, 733
- \ledRcoltrue . . . . . 722
- \ledrlfill . . . . . 855, 888
- \ledsavedprintlines . . . . . 7, 1267
- \ledsetnormalparstuff 2130, 2138, 2151
- \ledsidenote . . . . . 2271
- \ledstrutL . . . . . 1717, 1719, 1769
- \ledstrutR . . . . . 1737, 1739, 1769
- \ledthegoal . . . . . 1849, 1863, 1874
- \leftlinenumR . . . . . 188, 1074, 1086
- Leftside (environment) . . . . . 5, 709
- \Leftsidehook . . . . . 714, 716
- \Leftsidehookend . . . . . 715, 716
- \lemma . . . . . 2012, 2238
- \lemmafонт . . . . . 2133, 2154
- \line@list . . . . . 676, 680
- \line@list@stuff . . . . . 47, 111
- \line@list@stuffR . . . 66, 97, 119, 552
- \line@listR . 79, 206, 219, 530, 667, 671
- \line@margin . . . . . 148
- \line@marginR . . . . . 141, 1079
- \line@num . . . . . 320, 351,
- 352, 354, 372, 383, 384, 412, 1320
- \line@numR . 59, 195, 202, 257, 291,
- 310, 345, 346, 348, 365, 379,
- 380, 403, 523, 527, 910, 961,
- 970, 1058, 1060, 1062, 1063, 1316
- \lineation . . . . . 730, 2070
- \lineationR . . . . . 124, 730
- \linenum@out 571, 579, 584, 590, 596,
- 603, 611, 616, 620, 1293, 1653, 1823
- \linenum@outR . . . . .
- . 549, 555, 557, 559, 560, 564,
- 567, 577, 583, 589, 595, 602,
- 610, 615, 619, 624, 1284, 1658, 1826
- \linenumberlist . . . . . 1059, 1063
- \linenumincrement . . 5, 162, 1960,
- 1975, 2005, 2026, 2345, 2386, 2459
- \linenummargin . . . . .
- 141, 1961, 1976, 2006, 2027, 2071
- \linenumr@p . . . . 1270, 1274, 1316, 1320
- \linenumrepR . . . . . 185, 195
- \linenumsep . . . . . 190, 192
- \linesinpar@listL . . . . .
- . . . . . 211, 227, 537, 1797, 1800
- \linesinpar@listR . . . . .
- . . . . . 211, 223, 540, 1803, 1806
- \linesonpage@listL 228, 544, 1810, 1813
- \linesonpage@listR 224, 547, 1816, 1819
- \list@clear . . . . .
- . 219–224, 227, 228, 230, 756, 785

\list@clearing@reg ..... 226  
 \list@create .....  
     ... 206–209, 211–213, 1093, 1280  
 \lock@disp ..... 1020, 1024, 1029  
 \lock@off .... 461, 462, 470, 619, 620  
 \lock@on ..... 615, 616  
 \longdash ..... 2340, 2354

## M

\maxchunks ..... 3, 1477, 2337  
 \maxdimen ..... 2107  
 \maxlinesinpar@list ..... 211, 230  
 \memorydump ..... 6, 713, 727  
 \memorydumpL ..... 105, 713  
 \memorydumpR ..... 105, 727  
 \message ..... 46, 65  
 \morenoexpands ..... 2125  
 \mpAfootnote ..... 1197  
 \mpBfootnote ..... 1197  
 \mpCfootnote ..... 1197  
 \mpDfootnote ..... 1197  
 \mpEfootnote ..... 1197  
 \mpvAfootnote ..... 1200, 1204, 1209  
 \mpvBfootnote ..... 1214, 1218, 1223  
 \mpvCfootnote ..... 1228, 1232, 1237  
 \mpvDfootnote ..... 1242, 1246, 1251  
 \mpvEfootnote ..... 1256, 1260, 1265  
 \multiply ..... 994

## N

\n@num ..... 498, 624  
 \n@num@reg ..... 504  
 \namebox ..... 839, 844, 872,  
     877, 1461, 1643, 1646, 1888, 1908  
 \NeedsTeXFormat ..... 2  
 \new@line ..... 855  
 \new@lineR ..... 563, 888  
 \newbox ..... 738, 824, 825, 1462  
 \newcounter ..... 153, 155, 157, 159  
 \newif ..... 5, 10,  
     12, 31, 33, 122, 550, 1518, 1520,  
     1628, 1639, 1842, 1844, 1884, 1885  
 \newnamebox ..... 1461, 1489, 1490  
 \newnamecount ..... 1472, 1497  
 \newwrite ..... 549  
 \next@action ..... 247  
 \next@actionline ..... 244, 246  
 \next@actionlineR .....  
     ..... 236, 238, 919, 957, 979, 981

\next@actionR ..... 239,  
     920, 958, 959, 964, 965, 973, 982  
 \next@insert ..... 757  
 \next@insertR .....  
     786, 1097, 1100, 1102, 1105, 1109  
 \next@page@num ..... 324, 395  
 \next@page@numR 63, 260, 262, 314, 392  
 \no@expands ..... 632, 649  
 \nobrak ..... 2123, 2124  
 \noindent ..... 2108, 2183, 2263  
 \normal@pars ..... 72, 760, 789  
 \normalbfnoteX ..... 1410  
 \notefontsetup ..... 2092  
 \notenunfont ... 2076, 2132, 2140, 2153  
 \noteschanged@true .....  
     ..... 77, 80, 668, 677, 1099  
 \notetextfont .. 2077, 2134, 2147, 2155  
 \num@lines ..... 805, 1581, 1675  
 \num@linesR ..... 737, 818, 1582, 1676  
 \numberedpar@true ..... 768, 797  
 \numberingRfalse ..... 71  
 \numberingRtrue ..... 56, 91, 115  
 \numberingtrue ..... 43, 107  
 \numlabfont ..... 195  
 \numpagelinesL .....  
     ..... 1662, 1715, 1725, 1729, 1850  
 \numpagelinesR .....  
     ..... 1662, 1735, 1745, 1749, 1864

## O

\oldBfootfmt ... 1936, 1939, 2347, 2350  
 \oldchapter ..... 693, 702  
 \one@line ..... 844, 846, 855  
 \one@lineR ..... 737, 877, 879, 888  
 \openout ..... 557, 560

## P

\page@action ..... 261, 389, 517  
 \page@num ..... 242, 322, 1387  
 \page@numR .....  
     215, 234, 312, 522, 527, 959, 1081  
 \pagegoal ..... 1882  
 \Pages ..... 4, 1666, 2322, 2422, 2499  
 \pages (environment) ..... 4, 686  
 \pagetotal ..... 1776, 1849, 1863  
 \pairs (environment) ..... 3, 686  
 \par@line ..... 806, 1583, 1677  
 \par@lineR ..... 737, 819, 1584, 1678  
 \para@vfootnote ..... 2089  
 \pausenumbering ..... 725

- \pausenumberingR ..... 90, 725
  - \pend ... 5, 712, 729, 753, 1447, 2216,  
2248, 2252, 2295, 2315, 2319, 2354
  - \pendL ..... 712, 798
  - \pendR ..... 729, 782, 811
  - \prevgraf .....  
. 805, 818, 1581, 1582, 1675, 1676
  - \previous@A@number ..... 2117
  - \previous@B@number ..... 2118
  - \previous@C@number ..... 2119
  - \previous@page ..... 2101, 2106, 2120
  - \printindex ..... 2326
  - \printlines .....  
1278, 1938, 2132, 2140, 2153, 2349
  - \printlinesR ..... 7, 1267, 1938, 2349
  - \ProcessOptions ..... 8
  - \protected@write ... 1290, 1300, 1533
  - \ProvidesPackage ..... 3
  - \pst@rtedLfalse ..... 32, 42
  - \pst@rtedLtrue ..... 108, 758
  - \pst@rtedRfalse ..... 34, 55, 74
  - \pst@rtedRtrue ..... 94, 116, 787
  - \pstart 5, 20, 24, 711, 728, 1449, 2179,  
2218, 2250, 2259, 2297, 2317, 2354
  - \pstartL ..... 711, 740
  - \pstartR ..... 728, 740
- R**
- \rbracket ..... 2123, 2146
  - \Rcolwidth ..... 3, 4,  
14, 698, 796, 882, 895, 1955, 1999
  - \read@linelist ..... 217, 553
  - \rem@inder ..... 1063, 1065–1067
  - \resumenumbering ..... 726
  - \resumenumberingR ..... 90, 726
  - \rightlinenumR ..... 188, 1076, 1084
  - Rightside (environment) ..... 5, 721
  - \Rightsidehook ..... 716, 731
  - \Rightsidehookend ..... 716, 734
  - \rlap ..... 1076, 1084
  - \Rlineflag ..... 7, 183, 195,  
1270, 1274, 1308, 1934, 1951, 2168
  - \rule ..... 1624
- S**
- \sc@n@list ..... 1064, 1066
  - \secdef ..... 707
  - \section@num ..... 44, 46, 47, 109–111
  - \section@numR .....  
... 29, 57, 65, 66, 95–97, 117–119
  - \select@language ... 1531, 1533–1536
  - \select@lemmafnt ..... 2146
  - \selectlanguage .... 1539, 2180, 2257
  - \set@line ..... 634, 651, 665
  - \set@line@action .....  
..... 254, 358, 367, 374, 397, 519
  - \setl@dlp@rbox ..... 1380, 1392
  - \setl@drp@rbox ..... 1382, 1390
  - \setline ..... 598, 2183, 2263
  - \setlinenum ..... 606
  - \setnamebox ..... 766, 795, 1461
  - \setprintlines ..... 1268
  - \setstanzaindents .....  
.... 1949, 1996, 2338, 2387, 2460
  - \shiftedversesfalse ..... 6
  - \shiftedversestrue ..... 7
  - \showlemma ..... 640, 657
  - \sidenote@margin ..... 1332, 1336
  - \sidenote@marginR ..... 1325, 1385
  - \sidenotemargin .... 1325, 2072, 2257
  - \skip ..... 2083–2085
  - \skip@lockoff ..... 462, 470
  - \skipnumbering ..... 7, 623, 2354
  - \skipnumbering@reg ..... 627
  - \smash ..... 1624
  - \splitmaxdepth ..... 2095
  - \splittopskip ..... 841, 874, 2095
  - \stanza ... 1963, 1978, 2008, 2029,  
2361, 2367, 2373, 2389, 2397, 2405
  - \stanza@count ..... 1429, 1443, 1454
  - \stanza@chang ..... 1431, 1456
  - \stanzaindentbase ..... 1454
  - \startlock ..... 614
  - \startstanzahook ..... 1427
  - \startsub ..... 581
  - \sub@action ..... 270, 418, 518
  - \sub@change ..... 64, 264, 265, 271
  - \sub@lockR ..... 61, 279, 281, 283,  
286, 436, 442, 443, 445, 446,  
476, 477, 479, 905, 940, 942,  
943, 945, 1001, 1041, 1043, 1045
  - \sub@off ..... 589, 590
  - \sub@on ..... 583, 584
  - \subline@num ..... 197,  
320, 337, 338, 340, 370, 410, 1321
  - \subline@numR .....  
. 198, 202, 287, 291, 310, 331,  
332, 334, 363, 401, 524, 528,  
906, 911, 961, 968, 1054, 1055, 1317
  - \sublinenumincrement ..... 5, 162



`\sublinenumr@p` . 1271, 1275, 1317, 1321  
`\sublinenumrepR` . . . . . [185](#), 198  
`\sublines@false` . . . . . 62, 268, 930  
`\sublines@true` . . . . . 266, 928  
`\sublock@disp` . . . . . 1003, 1007, 1012  
`\symplinenum` . . . . . 1270  
`\sza@penalty` . . . . . 1438, 1442

## T

`\textdagger` . . . . . 2282  
`\textheight` . . . . . 1931, 2335  
`\textit` 1993, 2018, 2020, 2041, 2043,  
     2053, 2193, 2206, 2227, 2232,  
     2238, 2240, 2242, 2290, 2409, 2484  
`\textnormal` . . . . . 2124  
`\textsc` . . . . . 1992, 2188  
`\textwidth` 15, 17, 697, 698, 1954, 1998  
`\theledlanguageL` 1524, [1539](#), 1598, 1711  
`\theledlanguageR` 1524, [1539](#), 1600, 1732  
`\thepage` . . . . . 564, 1291, 1301  
`\thr@@` . . . . . 445, 454, 477, 484, 935, 943  
`\title` . . . . . 2173  
`\topskip` . . . . . 1776

## U

`\unhbox` . . . . . 1466,  
     1604, 1606, 1717, 1719, 1737, 1739  
`\unhnamebox` . . . . . [1461](#)  
`\unvbox` . . . . . 846, 879, 1468  
`\unvnamebox` . . . . . [1461](#)  
`\unvxh` . . . . . 2109  
`\usernamecount` . . . . .  
     . 1430, 1437, [1472](#), 1504, 1690,  
     1890, 1899, 1901, 1910, 1919, 1921  
`\usepackage` . . . . . 1932, 2067–2069, 2334

## V

`\vAfootnote` . . . . . 1129, 1133, 1138  
`\vbadness` . . . . . 840, 873  
`\vbfnoteX` . . . . . 1413, 1417

`\vBfootnote` . . . . . 1143, 1147, 1152  
`\vbox` . . . . . 766, 795, 2107  
`\vCfootnote` . . . . . 1157, 1161, 1166  
`\vDfootnote` . . . . . 1171, 1175, 1180  
`\vEfootnote` . . . . . 1185, 1189, 1194  
`\vl@dbfnote` . . . . . 1400, 1404  
`\vl@dcnote` . . . . . 1365, 1369  
`\vl@dlnote` . . . . . 1341, 1345  
`\vl@drsnote` . . . . . 1353, 1357  
`\vsplit` . . . . . 844, 877

## W

`\wd` . . . . . 855, 888, 2111  
`\writtenlinesLfalse` . . . . . 1681, 1898  
`\writtenlinesLtrue` . . . . . 1895  
`\writtenlinesRfalse` . . . . . 1682, 1918  
`\writtenlinesRtrue` . . . . . 1915

## X

`\x@lemma` . . . . . 642–644, 659–661  
`\xlineref` . . . . . 2416, 2491  
`\xpg@main@language` . . . . . 1569, 1570  
`\xpg@set@language` . . . . . 1564, 1568  
`\xright@appenditem` . . . . .  
     . . . . . 391, 392, 394, 395, 399,  
     406, 408, 415, 420, 422, 424,  
     427, 429, 431, 439, 441, 450,  
     473, 475, 482, 501, 502, 512,  
     526, 537, 540, 544, 547, 1129,  
     1133, 1143, 1147, 1157, 1161,  
     1171, 1175, 1185, 1189, 1200,  
     1204, 1214, 1218, 1228, 1232,  
     1242, 1246, 1256, 1260, 1316,  
     1320, 1341, 1345, 1353, 1357,  
     1365, 1369, 1400, 1404, 1413, 1417

## Z

`\z@skip` . . . . . 2096  
`\zz@@@` . . . . . 1286, 1295

## Change History

|      |                                                                                                                                                      |    |                                                                                                                                                                              |                                                                                                                       |    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|----|
| v0.1 | General: First public release . . . . .                                                                                                              | 1  | <code>\normalbfnoteX</code> : Removed extraneous spaces from <code>\normalbfnoteX</code> . . . . .                                                                           | 50                                                                                                                    |    |
| v0.2 | General: Added section of babel related code . . . . .                                                                                               | 53 | <code>\Pages</code> : Added <code>\ledstrutL</code> to <code>\Pages</code> . . . . .                                                                                         | 60                                                                                                                    |    |
|      | Fix babel problems . . . . .                                                                                                                         | 1  | Added <code>\ledstrutR</code> to <code>\Pages</code> . . . . .                                                                                                               | 60                                                                                                                    |    |
|      | <code>\Columns</code> : Added <code>\l@dchecklang</code> and <code>\l@duselanguage</code> to <code>\Columns</code> . . . . .                         | 56 | <code>\Rightsidehookend</code> : Added <code>\Leftsidehook</code> , <code>\Leftsidehookend</code> , <code>\Rightsidehook</code> and <code>\Rightsidehookend</code> . . . . . | 31                                                                                                                    |    |
|      | <code>\Pages</code> : Added <code>\l@duselanguage</code> to <code>\Pages</code> . . . . .                                                            | 60 | <code>\sublinenumrepR</code> : Added <code>\linenumrepR</code> and <code>\sublinenumrepR</code> . . . . .                                                                    | 16                                                                                                                    |    |
| v0.3 | General: Reorganize for ledarab . . . . .                                                                                                            | 1  | v0.3a                                                                                                                                                                        | General: Minor <code>\linenummargin</code> fix . . . . .                                                              | 1  |
|      | <code>\affixline@numR</code> : Changed <code>\affixline@numR</code> to match new ledmac . . . . .                                                    | 39 |                                                                                                                                                                              | <code>\line@marginR</code> : Don't just set <code>\line@marginR</code> in <code>\linenummargin</code> . . . . .       | 15 |
|      | <code>\do@actions@nextR</code> : Used <code>\do@actions@fixedcode</code> in <code>\do@actionsR</code> . . . . .                                      | 37 | v0.3b                                                                                                                                                                        | General: Improved parallel page balancing . . . . .                                                                   | 1  |
|      | <code>\do@lineL</code> : Added <code>\do@lineLhook</code> to <code>\do@lineL</code> . . . . .                                                        | 35 |                                                                                                                                                                              | <code>\Pages</code> : Added <code>\l@dminpagelines</code> calculation for succeeding page pairs . . . . .             | 61 |
|      | Simplified <code>\do@lineL</code> by using macros for some common code . . . . .                                                                     | 35 | v0.3c                                                                                                                                                                        | General: Compatibilty with Polyglossia . . . . .                                                                      | 1  |
|      | <code>\do@lineR</code> : Changed <code>\do@lineR</code> similarly to <code>\do@lineL</code> . . . . .                                                | 36 | v0.4                                                                                                                                                                         | General: No more ledparpatch. All patches are now in the main file. . . . .                                           | 1  |
|      | <code>\do@lineRhook</code> : Added <code>\do@lineLhook</code> and <code>\do@lineRhook</code> . . . . .                                               | 36 | v0.5                                                                                                                                                                         | General: Corrections about <code>\section</code> and other titles in numbered sections . . . . .                      | 1  |
|      | Leftside: Added hooks into Leftside environment . . . . .                                                                                            | 31 | v0.6                                                                                                                                                                         | General: Be able to use <code>\chapter</code> in parallel pages. . . . .                                              | 1  |
|      | <code>\flag@end</code> : Removed extraneous spaces from <code>\flag@end</code> . . . . .                                                             | 27 | v0.7                                                                                                                                                                         | General: Option 'shiftedverses' which make there is no blank between two parallel verses with unequal length. . . . . | 1  |
|      | <code>\ifledRcol</code> : Moved <code>\ifl@dpairing</code> to ledmac . . . . .                                                                       | 11 |                                                                                                                                                                              |                                                                                                                       |    |
|      | <code>\ifpst@rtedR</code> : Moved <code>\ifpst@rtedL</code> to ledmac . . . . .                                                                      | 12 |                                                                                                                                                                              |                                                                                                                       |    |
|      | <code>\l@dlinenumR</code> : Simplified <code>\leftlinenumR</code> and <code>\rightlinenumR</code> by introducing <code>\l@dlinenumR</code> . . . . . | 16 |                                                                                                                                                                              |                                                                                                                       |    |
|      | <code>\l@dnumpstartsR</code> : Moved <code>\l@dnumpstartsL</code> to ledmac . . . . .                                                                | 52 |                                                                                                                                                                              |                                                                                                                       |    |
|      | <code>\ledsavedprintlines</code> : Simplified <code>\printlinesR</code> by using <code>\setprintlines</code> . . . . .                               | 46 |                                                                                                                                                                              |                                                                                                                       |    |
|      | <code>\ledstrutR</code> : Added <code>\ledstrutL</code> and <code>\ledstrutR</code> . . . . .                                                        | 61 |                                                                                                                                                                              |                                                                                                                       |    |