

# Parallel typesetting for critical editions: the `reledpar` package\*

Maïeul Rouquette<sup>†</sup> based on the original `ledpar` by Peter Wilson  
Herries Press<sup>‡</sup>

## Abstract

The `reledmac` package has been used for some time for typesetting critical editions. The `reledpar` package is an extension to `reledmac` which enables texts and their critical apparatus to be typeset in parallel, either in two columns or on pairs of facing pages.

`reledpar` provides many tools and options. Normally, they are all documented in this file. Also provided is a help folder, “examples”. The folder contains additional examples (although not for all cases). Examples starting by “3-” are for basic uses, those starting by “4-” are for advanced uses.

To report bugs, please go to `ledmac`’s GitHub page and click “New Issue”: <https://github.com/maieul/ledmac/issues/>. You must open an account with github.com to access my page (maieul/ledmac). GitHub accounts are free for open-source users. You can report bug in English or in French (better).

You can subscribe to the `reledmac` email list in:  
<http://geekographie.maieul.net/146>

## Contents

<b>1 Introduction</b>	<b>5</b>
1.1 Aim of this package . . . . .	5
1.2 Historical overview . . . . .	6
<b>2 Options</b>	<b>6</b>
2.1 Synchronization’s options . . . . .	6
2.2 Other options . . . . .	6
<b>3 General</b>	<b>7</b>

---

\*This file (`reledpar.dtx`) has version number v2.12.0, last revised 2016/07/01.

<sup>†</sup>maieul at maieul dot net

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

<b>4 Parallel columns</b>	<b>8</b>
4.1 Basic use	8
4.2 Setting	8
4.2.1 Column's width	8
4.2.2 Column's separator	8
4.2.3 Column's positions	9
4.2.4 Mixing two columns and one column texts	9
<b>5 Facing pages</b>	<b>9</b>
5.1 Basic usage	9
5.2 Setting	10
5.2.1 Text width	10
5.2.2 Way of synchronizing	10
5.2.3 Page number	12
5.2.4 Page breaking	12
5.2.5 Right page before \Pages	12
5.2.6 Notes about \mainmatter	12
5.3 Critical and familiar footnotes	13
5.3.1 Notes height setting	13
5.3.2 About the numbering of familiar footnotes	13
5.3.3 Using perpage package	13
5.3.4 Notes for one side only	14
5.3.5 Familiar notes called in the right side, but to be printed in the left side	14
5.4 Critical notes called on leftside but printed on right side	15
5.5 Using line flag	15
<b>6 Left and right texts</b>	<b>15</b>
6.1 Environments	15
6.2 Numbering text lines and paragraphs	15
6.3 Lineation system	16
6.4 Chunks	17
6.5 \AtEveryPstart and \AtEveryPstartCall	18
6.6 Language setting	18
<b>7 Verse</b>	<b>18</b>
<b>8 Side notes</b>	<b>19</b>
<b>9 Parallel ledgroups</b>	<b>19</b>
9.1 General	19
9.2 Parallel ledgroups and setspace package	20
<b>10 Sectioning commands</b>	<b>20</b>
<b>11 Notes about page number</b>	<b>21</b>
<b>I Implementation overview</b>	<b>22</b>

<b>II Preliminaries</b>	<b>22</b>
II.1 Package's meta-data . . . . .	22
II.2 Package's requirement . . . . .	22
II.3 Package's options . . . . .	22
II.4 Package's options . . . . .	23
II.4.1 Synchronization's options . . . . .	23
II.4.2 Other options . . . . .	24
II.5 Determining side and category of parallel processing . . . . .	24
II.6 Text's width . . . . .	25
II.7 Messages . . . . .	25
II.8 Naming macros . . . . .	27
<b>III Sectioning commands</b>	<b>28</b>
<b>IV Line counting</b>	<b>32</b>
IV.1 Setting lineation reset . . . . .	32
IV.2 Setting line number margin . . . . .	33
IV.3 Setting lineation start and step . . . . .	34
IV.4 Setting line flag . . . . .	35
IV.5 Setting line number style . . . . .	35
IV.6 Print marginal line number . . . . .	36
IV.7 Line-number counters and lists . . . . .	37
IV.7.1 Correspond to those in <code>reledmac</code> for regular or left text . . . . .	37
IV.7.2 Specific to <code>reledpar</code> . . . . .	37
IV.8 Reading the line-list file . . . . .	38
IV.9 Commands within the line-list file . . . . .	38
IV.10 Writing to the line-list file . . . . .	44
<b>V Marking text for notes</b>	<b>46</b>
V.1 Specific hooks and commands for notes . . . . .	46
V.1.1 Notes to be printed on one side only . . . . .	47
V.1.2 Tools specific to familiar footnotes . . . . .	47
V.1.3 Get correct footnote number . . . . .	49
V.2 Create hooks . . . . .	49
V.3 Init standards series (A,B,C,D,E,Z) . . . . .	50
V.4 Tools specific to $\LaTeX$ 's classical footnotes . . . . .	50
<b>VI Pstart numbers dumping and restoration</b>	<b>50</b>
<b>VII Parallel environments</b>	<b>51</b>
<b>VIII Paragraph decomposition and reassembly</b>	<b>54</b>
VIII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code> . . . . .	54
VIII.2 Processing one line . . . . .	59
VIII.3 Line and page number computation . . . . .	64
VIII.4 Line number printing . . . . .	67
VIII.5 Pstart number printing in side . . . . .	69

VIII.6 Add insertions to the vertical list . . . . .	71
VIII.7 Penalties . . . . .	71
VIII.8 Printing leftover notes . . . . .	72
<b>IX Footnotes</b>	<b>73</b>
IX.1 Footnotes output specific to \Pages . . . . .	73
IX.2 Critical footnote printed in right side . . . . .	76
<b>X Cross referencing</b>	<b>78</b>
<b>XI Side notes</b>	<b>79</b>
<b>XII Verse</b>	<b>80</b>
<b>XIII Fixing babel and polyglossia</b>	<b>82</b>
<b>XIV Counts and boxes for parallel texts</b>	<b>84</b>
<b>XV Checking text to be processed</b>	<b>86</b>
<b>XVI Parallel columns</b>	<b>87</b>
<b>XVII Parallel pages</b>	<b>96</b>
XVII.1 Specific counters . . . . .	96
XVII.2 Main macro . . . . .	96
XVII.3 Ensure all notes are printed at the end of parallel pages . . . . .	102
XVII.4 Struts . . . . .	103
XVII.5 Page clearing . . . . .	103
XVII.6 Lines managing . . . . .	104
XVII.7 Page break managing . . . . .	106
XVII.8 Getting boxes content . . . . .	109
<b>XVIII Page numbering</b>	<b>113</b>
XVIII.1 Global options . . . . .	113
XVIII.2 mainmatter option of \Pages . . . . .	115
<b>XIX Sections' titles' commands</b>	<b>115</b>
<b>XX Page break/no page break, depending on the specific line</b>	<b>116</b>
<b>XXI Parallel ledgroup</b>	<b>117</b>
<b>XXII Compatibility with eledmac</b>	<b>121</b>
<b>XXIII The End</b>	<b>121</b>

<b>Appendix A Some things to do when changing version</b>	<b>122</b>
Appendix A.1 Migration to <code>eledpar</code> 1.4.3 . . . . .	122
Appendix A.2 Migration from <code>eledpar</code> to <code>reledpar</code> . . . . .	122
Appendix A.2.1 Deprecated options . . . . .	122
Appendix A.2.2 <code>\renewcommandreplaced</code> with <code>command</code> . . . . .	122
Appendix A.2.3 Commands the names of which have changed . . . . .	123
Appendix A.3 Migration to <code>reledpar</code> 2.2.0 . . . . .	123
Appendix A.4 Migration to <code>reledpar</code> 2.3.0 . . . . .	123
Appendix A.5 Migration to <code>reledpar</code> 2.4.0 . . . . .	123
Appendix A.6 Migration to <code>reledpar</code> 2.5.0 . . . . .	123
Appendix A.7 Migration to <code>reledpar</code> 2.6.0 . . . . .	123
Appendix A.8 Migration to <code>reledpar</code> 2.6.1 . . . . .	123
<b>References</b>	<b>124</b>
<b>Index</b>	<b>124</b>
<b>Change History</b>	<b>142</b>

## 1 Introduction

### 1.1 Aim of this package

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 `reledpar` package is an extension to `reledmac` 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. In this case, please reports them to the author via github's issues: <https://github.com/maieul/ledmac/issues/>.

This manual contains a general description of how to use `reledpar` starting in section 3; the complete source code for the package, with extensive documentation (in sections I through XXIII); and an Index to the source code. As `reledpar` is an adjunct to `reledmac` we assume that you have read the `reledmac` manual. Also `reledpar` requires `reledmac` to be used, in the version distributed with version.

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. The documentation's sections are numbered in roman numeral.

On a first reading, We suggest that you should skip anything after the general documentation in first sections until I, unless you are particularly interested in the innards of `reledpar`.

## 1.2 Historical overview

Many of the code of this package is based on the `eledpar` package, which was based on the `ledpar`, created as an extension of the `ledmac` package.

Names of the package related to parallel typesetting have moved in parallel of names of the package related to critical edition.

Please read `reledmac`'s handbook in order to understand this evolution.

## 2 Options

The package can be loaded with a number of global options which are listed here. Those options are fully described in the paragraphs devoted to their feature.

### 2.1 Synchronization's options

Please read the paragraph on synchronization's option on 5.2.2 p. 10 to understand better those options.

**shiftedpstarts** prevents white space between paragraphs on facing pages, the white space necessary to sync pages is collected at the bottom of the page instead.

**advancedshiftedpstarts** does the same as **shiftedpstarts**, but the `pstart` shift are not counted to determine when cutting the page. That could help to avoid page with blank lines at the bottom.

**nomaxlines** allows facing pages to have different numbers of lines.

**nosyncpstarts** disables syncing on facing pages. In that case the pages are filled as two streams normal.

### 2.2 Other options

**parledgroup** allows the use of `ledgroup` environment with `reledpar`.<sup>1</sup>

**widthliketwocolumns** set the width of the text printed in a single column to be the same as the width of the text printed in two parallel columns with `reledpar`. This is useful when alternating between normal and parallel typesetting.<sup>2</sup>

**sameparallelpagenumbers** sets page numbers on facing pages to the same value.

**prevpgnotnumbered** enables that the page before facing pages (the one automatically inserted to start parallel pages on a left page) is not counted. This applies only if the page is empty.

---

<sup>1</sup>This option can either be used on `reledmac` or `reledpar`.

<sup>2</sup>This option can either be used on `reledmac` or `reledpar`.

### 3 General

A file may mix *numbered* and *unnumbered* text. Numbered text is printed with marginal line numbers and can include footnotes and endnotes that are referenced to those line numbers: this is how you will want to print the text that you are editing. Unnumbered text is not printed with line numbers, and you can't use `reledmac`'s note commands with it: this is appropriate for introductions and other material added by the editor around the edited text.

The `reledpar` package lets you typeset two *numbered* texts in parallel<sup>3</sup>. 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.

`reledmac` 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.

`reledpar` 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{5120}
```

meaning that there can be up to 5120 chunks in the left text and up to 5120 chunks in the right text, requiring a total of 10240 boxes. If you need more chunks then you can increase `\maxchunks`. The `\maxchunks` must be called in the preamble.

If you `\maxchunks` is too little you can get a `reledpar` 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 `\stanza`, 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, `reledmac` is a  $\TeX$  resource hog, and `reledpar` only makes things worse

---

<sup>3</sup>You can use, anyway, `\numberlinefalse` to disable printing of line numbers.

in this respect.

## 4 Parallel columns

### 4.1 Basic use

`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 6.

`\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} reledmac numbered text commands \end{Leftside}
\begin{Rightside} reledmac numbered text commands \end{Rightside}
\end{pairs}
\Columns
\begin{pairs}
\begin{Leftside} reledmac numbered text commands \end{Leftside}
...
\end{pairs}
\Columns
```

`\AtBeginPairs` Keep in mind that the `\Columns` **must be** outside of the `pairs` environment. You can use the macro `\AtBeginPairs` to insert a code at the beginning of each `pairs` environments. That could be useful to add the `\sloppy` macro to prevent overfull hboxes in two columns.

```
\AtBeginPairs{\sloppy}
```

There is no required pagebreak before or after the columns.

### 4.2 Setting

#### 4.2.1 Column's width

`\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.

#### 4.2.2 Column's separator

`\columnrulewidth` The macro `\columnseparator` is called between each left/right pair of lines. By default  
`\columnseparator` it inserts a vertical rule of width `\columnrulewidth`. As this is initially defined to be



Opt 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.

### 4.2.3 Column's positions

`\columnspan`

By default, columns are positioned to the right of the page. However, you can use `\columnspanposition{L}` to align them to the left, or `\columnspanposition{C}` to center them.

When you use `\stanza`, the visible rule may shift when a verse has a hanging indent. To prevent shifting, use `\setstanzaindent` outside the `Leftside` or `Rightside` environment.

`\beforecolumnseparator`

By default, the spaces around column separator are the same as the space:

`\aftercolumnseparator`

- On the left of columns, if columns are aligned right.
- On the right of columns, if columns are aligned left.
- On both the left and right columns, if columns are centered.

You can redefine `\beforecolumnseparator` and `\aftercolumnseparator` length to define spaces before or after the column separator, instead of letting `reledpar` calculate them automatically.

```
\setlength{\beforecolumnseparator}{length}
\setlength{\aftercolumnseparator}{length}
```

If you want to revert to the previous behavior, just set with a negative value.

### 4.2.4 Mixing two columns and one column texts

`\widthliketwocolumns`

If you want to mix two-column with single-column text, you can align horizontally single-column text to two-column text with `\widthliketwocolumnstrue`. To reset this feature, use `\widthliketwocolumnsfalse`. You can also use `widthliketwocolumns` as a global option when loading `reledmac` or `reledpar`.

`\Xnoteswidthliketwocolumns`

`\notesXwidthliketwocolumns`

In most cases, you should use `\widthliketwocolumns` in combination with `\Xnoteswidthliketwocolumns` and `\notesXwidthliketwocolumns` to align the critical/familiar footnotes with the two columns. See `reledmac`'s handbook for more details.

If you want to have continuous line numbers between multiple columns and single columns, use the `continuousnumberingwithcolumns` option when loading `reledmac` or `reledpar`. You will need to use `\pausenumbering... \resumenumbering` instead of `\endnumbering...endnumbering` (see 5.2.7 p. 18).

## 5 Facing pages

### 5.1 Basic usage

`pages` Numbered text that is to be set on facing pages must be within a `pages` environment.

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} reledmac numbered text commands \end{Leftside}
\begin{Rightside} reledmac numbered text commands \end{Rightside}
\begin{Leftside} reledmac numbered text commands \end{Leftside}
...
\end{pages}
\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. Note that the `\Pages` **must be** outside of the pages environment.

## 5.2 Setting

### 5.2.1 Text width

`\Lcolwidth` Within the pages environment the lengths `\Lcolwidth` and `\Rcolwidth` are the  
`\Rcolwidth` 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.

### 5.2.2 Way of synchronizing<sup>4</sup>

Synchronization of left and right texts in parallel processing requires some ‘numbered’ auxiliary files to be written (namely .1, .1R, .2, .2R, and so forth), the content of which may change as long as synchronization is not complete. This usually requires LaTeX to be run several times. Therefore, it is advised to use in conjunction utilities such as latexmk to ensure that synchronization is complete.

Numbered paragraphs which are contained between the `\pstart` and `\pend` macros are thereafter called ‘chunks’.

In short, the default setting is designed in such a way that corresponding chunks of text are always kept in synchronization, even at the cost of page padding, as it may result in leaving blank lines between chunks of text. Conversely, using in conjunction `advancedshiftedpstarts` and `nomaxlines` settings ensures that pages are filled with text to full advantage—at the cost of the chunks not being kept in synchronization—and every chunk starts on the facing page of its corresponding chunk.

To understand better how each of the synchronization settings of `reledpar` works, one must first understand how the default setting of `reledpar` synchronizes the left and right chunks.

The aim of the default setting is twofold:

---

<sup>4</sup>There is a French version of this article on <http://geekographie.maieul.net/185>.

- To ensure that left pages contain what is to be on left sides and that right pages contain what is to be on right sides.
- To ensure that every chunk starts on the page that is facing its corresponding chunk.

As regards the latter, `reledpar` checks that both of the following rules are respected:

- The numbers of lines of every pair of chunks must be identical. To keep this rule, `reledpar` may insert some blank lines at the bottom of the chunk that is shorter so that it may eventually have the same number of lines as the one that is longer.
- The main content of two facing pages, apart from critical and familiar footnotes, must have the same numbers of lines, including those that may be blank. Consequently, if one left page contains more notes than the corresponding right page, the bottom of the right page must be left blank.

Each of these rules can be modified by a number of optional synchronization settings in `reledpar`:

1. Regarding the number of lines a pair of chunks may have:
  - (a) 'shiftedpstarts' setting merely moves any added blank lines from the bottom of the chunks to the bottom of the page. It does not allow to have more lines on a given page as it just removes the blank lines between the chunks and does nothing more. To understand better how this work, you may compare the total amounts of lines of text on a given page whether you have activated this setting or not: you will see that both amounts are the same.
  - (b) 'advancedshiftedpstarts' prevents any blank lines from being inserted at the bottom of the chunks, also taking them away from the total amount of lines the page may have. This allows to get more lines on the pages. However, please note that:
    - Blank lines are taken into account as `reledpar` moves from one to the following chunk of text, so that every pair of chunks may always start on the same facing pages.
    - Consequently, blank lines continue to be taken into account in the calculation of the amount of lines a given pair of pages may have. This is why when a longer chunk runs from one page to another the shorter corresponding one also runs across pages, even if this may result in some blank vertical space being left on the first page.
2. As regards the number of lines per page, including blank ones, the `nomaxlines` setting disregards the rule that forces two facing pages to have the same numbers of lines. So it allows to have more text on the pages. Then, by a complex mechanism it is ensured that two corresponding chunks may always start on the same facing pages, provided that `shiftedpstarts` or `advancedshiftedpstarts` settings shall not be activated.

Lastly, one may disregard all of the synchronization rules and content himself with parallel texts typesetting. To achieve this, please use the `nosyncpstarts` setting.

Please note that every change of synchronization setting resets the content of the ‘numbered’ auxiliary files to make sure that `reledpar` does not try to make the synchronization with wrong calculations.

### 5.2.3 Page number

By default, `\Pages` use the standard  $\TeX$  page number scheme. This means that pages are numbered continuously following printed-book conventions: from left-hand to right-hand side, left-hand pages having even numbers, right-hand pages having odd numbers.

However, you can use the package option `sameparallelpagelnumber` to have the same page number for both left and right side. In this case, this setting will apply only for pages typeset by `\Pages`, not for “normal” pages.

Please also read advising in 11 p. 21.

### 5.2.4 Page breaking

`\setgoalfraction` When doing parallel pages `reledpar` 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 standard value is 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. You can change it using `\setgoalfraction{<newvalue>}`.

### 5.2.5 Right page before `\Pages`

When `\Pages` are called, it starts at a new left page, in order to have parallel pages. Consequently, if it is called on a left page, it clears the current page and then lets a right void page.

`reledpar` provides two options to customize this (eventual) right page.

`prevpgstyle=<style>` in order to set the style of this page. A common value of `<style>` is empty. Use `prevpgstyle=empty` will suppress header and footer in this page. Please also read advising in 11 p. 21.

`prevpgnotnumbered` will make this page won’t be counted in the page counter.

### 5.2.6 Notes about `\mainmatter`

If you use `\frontmatter`, do not use `\mainmatter` directly before `\Pages` because it could create spurious empty pages.

Use instead `\pages` with the optional argument `[mainmatter]`. In this case, the content of `\Pages` will start on a left side, without any spurious empty page, and the left pages will be odd (and not event like in normal way), the first one being 1.

### 5.3 Critical and familiar footnotes

Of course, in “Facing pages”, the `reledmac`’s both critical and familiar footnotes can be used. However, some specific points must be taken into consideration.

#### 5.3.1 Notes height setting

Since `eledpar` v1.13.0, long notes in facing pages can flow from left to right pages, and *vice-versa*.

However, the `reledmac` default setting for the maximum allotted size to notes is greater than `\textheight`. That makes impossible for long notes to flow across pages.<sup>5</sup> We have not changed this default setting, because we do not want to break compatibility with older version of `reledmac` and we want to be as close as possible to default  $\text{\LaTeX}$ ’s feature.

So, you MUST change the default setting via `\Xmaxhnotes` (for critical notes) and `\maxhnotesX` (for familiar notes). Both commands are explained in `reledmac` handbook (7.13.5 p. 44). As an advisable setting:

```
\AtBeginDocument{%
  \Xmaxhnotes{0.6\textheight}
  \maxhnotesX{0.6\textheight}
}
```

#### 5.3.2 About the numbering of familiar footnotes

If you use the same series of familiar footnotes on both sides, the numbers won’t be correct in the first run. There will be a continuous numbering for left notes, and a continuous numbering for right notes. However, after the second run, the numbering will be continuous, alternating between the left and right side. For example if you have two left pages and two right pages, with one note by page, you will obtain the following numbering at the first run: 1 (left page), 3 (right page), 2 (left page), 4 (right page). But at the next run, you will obtain: 1 (left page), 2 (right page), 3 (left page), 4 (right page).

If you use parallel columns, during the second of run of typesetting the footnote numbering will not run down the columns. Instead, it will read both column lines completely across the page, and number footnotes from left to right.

#### 5.3.3 Using `perpage` package

It follows from what has been said in the preceding paragraph that if you use the `\MakePerPage` command of the `perpage` package for footnotes called in parallel typesetting, you must append to the counter the suffix `@typeset`.

So do not set:

```
\MakePerPage{footnote}
\MakePerPage{footnoteA}
```

---

<sup>5</sup>The same applies to  $\text{\LaTeX}$  normal notes. Read <http://tex.stackexchange.com/a/228283/7712> for technical informations.

```
\MakePerPage{footnoteB}
```

But set:

```
\MakePerPage{footnote@typeset}
\MakePerPage{footnoteA@typeset}
\MakePerPage{footnoteB@typeset}
```

### 5.3.4 Notes for one side only

`\Xonlyside` You may want to typeset notes on one side only (either left or right). Use `\Xonlyside[⟨s⟩]{⟨p⟩}` to set critical notes, and `\onlysideX[⟨s⟩]{⟨p⟩}` to set familiar notes. `⟨p⟩` must be set to L for notes to be confined only on the left side and to R for notes to be confined only on the right side.

Notice that these options just tell you  $\TeX$  to not continue long notes on the other side. It is not designed to allow you to call footnotes on one side but print them on the other side.

### 5.3.5 Familiar notes called in the right side, but to be printed in the left side

`\footnoteXnomk` As often happens, the left side has less room for text. We may want to call familiar notes in the right side while using at the same time the available space in the left side to print them.

`\footnoteXmk`

To achieve this, we call `\footnoteXnomk{⟨notecontent⟩}` in the left side. X is to be replaced by the series letter. We do this call in the left side after the word which matches up to the one in the right side after which we want to insert the actual footnote mark.

In the right side, we call `\footnoteXmk` at the place we want to have the footnote mark. X is to be replaced by the series letter. For example:

```
\begin{Leftside}
\beginnumbering
\pstart
A little cat\footnoteAnomk{A note.}. And so one ...
\pend
\endnumbering
\end{Leftside}
\begin{Rightside}
\beginnumbering
\pstart
Un petit chat\footnoteAmk. And so one ...
\pend
\endnumbering
\end{Rightside}
```

## 5.4 Critical notes called on leftside but printed on right side

Sometimes, you need to print the critical notes on the right side, despite the fact that they refer to the lemmas on the left side. In this case, you must use `\edtextlater` and `\edtextnow`. The `\edtextlater{⟨lemma⟩}{⟨footnote command⟩}` command must be called on the same side as the lemma. It is similar to the standard `\edtext` command, but does not add the footnote immediately on the page, keeping it for later. `\edtextnow` must be called on the side on which you want to print the lemma, approximately at the point corresponding to the equivalent `\edtextlater` command. It will add the notes stored by the equivalent `\edtextlater` command.

The relationship between the `\edtextlater` and `\edtextnow` commands is determined by the order of calling: the first `\edtextnow` corresponds to the first `\edtextlater`, the second `\edtextnow` corresponds to the second `\edtextlater` etc.

`\edtextnow` is a parameterless macro, so it gobbles the following space. If you want to keep it, add `{ }` or backslash followed by a space.

## 5.5 Using line flag

`\Xlineflag` Use `\Xlineflag[⟨s⟩]` to add right line flag (6.3 p. 17) to right critical footnotes and  
`\Xendlineflag` `\Xendlineflag[⟨s⟩]` to add it to right critical endnotes.

# 6 Left and right texts

## 6.1 Environments

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  
`Rightside` the Rightside environment. The number of Leftside and Rightside environments must be the same.

## 6.2 Numbering text lines and paragraphs

`\beginnumbering` Each section of numbered text must be preceded by `\beginnumbering` and followed by  
`\endnumbering` `\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 T<sub>E</sub>X'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{pages}
\begin{Leftside}
  \beginnumbering
  ...
\end{Leftside}
\begin{Rightside}
  \beginnumbering
  ...
\end{Rightside}
\end{pages}
\Pages
\begin{pages}
\begin{Leftside}
  \memorydump
  ...
\end{Leftside}
\begin{Rightside}
  \memorydump
  ...
\end{pages}

```

`\numberstarttrue``\numberstartfalse``\thepstartL``\thepstartR``\skipnumbering``\hidenumbering`

It is possible to insert a number at every `\pstart` command. You must use the `\numberstarttrue` command to have it. You can stop the numbering with `\numberstartfalse`. You can redefine the commands `\thepstartL` and `\thepstartR` to change style. The numbering restarts on each `\beginnumbering`.

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

### 6.3 Lineation system

`\firstlinenum``\linenumincrement``\firstsublinenum``\sublinenumincrement`

Following `\firstlinenum{<num>}` the first line number will be `<num>`, and following `\linenumincrement{<num>}` only every `<num>`th line will have a printed number.

The lineation commands which finish by a R apply for right text. The lineation commands which are starred apply for both left and right texts. The lineation command which does not finish by a R and who are not starred apply for the left side. **However,**



**these commands apply to right side when they are called inside a left environment. However, such features should not be used any more. The recommended practice is to add all setting commands to the preamble.**

<code>\firstlinenum*</code> <code>\linenumincrement*</code> <code>\firstsublinenum*</code> <code>\sublinenumincrement*</code> <code>\firstlinenumR</code> <code>\linenumincrementR</code> <code>\firstsublinenumR</code> <code>\sublinenumincrementR</code> <code>\lineationR</code> <code>\lineation*</code> <code>\linenumberstyleR</code> <code>\sublinenumberstyleR</code> <code>\linenumberstyle*</code> <code>\sublinenumberstyle*</code> <code>\linenummarginR</code> <code>\linenummargin*</code> <code>\setRlineflag</code> <code>\linenumberLevenifblanktrue</code> <code>\linenumberRevenifblanktrue</code>	<p>The starred versions change both left and right numbering schemes.</p> <p>The suffixed version change the right side, without regard to the position they are called.</p> <p><code>\lineationR</code> macro is the equivalent of <code>reledmac \lineation</code> macro for the right side.</p> <p><code>\lineation*</code> macro is the equivalent of <code>reledmac \lineation</code> macro for both sides.</p> <p><code>\linenumberstyleR</code> is the equivalent of <code>reledmac \linenumberstyle</code> for right text. <code>\sublinenumberstyleR</code> is the equivalent of <code>reledmac \sublinenumberstyle</code> right text. The starred version are for both side.</p> <p><code>\linenummarginR{&lt;margin&gt;}</code> sets the line margin for right side. <code>\linenummargin*{&lt;margin&gt;}</code> sets for both side. <code>&lt;margin&gt;</code> can be, as for <code>reledmac's \linenummargin</code> one of these values: left, right, inner, outer. A "R" is appended to the line numbers of the right texts. This may be useful for parallel columns but for parallel pages it might be more appropriate to redefine it using <code>\setRlineflag{&lt;flag&gt;}</code>. Use <code>\setRlineflag{}</code> to empty it.</p> <p>By default, when a blank line is printed on one side, in order to synchronize with the other side, no line number is printed. However, you can decide to print them for blank lines, also. Use <code>\linenumberLevenifblanktrue</code> to enable it on the left side, and <code>\linenumberRevenifblanktrue</code> to enable it on right side.</p>
--	--

## 6.4 Chunks

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

```

\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.

`\autopar`

The `\autopar` macro can be used, instead of manually inserting `\pstart... \pend`s. Please read `reledmac`'s handbook (5.2.2 p. 17).

## 6.5 `\AtEveryPstart` and `\AtEveryPstartCall`

In general, remember that the moment where a `\pstart` is called is different from the moment when the `\pstart... \pend` content is printed, which is when `\Pages` or `\Columns` is processed.

Consequently:

- The argument of `\AtEveryPstart` (see 5.2.4 p. 18) is called before every chunk is printed, except if you used an optional argument for the `\pstart`.
- The argument of `\AtEveryPstartCall` is called before every `\pstart`.

## 6.6 Language setting

If you are using the `babel` package or the `polyglossia` 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* language setting 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.

## 7 Verse

If you are typesetting verses with `reledmac` 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`

`reledpar` provides an `astanza` environment which you can use instead of `\stanza`. A `astanza` environment is a chunk. Consequently left and right *verse* are matched, and not, as with standard `\stanza`, left and right *verse lines*.

Within the `astanza` environment each verse line is treated as an individual paragraph, so there must be no blank lines in the environment otherwise there will be some extraneous vertical spacing. To use `astanza`, simply replace `\stanza` by `\begin{astanza}` and add `\end{astanza}` after the ending `\&`.

The difference between `astanza` and `\stanza` is, that the letter syncs verse by verse, while the environment syncs stanza by stanza.

If you get an error message along the lines of ‘Missing number, treated as zero \sza@00’ it is because you have forgotten to use `\setstanzaindent` to set the stanza indents.

As `astanza` is a specify type `\pstart...\pend` structure, you can:

- Add optional argument (in brackets) after `\begin{astanza}`, as the optional argument of `\pstart`.
- Use optional argument after the last `\&` as optional argument of `\pend`.

`\sethangingsymbol` Like in `reledmac`, you could use the `\sethangingsymbol` command to insert a character in each hanging line. If you use it, you must run  $\TeX$  two time. Example for the French typography

```
\sethangingsymbol{[,]}
```

You can also use it to force hanging verse to be flush right:

```
\sethangingsymbol{\protect\hfill}
```

When you use `\lednopb` make sure to use it on both sides in the corresponding verses to keep the pages in sync.

`\thestanzaL` When using `\stanzanumtrue` (9.9 p. 48) in parallel typesetting, stanza counter is replaced by `stanzaL` counter in left side and by `stanzaR` counter in right side. Consequently, you can redefine `\thestanzaL` and `\thestanzaR` to change their aspect.

## 8 Side notes

As in `reledmac`, you must use one of the following commands to add side notes: `\ledsidenote`, `\ledleftnote`, `\ledrightnote`, `\ledouterote`, `\ledinnerrote`.

The `\sidenotemargin` defines the margin of the sidenote for either left or right side, depending on the current environment. You can use `\sidenotemargin*` to define it for both sides.

## 9 Parallel ledgroups

### 9.1 General

You can also make parallel ledgroups (see the documentation of `reledmac` about ledgroups, 10 p. 49). To do it you have:

- To load `reledpar` package with the `parledgroup` option, or to add `\parledgrouptrue`.
- To push each ledgroup between `\pstart...\pend` command.

See the following example:

```
\begin{pages}
```

```

\begin{Leftside}
  \beginnumbering
  \pstart
  \begin{ledgroup}
    ledgroup content
  \end{ledgroup}
\pend
\pstart
  \begin{ledgroup}
    ledgroup content
  \end{ledgroup}
\pend
\endnumbering
\end{Leftside}
\begin{Rightside}
  \beginnumbering
  \pstart
  \begin{ledgroup}
    ledgroup content
  \end{ledgroup}
\pend
\pstart
  \begin{ledgroup}
    ledgroup content
  \end{ledgroup}
\pend
\endnumbering
\end{Rightside}
\end{pages}
\Pages

```

## 9.2 Parallel ledgroups and setspace package

If you use the `setspace` package and want your notes in parallel ledgroups to be single-spaced (not half-spaced or double-spaced), just add to your preamble:

```
\setparledgroupnotespacing{\singlespacing}
```

*In effect, to have correct spacing, do not change the font size of your notes.*

## 10 Sectioning commands

The standard sectioning commands of `reledmac` are available, and provide parallel sectioning, for both two-column and two-page layout.

`\eledsectnotoc`

By default, the section commands of the right side are not added to the table of contents. But you can change it, using `\eledsectnotoc{⟨arg⟩}`, where `⟨arg⟩` could be L (for left side) or R (for right side).

`\eledsectmark` By default, the headers are tokens from the left side. You can change them, using `\eledsectmark{⟨arg⟩}`, where `⟨arg⟩` could be L (for left side) or R (for right side).

## 11 Notes about page number

If you use `sameparallepagenumber` option (5.2.3 p. 12 or `prevpgnotnumbered` option (5.2.5 p. 12), please read the following paragraph if you want to manipulate page numbers manually.

In order to implement these two options, `reledpar` uses its own page counter, called `par@page`. Consequently, if you use at least one of these options:

1. If you modify `\thepage` command, use the value of `par@page` counter inside and not the value of `page` counter.
2. If you want to modify a page number, modify the value of `page` counter AND the value `par@page` counter.

Notes that `reledpar` automatically do it when you use `\frontmatter` and `\mainmatter` commands.

## I Implementation overview

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

`reledmac` 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  $\text{\TeX}$  to put them onto the page with the appropriate page breaks. Most of the `reledmac` code is concerned with handling this box and its contents.

`reledpar`’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.

## II Preliminaries

### II.1 Package’s meta-data

Announce the name and version of the package, which is targeted for  $\text{\LaTeX}2\text{\epsilon}$ . The package also requires the `reledmac` package, however we do not load it automatically, because we prefer users to know it.

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledpar}[2016/07/01 v2.12.0 reledmac extension for
  parallel texts]%
4
5 %
```

### II.2 Package’s requirement

Few commands use `\xspace` command.

```

6 \RequirePackage{xspace}%
7 %
```

### II.3 Package’s options

We use `xkeyval` in order to manage options with arguments.

```

8 \RequirePackage{xkeyval}
9 %
```

## II.4 Package's options

### II.4.1 Synchronization's options

`\@par@this@sync@option` The `\par@sync@option` stores the options of synchronization. It use to ensure these options do not change between two run.

```
10 \def\@par@this@sync@option{%
11 %
```

With the option 'shiftedpstarts' a long pstart on the left side (or on the right side) does not make a blank on the corresponding pstart, but the blank is put on the bottom of the page. Consequently, the pstarts on the parallel pages are shifted, but the shift stops at every end of pages.

```
\ifshiftedpstarts12 \newif\ifshiftedpstarts
13 \DeclareOptionX{shiftedpstarts}{%
14   \shiftedpstartstrue%
15   \apptocmd{\@par@this@sync@option}{shifted}{-}{-}%
16 }%
17 %
```

With the option 'advancedshiftedpstarts' a long pstart on the left side (or on the right side) does not make a blank on the corresponding pstart, but the blank is put on the bottom of the page. Consequently, the pstarts on the parallel pages are shifted, but the shift stops at every end of pages. Differing to `shiftedpstarts`, the pstart shift are not counted to determine when cutting the page. That could help to avoid page with blank lines at the bottom.

```
\ifshiftedpstarts18 \newif\ifadvancedshiftedpstarts
19 \DeclareOptionX{advancedshiftedpstarts}{%
20   \advancedshiftedpstartstrue%
21   \shiftedpstartstrue%
22   \apptocmd{\@par@this@sync@option}{advancedshifted}{-}{-}%
23 }%
24 %
```

With the option `nomaxlines`, `reledpar` allows facing pages to have not the same number of lines.

```
\ifnomaxlines25 \newif\ifnomaxlines%
26 \DeclareOptionX{nomaxlines}{%
27   \nomaxlinestrue%
28   \apptocmd{\@par@this@sync@option}{nomax}{-}{-}%
29 }%
30 %
```

With the option `nosyncpstarts`, `reledpar` only alternate between left and right side, and does not try to obtain the same number of line in corresponding page.

```

\ifnosyncpstarts 31 \newif\ifnosyncpstarts%
32 \DeclareOptionX{nosyncpstarts}{%
33   \shiftedpstartstrue%
34   \nomaxlinestrue%
35   \nosyncpstartstrue%
36   \apptocmd{\@par@this@sync@option}{nosync}{-}{-}%
37 }%
38 %

```

### II.4.2 Other options

The `parledgroup` can be called either on `reledmac` or `reledpar`.

```

39 \DeclareOptionX{parledgroup}{\parledgrouptrue}
40 %

```

`\ifwidthliketwocolumns` The `widthliketwocolumns` and `continuousnumberingwithcolumns` options can be called either on `reledmac` or `reledpar`.

```

41 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
42 \DeclareOptionX{continuousnumberingwithcolumns}{\
continuousnumberingwithcolumnstrue}%
43 %

```

`\ifsameparallelpagenumber` Options related to page numbering

`\ifprevpgnotnumbered`

```

44 \newif\ifsameparallelpagenumber
45 \newif\ifprevpgnotnumbered
46 \DeclareOptionX{sameparallelpagenumber}{\sameparallelpagenumbertrue}
47 \DeclareOptionX{prevpgnotnumbered}{\prevpgnotnumberedtrue}
48 %

```

`\prevpgstyle` We store on `\prevpgstyle` the argument of the option `prevpgstyle`.

```

49 \DeclareOptionX{prevpgstyle}{\gdef\prevpgstyle{#1}}%
50 %

```

```

51 \ProcessOptionsX%
52 %

```

## II.5 Determining side and category of parallel processing

As noted above, much of the code is a duplication of the original `reledmac` 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 we 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` is also set TRUE if we are doing parallel pages. `\ifledRcol` is set TRUE if we are doing the right hand text. They are defined in `reledmac`.

## II.6 Text's width

`\Lcolwidth` The widths of the left and right parallel columns (or pages).

```
\Rcolwidth
53 \newdimen\Lcolwidth
54 \Lcolwidth=0.45\textwidth
55 \newdimen\Rcolwidth
56 \Rcolwidth=0.45\textwidth
57 %
```

## II.7 Messages

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

```
\reledpar@error58 \newcommand{\reledpar@error}[2]{\PackageError{reledpar}{#1}{#2}}
59 %
```

```
\reledpar@warning60 \newcommand{\reledpar@warning}[1]{\PackageWarning{reledpar}{#1}}%
61 %
```

```
\led@err@TooManyPstarts62 \newcommand*\led@err@TooManyPstarts{%
63 \reledpar@error{Too many \string\pstart\space without printing.
64 Some text will be lost}{\@ehc}}
65 %
```

```
d@err@BadLeftRightPstarts66 \newcommand*\led@err@BadLeftRightPstarts}[2]{%
67 \reledpar@error{The numbers of left (#1) and right (#2)
68 \string\pstart s do not match}{\@ehc}}
69 %
```

```
\led@err@LeftOnRightPage70 \providebool{syntax@}
\led@err@RightOnLeftPage71 \newcommand*\led@err@LeftOnRightPage{%
72 \notbool{syntax@}%
73 {\reledpar@error{The left page has ended on a right page}{\@ehc}}%
74 {}%
75 }
76 \newcommand*\led@err@RightOnLeftPage{%
77 \notbool{syntax@}%
78 {\reledpar@error{The right page has ended on a left page}{\@ehc}}
79 {}%
80 }%
81 %
```

```

\led@err@Leftside@PreviousNotPrinted82 \newcommand*{\led@err@Leftside@PreviousNotPrinted}{%
\led@err@Rightside@PreviousNotPrinted83   \reledpar@error{You call a new Leftside environment while the previous
one has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}
84 \newcommand*{\led@err@Rightside@PreviousNotPrinted}{%
85   \reledpar@error{You call a new Rightside environment while the previous
one has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}
86 %

\led@err@Pages@InsideEnv87 \newcommand*{\led@err@Pages@InsideEnv}{%
\led@err@Columns@InsideEnv88   \reledpar@error{\string\Pages\space must be called *outside* of the `
pages` environment}{\@ehc}}
89 \newcommand*{\led@err@Columns@InsideEnv}{%
90   \reledpar@error{\string\Columns\space must be called *outside* of the `
pairs` environment}{\@ehc}}
91 %

\led@err@Pages@WithoutEnv92 \newcommand*{\led@err@Pages@WithoutEnv}{%
\led@err@Columns@WithoutEnv93   \reledpar@error{\string\Pages\space called without previous `pages`
environment}{\@ehc}}
94 \newcommand*{\led@err@Columns@WithoutEnv}{%
95   \reledpar@error{\string\Columns\space called without previous `pairs`
environment}{\@ehc}}
96 %

\led@error@fail@patch@thepage97 \newcommand{\led@error@fail@patch@thepage}{%
98   \reledpar@error{Fail to patch \string\thepage\space command.}{\@ehc}%
99 }%
100 %

\led@error@fail@patch@pagenumbering101 \newcommand{\led@error@fail@patch@pagenumbering}{%
102   \reledpar@error{Fail to patch \string\pagenumbering\space command.}{\@ehc
}%
103 }%
104 %

\led@error@note@called@onrightside105 \newcommand{\led@error@note@called@onrightside}[1]{%
\led@error@note@called@onleftside106   \reledpar@error{#1 called on right side, despite your configuring it to
be for the left side only}{\@ehc}%
107 }%
108 \newcommand{\led@error@note@called@onleftside}[1]{%
109   \reledpar@error{#1 called on left side, despite your configuring it to be
for the right side only}{\@ehc}%
110 }%
111 %

```

```

error@fail@patch@@mempnum112 \newcommand{\led@error@fail@patch@@mempnum}{%
113 \reledpar@error{Fail to patch \string\@mempnum\space command.}\@ehc}%
114 }%
115 %

or@fail@patch@@outputpage116 \newcommand{\led@error@fail@patch@@outputpage}{%
117 \reledpar@error{Fail to patch \string\@outputpage\space command.}\@ehc}%
118 }%
119 %

ed@error@edtext@later@now20 \newcommand{\led@error@edtext@later@now}{%
121 \reledpar@error{You call \the\edtext@later\space \string\edtextlater\
space commands, but \the\edtext@now\space \string\edtextnow\space commands
}\@ehc}% (did I follow your intent correctly here?)
122 }%
123 %

led@warn@ChangeSyncOption24 \newcommand*{\led@warn@ChangeSyncOption}[1]{%
125 \reledpar@warning{You have changed synchronization's options since the
last run. We have not read line-list file #1. Please run LaTeX again.}%
126 }%
127 %

warn@setting@in@rightside28 \newcommand{\led@warn@setting@in@rightside}[1]{%
129 \reledpar@warning{You use #1 inside rightside environment.\MessageBreak%
130 Such behavior is deprecated.\MessageBreak%
131 Use instead #1R or #1* in your preamble.}%
132 }
133 %

d@error@missing@numbering34 \newcommand{\led@error@missing@numbering}[1]{%
135 \reledpar@error{Missing \string\...pstart\string\pend\space inside `#1`
environment}\@ehc}%
136 }%
137 %

```

## II.8 Naming macros

The  $\text{\LaTeX}$  kernel provides  $\backslash\@namedef$  and  $\backslash\@namuse$  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.

$\backslash\newnamebox$  A set of macros for creating and using ‘named’ boxes; the macros are called after the  
 $\backslash\setnamebox$  regular box macros, but including the string ‘name’.  
 $\backslash\unhnamebox$   
 $\backslash\unvnamebox$   
 $\backslash\namebox$

```

138 \providecommand*\newnamebox}[1]{%
139   \expandafter\newbox\csname #1\endcsname}
140 \providecommand*\setnamebox}[1]{%
141   \expandafter\setbox\csname #1\endcsname}
142 \providecommand*\unhnamebox}[1]{%
143   \expandafter\unhbox\csname #1\endcsname}
144 \providecommand*\unvnamebox}[1]{%
145   \expandafter\unvbox\csname #1\endcsname}
146 \providecommand*\namebox}[1]{%
147   \csname #1\endcsname}
148
149 %

```

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

```

\usernamecount
150 \providecommand*\newnamecount}[1]{%
151   \expandafter\newcount\csname #1\endcsname}
152 \providecommand*\usernamecount}[1]{%
153   \csname #1\endcsname}
154
155 %

```

### III 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.

```

156 \newcount\section@numR
157 \section@numR=\z@
158 %

```

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

```

159 \pst@rtedLfalse
160 \newif\ifpst@rtedR
161
162 %

```

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

```

163 \newcommand*\beginnumberingR{%
164   \ifnumberingR

```

```

165     \led@err@NumberingStarted
166     \endnumberingR
167     \fi
168     \global\l@dnumpstartsR \z@
169     \global\pst@rtedRfalse
170     \global\numberingRtrue
171     \global\advance\section@numR \@ne
172     \global\absline@numR \z@
173     \gdef\normal@page@breakR{}
174     \gdef\l@prev@pbR{}
175     \gdef\l@prev@nopbR{}
176     \global\line@numR \z@
177     \global\@lockR \z@
178     \global\sub@lockR \z@
179     \global\sublines@false
180     \global\let\next@page@numR\relax
181     \global\let\sub@change\relax
182     \global\stopmsdata@inserted@true%
183     \set@continuousnumberingforR%
184     \message{Section \the\section@numR R }%
185     \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
186     \l@dend@stuff
187     \setcounter{pstartR}{1}
188     \begingroup
189     \initnumbering@sectcountR
190     \gdef\eled@sectionsR@{ }%
191     \if@noeled@sec\else%
192         \makeatletter\inputIfFileExists{\jobname.eledsec\the\section@numR R
193         }{}{\makeatother%
194         \immediate\openout\eled@sectioningR@out=\l@auxdir\jobname.eledsec\the\
195         section@numR R\relax%
196         \fi%
197     }
198     %

```

**\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 `reledmac`.

**\endnumberingR** This is the right text equivalent of `\endnumbering` and must follow the last text for a right text numbered section.

```

197 \def\endnumberingR{%
198     \ifnumberingR
199         \global\numberingRfalse
200         \normal@pars
201         \ifnum\l@dnumpstartsR=0%
202             \led@err@NumberingWithoutPstart%
203         \fi%
204         \ifl@dpairing

```

```

205 \global\pst@rtedRfalse
206 \else
207 \ifx\insertlines@listR\empty\else
208 \global\noteschanged@true
209 \fi
210 \ifx\line@listR\empty\else
211 \global\noteschanged@true
212 \fi
213 \fi
214 \ifnoteschanged@
215 \led@mess@NotesChanged
216 \fi
217 \else
218 \led@err@NumberingNotStarted
219 \fi
220 \endgroup
221 \if@noeled@sec\else%
222 \immediate\closeout\eled@sectioningR@out%
223 \fi%
224 }
225
226 %

```

`\initnumbering@sectcountR` We do not want the right side section commands to be numbered after the left side ones, instead we want them numbered after which is typeset before the pages or columns environments. we switch the  $\LaTeX$  counter in `\numberingR`.

`\save@section@number`

`\set@sectcountR`

```

227 \newcounter{chapterR}
228 \newcounter{sectionR}
229 \newcounter{subsectionR}
230 \newcounter{subsubsectionR}
231
232 \newcount\old@chapter%
233 \newcount\old@section%
234 \newcount\old@subsection%
235 \newcount\old@subsubsection%
236 \newcommand{\save@section@number}{%
237 \ifdefined\c@chapter%
238 \global\old@chapter\value{chapter}%
239 \fi%
240 \global\old@section\value{section}%
241 \global\old@subsection\value{subsection}%
242 \global\old@subsubsection\value{subsubsection}%
243 }%
244 \newcommand{\initnumbering@sectcountR}{
245 \ifdefined\c@chapter%
246 \setcounter{chapterR}{\old@chapter}%
247 \fi%
248 \setcounter{sectionR}{\old@section}%
249 \setcounter{subsectionR}{\old@subsection}%

```

```

250 \setcounter{subsubsectionR}{\old@subsubsection}%
251 \set@sectcountR%
252 }
253 \newcommand{\set@sectcountR}{%
254   \let\c@chapter\c@chapterR%
255   \let\c@section\c@sectionR%
256   \let\c@subsection\c@subsectionR%
257   \let\c@subsubsection\c@subsubsectionR%
258 }%
259 %

```

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

```

\resumenumberingR
260 \newcommand*{\pausenumberingR}{%
261   \endnumberingR\global\numberingRtrue}
262 \newcommand*{\resumenumberingR}{%
263   \ifnumberingR
264     \global\pst@rtedRtrue
265     \global\advance\section@numR \@ne
266     \led@mess@SectionContinued{\the\section@numR R}%
267     \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
268     \ldend@stuff
269     \begingroup%
270     \initnumbering@sectcountR%
271     \set@continuousnumberingforR%
272   \else
273     \led@err@numberingShouldHaveStarted
274     \endnumberingR
275     \beginnumberingR
276   \fi}
277
278 %

```

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

```

279 \newcommand*{\memorydumpL}{%
280   \endnumbering
281   \numberingtrue
282   \global\pst@rtedLtrue
283   \global\advance\section@num \@ne
284   \led@mess@SectionContinued{\the\section@num}%
285   \line@list@stuff{\jobname.\extensionchars\the\section@num}%
286   \ldend@stuff}
287
288 \newcommand*{\memorydumpR}{%
289   \endnumberingR
290   \numberingRtrue
291   \global\pst@rtedRtrue
292   \global\advance\section@numR \@ne

```

```

293 \led@mess@SectionContinued{\the\section@numR R}%
294 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
295 \l@dend@stuff}
296
297 %

```

## IV Line counting

### IV.1 Setting lineation reset

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

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

```

298 \newcommand*{\lineationR}[1]{%
299   \ifnumbering
300     \led@err@LineationInNumbered
301   \else
302     \def\@tempa{#1}\def\@tempb{page}%
303     \ifx\@tempa\@tempb
304       \global\bypage@Rtrue
305       \global\bypstart@Rfalse
306       \unless\ifnocritical@%
307         \Xpstart[] [false]%
308       \fi%
309     \else
310       \def\@tempb{pstart}%
311       \ifx\@tempa\@tempb
312         \global\bypage@Rfalse
313         \global\bypstart@Rtrue
314         \unless\ifnocritical@%
315           \Xpstart%
316         \fi%
317       \else
318         \def\@tempb{section}
319         \ifx\@tempa\@tempb
320           \global\bypage@Rfalse%
321           \global\bypstart@Rfalse%
322           \unless\ifnocritical@%
323             \Xpstart[] [false]%
324           \fi%
325         \else
326           \led@warn@BadLineation
327         \fi%
328       \fi

```



```

329 \fi
330 \fi}}
331 %

```

`\set@continuousnumberingforR` set the right line numbers at a `\beginnumberingR` or a `\resumenumberingR` in order to have continuous numbering with single column text.

```

332 \newcommand{\set@continuousnumberingforR}{%
333 \ifcontinuousnumberingwithcolumns%
334 \ifnum\line@numR<\line@num%
335 \expandafter\setlinenum\expandafter{\the\line@num}%
336 \fi%
337 \ifnum\last@page@num>\last@page@numR%
338 \global\last@page@numR=\last@page@num%
339 \fi%
340 \fi%
341 }
342 %

```

`\lineation*` `\lineation*` change the lineation system for both sides.

```

343 \WithSuffix\newcommand\lineation*[1]{%
344 \lineation{#1}%
345 \lineationR{#1}%
346 }%
347 %

```

## IV.2 Setting line number margin

`\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 would like; if it is 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.

It is defined only once time, in `reledmac`.

```

348 \newcount\line@marginR
349 %

```

By default put right text numbers at the right.

```

350 \line@marginR=\@ne
351
352 %

```

`\linenummarginR` `\linenummarginR` applies directly for right side, while `\linenummargin*` applies for both side.

```

353 \newcommand{\linenummarginR}[1]{%
354   \l@getline@margin{#1}%
355   \ifnum\@l@tempcntb>\m@ne%
356     \global\line@marginR=\@l@tempcntb%
357   \fi%
358 }
359 \WithSuffix\newcommand\linenummargin*[1]{%
360   \l@getline@margin{#1}%
361   \ifnum\@l@tempcntb>\m@ne%
362     \global\line@marginR=\@l@tempcntb%
363     \global\line@margin=\@l@tempcntb%
364   \fi%
365 }
366 %

```

### IV.3 Setting lineation start and step

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

```

367 \newcounter{firstlinenumR}
368   \setcounter{firstlinenumR}{5}
369 \newcounter{linenumincrementR}
370   \setcounter{linenumincrementR}{5}
371 %

```

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

```

372 \newcounter{firstsublinenumR}
373   \setcounter{firstsublinenumR}{5}
374 \newcounter{sublinenumincrementR}
375   \setcounter{sublinenumincrementR}{5}
376
377 %

```

`\firstlinenum` `\firstlinenum` These are the user's macros for changing (sub) line numbers. They are defined in reledmac. The starred versions are specific to eledpar.

`\linenumincrement` `\linenumincrement`

`\firstsublinenum` `\firstsublinenum`

`\sublinenumincrement` `\sublinenumincrement`

`\firstlinenum*` `\firstlinenum*`

`\linenumincrement*` `\linenumincrement*`

`\firstsublinenum*` `\firstsublinenum*`

`\sublinenumincrement*` `\sublinenumincrement*`

```

383 \setcounter{linenumincrementR}{#1}%
384 \setcounter{linenumincrement}{#1}%
385 }
386 \WithSuffix\newcommand\firstsublinenum*[1]{%
387 \setcounter{subfirstlinenumR}{#1}%
388 \setcounter{subfirstlinenum}{#1}%
389 }
390 \WithSuffix\newcommand\sublinenumincrement*[1]{%
391 \setcounter{sublinenumincrementR}{#1}%
392 \setcounter{sublinenumincrement}{#1}%
393 }
394 %

```

`\firstlinenumR` And the ‘R’ suffixed version.

```

\linenumincrementR
\firstsublinenumR
\sublinenumincrementR
395 \newcommand\firstlinenumR[1]{%
396 \setcounter{firstlinenumR}{#1}%
397 }
398 \newcommand\linenumincrementR[1]{%
399 \setcounter{linenumincrementR}{#1}%
400 }
401 \newcommand\firstsublinenumR[1]{%
402 \setcounter{subfirstlinenumR}{#1}%
403 }
404 \newcommand\sublinenumincrementR[1]{%
405 \setcounter{sublinenumincrementR}{#1}%
406 }
407 %

```

## IV.4 Setting line flag

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

```

408 \newcommand{\setRlineflag}[1]{%
409 \gdef\Rlineflag{#1}%
410 }
411 \setRlineflag{R}
412 %

```

## IV.5 Setting line number style

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

```

413 \newcommand*\linenumrepR[1]{\@arabic{#1}}
414 \newcommand*\sublinenumrepR[1]{\@arabic{#1}}
415
416 %

```

`\linenumberstyleR` The style can be changed by some user level command  
`\sublinenumberstyleR`

```

417 \newcommand*\linenumberstyleR[1]{%
418   \def\linenumrepR##1{\@nameuse{##1}}
419 \newcommand*\sublinenumberstyleR[1]{%
420   \def\sublinenumrepR##1{\@nameuse{##1}}
421 %

```

`\linenumberstyle*` And for both side.  
`\sublinenumberstyle*`

```

422 \WithSuffix\newcommand\linenumberstyle*[1]{%
423   \linenumberstyle{#1}%
424   \linenumberstyleR{#1}%
425 }%
426
427 \WithSuffix\newcommand\sublinenumberstyle*[1]{%
428   \sublinenumberstyle{#1}%
429   \sublinenumberstyleR{#1}%
430 }%
431 %
432 %

```

## IV.6 Print marginal line number

`\iflinenumberLevenifblank` `\iflinenumberLevenifblank` and `\iflinenumberRevenifblank` can be switched to TRUE if we want to print the line number, even if the line is blank.

```

433 \newif\iflinenumberLevenifblank
434 \newif\iflinenumberRevenifblank
435 %

```

`\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`.

```

436 \newcommand*\leftlinenumR{%
437   \l@dlinenumR
438   \kern\linenumsep}
439 \newcommand*\rightlinenumR{%
440   \kern\linenumsep
441   \l@dlinenumR}
442 \newcommand*\l@dlinenumR{%
443   \numlabfont\linenumrepR{\line@numR}\@Rlineflag%
444   \ifsublines@
445     \ifnum\subline@num>\z@
446       \unskip\fullstop\sublinenumrepR{\subline@numR}%
447     \fi
448   \fi}
449
450 %

```

## IV.7 Line-number counters and lists

### IV.7.1 Correspond to those in `reledmac` for regular or left text

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

`\line@numR` The count `\line@numR` stores the line number that is 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 have actually printed, no matter what numbers we attached to them.

```
451 \newcount\line@numR
452 \newcount\subline@numR
453 \newcount\absline@numR
454
455 %
```

`\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 `reledmac` manual.

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

`\actionlines@listR`

`\actions@listR`

```
456 \list@create{\line@listR}
457 \list@create{\insertlines@listR}
458 \list@create{\actionlines@listR}
459 \list@create{\actions@listR}
460
461 %
```

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

```
462 \newcount\page@numR
463
464 %
```

### IV.7.2 Specific to `reledpar`

`\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.

`\linesinpar@listR`

`\maxlinesinpar@list`

```
465 \list@create{\linesinpar@listL}
466 \list@create{\linesinpar@listR}
467 \list@create{\maxlinesinpar@list}
468
469 %
```

## IV.8 Reading the line-list file

`\list@clearing@regR` \Clear the right lines for \read@linelist

```

470 \newcommand{\list@clearing@regR}{%
471   \list@clear{\line@listR}%
472   \list@clear{\insertlines@listR}%
473   \list@clear{\actionlines@listR}%
474   \list@clear{\actions@listR}%
475   \list@clear{\linesinpar@listR}%
476   \list@clear{\linesonpage@listR}
477 }
478 %

```

`\@par@sync@option` When typesetting parallel pages, `\@par@sync@option` check if we have changed the synchronization's option since the last run. If true, we just not read the numbered file.

```

479 \newcommand{\@par@sync@option}[1]{%
480   \IfStrEq{#1}{\@par@this@sync@option}%
481   {}%
482   {\ifledRcol%
483     \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}%
484     %
485     \else%
486     \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}%
487     %
488     \fi%
489     \endinput%
490   }%
491 }%
492 %

```

`\read@linelist` `\read@linelist{⟨file⟩}` is the control sequence that is called by `\beginnumbering` (via `\line@list@stuff`) to open and process a line-list file; its argument is the name of the file. . It is defined only once time in `reledmac`.

## IV.9 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.

`\@nl@regR` `\@nl@regR` is called by `\@nl` if we are on a right side. It does everything related to the start of a new line of numbered text on a right side.

```

491 \newcommand{\@nl@regR}{%
492   \ifx\l@dchset@num\relax \else
493   \advance\absline@numR \@ne

```

```

494 \set@line@action
495 \let\l@dchset@num\relax
496 \advance\absline@numR \m@ne
497 \advance\line@numR \m@ne% % do we need this?
498 \fi
499 \advance\absline@numR \@ne
500 \ifx\next@page@numR\relax \else
501 \page@action
502 \let\next@page@numR\relax
503 \fi
504 \ifx\sub@change\relax \else
505 \ifnum\sub@change>\z@
506 \sublines@true
507 \else
508 \sublines@false
509 \fi
510 \sub@action
511 \let\sub@change\relax
512 \fi
513 \ifcase\@lockR
514 \or
515 \@lockR \tw@
516 \or\or
517 \@lockR \z@
518 \fi
519 \ifcase\sub@lockR
520 \or
521 \sub@lockR \tw@
522 \or\or
523 \sub@lockR \z@
524 \fi
525 \ifsublines@
526 \ifnum\sub@lockR<\tw@
527 \advance\subline@numR \@ne
528 \fi
529 \else
530 \ifnum\@lockR<\tw@
531 \advance\line@numR \@ne \subline@numR \z@
532 \fi
533 \fi}
534
535
536 %

```

`\last@page@numR` `\last@page@numR` store the page number of the last right page. It is modified by `\fix@page` `\fix@page`, defined by `reledmac`.

```

537 \newcount\last@page@numR
538 \last@page@numR=-10000
539

```

540 %

- \@adv** The `\@adv{⟨num⟩}` macro advances the current visible line number by the amount specified as its argument. This is used to implement `\advanceline`. It is defined in `reledmac`.
- \@set** The `\@set{⟨num⟩}` macro sets the current visible line number to the value specified as its argument. This is used to implement `\setline`. It is defined in `reledmac`.
- \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`. It is defined in `reledmac`.
- \page@action** `\page@action` adds an entry to the action-code list to change the page number. It is defined in `reledmac`.
- \set@line@action** `\set@line@action` adds an entry to the action-code list to change the visible line number. It is defined in `reledmac`.
- \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. It is defined in `reledmac`.
- \do@lockon** `\lock@on` adds an entry to the action-code list to turn line number locking on. The current setting of the sub-lineation flag tells us whether this applies to line numbers or sub-line numbers. It is defined in `reledmac`, however the code specific to right side is defined here, in `\do@lockonR`.

```

541 \newcount\@lockR
542 \newcount\sub@lockR
543
544 \newcommand*{\do@lockonR}{%
545   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
546   \ifsublines@
547     \xright@appenditem{-1005}\to\actions@listR
548     \ifnum\sub@lockR=\z@
549       \sub@lockR \@ne
550     \else
551       \ifnum\sub@lockR=\thr@@
552         \sub@lockR \@ne
553       \fi
554     \fi
555   \else
556     \xright@appenditem{-1003}\to\actions@listR
557     \ifnum\@lockR=\z@
558       \@lockR \@ne
559     \else
560       \ifnum\@lockR=\thr@@
561         \@lockR \@ne
562       \fi
563     \fi

```



```

564 \fi}
565
566 %

```

`\lock@off` `\lock@off` adds an entry to the action-code list to turn line number locking off. It is defined in `reledmac`, however the code specific to right side is defined here, in `\do@lockoffR`.

```

\skip@lockoff
567
568
569 \newcommand{\do@lockoffR}{%
570 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
571 \ifsublines@
572 \xright@appenditem{-1006}\to\actions@listR
573 \ifnum\sub@lockR=\tw@
574 \sub@lockR \thr@@
575 \else
576 \sub@lockR \z@
577 \fi
578 \else
579 \xright@appenditem{-1004}\to\actions@listR
580 \ifnum\@lockR=\tw@
581 \@lockR \thr@@
582 \else
583 \@lockR \z@
584 \fi
585 \fi}
586
587
588 %

```

`\n@num`

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

`\@ref@regR`

`\insert@countR`

- #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`.

```

589 \newcount\insert@countR
590 %

```

- #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`.

`\@ref` itself is defined in `reledmac`. It calls `\ref@reg` or `\ref@regR`, depending whether we are in left or right side. Here, we define only `\ref@regR`, `\ref@reg` is already defined in `reledmac`.

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

```

591 \newcommand*{\@ref@regR}[2]{%
592   \global\advance\@edtext@level by 1%
593   \global\insert@countR=#1\relax
594   \loop\ifnum\insert@countR>\z@
595     \xright@appenditem{\the\absline@numR}\to\insertlines@listR
596     \global\advance\insert@countR \m@ne
597   \repeat
598 %

```

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.

```

599 \begingroup
600   \let\@ref=\dummy@ref
601   \let\@lopR\@gobble
602   \let\page@action=\relax
603   \let\sub@action=\relax
604   \let\set@line@action=\relax
605   \let\@lab=\relax
606   \let\@lemma=\relax
607   \let\@sw\@gobblethree%
608   #2
609   \global\endpage@num=\page@numR
610   \global\endline@num=\line@numR
611   \global\endsubline@num=\subline@numR
612 \endgroup
613 %

```

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

```

614 \xright@appenditem%
615   {\the\page@numR|\the\line@numR}%
616   \ifsublines@ \the\subline@numR \else 0\fi}%
617   \the\endpage@num|\the\endline@num}%
618   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@listR
619 %

```

Create a list which will store all the second argument of each `\@sw` in this lemma, at this level.

```

620   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
@edtext@level\endcsname}%
621 %

```

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

```

622   \@ref@reg@parseR{#2}%
623 %

```

Decrease edtext level counter.

```

624     \global\advance\@edtext@level by -1%
625 }
626 %

```

```

\@ref@reg@parseR27 \newcommand{\@ref@reg@parseR}[1]{%
628 %

```

Declare and init boolean for lemma in this level.

```

629     \providebool{lemmacommand@the\@edtext@level}%
630     \boolfalse{lemmacommand@the\@edtext@level}%
631 %

```

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

```

632     #1%
633 % Now, we store the list of \protect\cs{@sw} of this current \protect\cs{
edtext} as an element of
634 % the global list of list of \protect\cs{@sw} for a \protect\cs{edtext}
depth.
635 % \begin{macrocode}
636     \ifnum\@edtext@level>0%
637         \def\create@this@edtext@level{\expandafter\list@create\expandafter{
csname sw@list@edtextR@the\@edtext@level\endcsname}}%
638         \ifcsundef{sw@list@edtextR@the\@edtext@level}{\
create@this@edtext@level}{}%
639         \letcs{\@tmp}{sw@list@edtextR@the\@edtext@level}%
640         \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}%
641         \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
642         \global\cslet{sw@list@edtextR@the\@edtext@level}{\@tmp}%
643     \fi%
644 }%
645 %

```

**\@pend**    \@pend{<num>} adds its argument to the \linesinpar@listL list, and analogously  
**\@pendR**    for \@pendR. If needed, it resets line number. Both are defined in reledmac, but they  
are empty. They are really defined only in reledpar.

```

646 \renewcommand*\@pend}[1]{%
647     \ifbypstart@global\line@num=0\fi%
648     \xright@appenditem{#1}\to\linesinpar@listL}
649 \renewcommand*\@pendR}[1]{%
650     \ifbypstart@Rglobal\line@numR=0\fi
651     \xright@appenditem{#1}\to\linesinpar@listR}
652 %
653 %

```

`\@pstart` `\@pstart` and `cs@pstartR` allows us to know, when using `\nomaxlines` option in which page we should start a `pstart`, and also how many empty lines we should let before starting this `pstart` at the beginning of the page

```

654 \newcommand{\@pstart}[3]{%
655   \ifcsdef{minpage@pstart@#1}%
656     {\ifnumgreater{#2}{\csuse{minpage@pstart@#1}}%
657       {\csnumgdef{minpage@pstart@#1}{#2}}%
658     }%
659   }%
660   {\csnumgdef{minpage@pstart@#1}{#2}}
661   \csnumgdef{afterlines@pstart@#1L}{#3}%
662 }%
663
664 \newcommand{\@pstartR}[3]{%
665   \numdef{\@tmp}{#2-1}%Because we have not to know in which page the pstart
666   starts, but in which pair of facing page
667   \ifcsdef{minpage@pstart@#1}%
668     {\ifnumgreater{\@tmp}{\csuse{minpage@pstart@#1}}%
669       {\csnumgdef{minpage@pstart@#1}{\@tmp}}%
670     }%
671     {\csnumgdef{minpage@pstart@#1}{\@tmp}}
672   \csnumgdef{afterlines@pstart@#1R}{#3}%
673 }%
674 %

```

`\@lopL` `\@lopL{<num>}` adds its argument to the `\linesonpage@listL` list, and analogously for `\@lopR`. Both are defined in `reledmac`, but they are empty. They are really defined only in `reledpar`.

```

675 \renewcommand*{\@lopL}[1]{%
676   \xright@appenditem{#1}\to\linesonpage@listL}
677 \renewcommand*{\@lopR}[1]{%
678   \xright@appenditem{#1}\to\linesonpage@listR}
679
680 %

```

## IV.10 Writing to the line-list file

We have now defined all the counters, lists, and commands involved in reading the line-list file at the start of a section. Now we will cover the commands that `reledmac` uses within the text of a section to write commands out to the line-list.

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

```

681 \newwrite\linenum@outR
682 %

```

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

```

683 \newif\iffirst@linenum@out@R
684 \first@linenum@out@Rtrue
685 %

```

`\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.

```

686 \newcommand*{\line@list@stuffR}[1]{%
687   \read@linelist{#1}%
688   \iffirst@linenum@out@R
689     \immediate\closeout\linenum@outR
690     \global\first@linenum@out@Rfalse
691     \immediate\openout\linenum@outR=\l@auxdir#1%
692     \immediate\write\linenum@outR{\string\line@list@version{\
this@line@list@version}}}%
693     \ifl@dpageing%
694       \immediate\write\linenum@outR{\string\@par@sync@option{\
@par@this@sync@option}}}%
695     \fi%
696   \else
697     \if@minipage%
698       \leavevmode%
699     \fi%
700     \closeout\linenum@outR%
701     \openout\linenum@outR=\l@auxdir#1%
702   \fi}
703
704 %

```

`\new@lineL` The `\new@lineL` macro sends the `\@nl` command to the left text line-list file, to mark the start of a new text line.

```

705 \newcommand*{\new@lineL}{%
706   \write\linenum@out{\string\@nl[\the\c@page][\thepage]}}
707 %

```

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

```

708 \newcommand*{\new@lineR}{%
709   \write\linenum@outR{\string\@nl[\the\c@page][\thepage]}}
710 %

```

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

<code>\startsub</code> <code>\endsub</code>	<code>\startsub</code> and <code>\endsub</code> turn sub-lineation on and off, by writing appropriate instructions to the line-list file. There are both defined in <code>reledmac</code> .
<code>\advanceline</code>	You can use <code>\advanceline{⟨num⟩}</code> in running text to advance the current visible line-number by a specified value, positive or negative. It is defined in <code>reledmac</code> .
<code>\setline</code>	You can use <code>\setline{⟨num⟩}</code> in running text (i.e., within <code>\pstart... \pend</code> ) to set the current visible line-number to a specified positive value. It is defined in <code>reledmac</code> .
<code>\setlinenum</code>	You can use <code>\setlinenum{⟨num⟩}</code> before a <code>\pstart</code> to set the visible line-number to a specified positive value. It writes a <code>\l@d@set</code> command to the line-list file. It is defined in <code>reledmac</code> .
<code>\startlock</code> <code>\endlock</code>	You can use <code>\startlock</code> or <code>\endlock</code> 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. They are defined in <code>reledmac</code> .
<code>\skipnumbering</code>	

## V Marking text for notes

The `\edtext` macro is used to create all footnotes and endnotes, as well as to print the portion of the main text to which a given note or notes is keyed. The idea is to have that lemma appear only once in the `.tex` file: all instances of it in the main text and in the notes are copied from that one appearance.

<code>\critext</code> <code>\edtext</code> <code>\set@line</code>	The <code>\set@line</code> macro is called by <code>\edtext</code> to put the line-reference field and font specifier for the current block of text into <code>\l@d@nums</code> . It is defined in <code>reledmac</code> .
---	--

### V.1 Specific hooks and commands for notes

The `reledmac \newseries@` initializes commands which are linked to notes series. However, to keep `reledmac` as light as possible, it does not define commands which are specific to `reledpar`. This is what does `\newseries@par`. The specific hooks are also defined here.

<code>\newseries@par</code> <sup>11</sup>	<code>\newcommand{\newseries@par}[1]{%</code> <code>712 %</code>
---	---

### V.1.1 Notes to be printed on one side only

`reledpar` allows notes to be printed on one side only. We need to declare these options. We also need `box` to store temporarily the footnote not printed. We check the `nofamiliar` and `nocritical` `reledmac` options.

```

713 \unless\ifnofamiliar%
714 \csgdef{onlysideX@#1}{}%
715 \newnamebox{footins#1@kept}%
716 \fi%
717 \unless\ifnocritical%
718 \csgdef{Xonlyside@#1}{}%
719 \newnamebox{#1footins@kept}%
720 \fi%
721 %

```

### V.1.2 Tools specific to familiar footnotes

```

722 \unless\ifnofamiliar%
723 %

```

**Managing correct number** One problem with using familiar footnotes in parallel typesetting is the fact that the order of reading notes is not the same as the order they are typeset, because  $\text{\TeX}$  reads first all the notes on one side, then all the notes on the other side. Then, however,  $\text{\TeX}$  alternates between typesetting left-side note and right-side notes. Consequently, if we do nothing special, the note numbers are sorted in the reading order, not in the typesetting order. So we could obtain something like 1,3,2,5,4.

To prevent this problem, we use a two new counters by series. Every note, in parallel typesetting, has three associated counters.

1. A  $\text{\TeX}$  counter `footnoteX`. This the only one manipulated by user, and the only one finally printed.
2. A  $\text{\TeX}$  counter `footnoteX@reading`. Its value is incremented when reading the `\footnoteX` command in left or right side environments. It is used to get the correct footnote number from the `.aux` file to be typeset in the main text. This counter is already defined in `reledmac`, as it is also used for hyperlink.
3. A  $\text{\TeX}$  counter `footnoteX@typeset`. Its value is increased when inserting footnotes. Its value is used in the `.aux` files to be used on the next run for the main text.

So here, we only defined the new counter.

```

724 \newcounter{footnote#1@typeset}%
725 %

```

**Familiar footnotes without marks** The `\footnoteXnomk` commands are for notes which are printed on the left side, while they are called in the right side. Basically, they set first toggle `\nomark@` to true, then call the `\footnoteX`. and finally add the footnote counter in the footnote counter list.

First, check the `nofamiliar` option of `reledmac`.

So declare the list.

```
726 \expandafter\list@create\csname footnote#1@mk\endcsname%
727 %
```

Then, declare the `\footnoteXnomk` command.

```
728 \expandafter\newcommand\csname footnote#1nomk\endcsname[1]{%
729 %
```

First step: just call the normal `\footnoteX`, saying that we do not want to print the mark.

```
730 \toggletrue{nomk@}%
731 \csuse{footnote#1}{##1}%
732 \togglefalse{nomk@}%
733 %
```

Second, and last, step: store the footnote counter in the footnote counters list. We use some `\let`, because `\xright@appenditem` is difficult to use with `\expandafter`.

```
734 \letcs{\@tmp}{footnote#1@mk}%
735 \numdef\@tmpa{\csuse{c@footnote#1}}%
736 \global\xright@appenditem{\@tmpa}\to\@tmp%
737 \global\cslet{footnote#1@mk}{\@tmp}%
738 }%
739 %
```

Then, declare the command which inserts the footnotemark in the right side.

```
740 \expandafter\newcommand\csname footnote#1mk\endcsname{%
741 %
```

Get the first element of the footnote mark list. As `\gl@p` is difficult to use with dynamic name macro, we use `\let` commands.

```
742 \letcs{\@tmp}{footnote#1@mk}%
743 \gl@p\@tmp\to\@tmpa%
744 \global\cslet{footnote#1@mk}{\@tmp}%
745 %
```

Set the `footnotecounter` with it. For the sake of security, we make a backup of the previous value.

```
746 \letcs{\old@footnote}{c@footnote#1}%
747 \setcounter{footnote#1}{\@tmpa}%
748 %
```

Define the footnote mark and print it



```

749 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
750 \csuse{@footnotemark#1}%
751 %

```

Restore previous footnote counter and finally add space.

```

752 \setcounter{footnote#1}{\old@footnote}%
753 \xspace%
754 }%
755 %

```

End of tools specific to familiar notes.

```

756 \fi
757 %

```

End of `\newseries@par`.

```

758 }%
759 %

```

### V.1.3 Get correct footnote number

As users can insert footnotes between two `\Pairs` or `\Pages` commands, we have to set the `\+footnoteX@typeset+` counter to the last value of the `footnoteX` counter at the beginning of these two commands.

```

760 \newcommand{\save@familiarfootnote@number}{%
761 \unless\ifnofamiliar%
762 \def\do##1{\csxdef{saved@footnote##1}{\the\csname c@footnote##1\
endcsname}}%
763 \dolistloop{\@series}%
764 \fi%
765 \xdef\saved@footnote{\the\c@footnote}%
766 }
767 \newcommand{\get@familiarfootnote@number}{%
768 \unless\ifnofamiliar%
769 \def\do##1{\setcounter{footnote##1@typeset}{\csuse{saved@footnote##1}}}
770 %
771 \dolistloop{\@series}%
772 \fi%
773 \setcounter{footnote@typeset}{\saved@footnote}%
774 }
775 %

```

## V.2 Create hooks

Read the `reledmac` code handbook about `\newhookcommand@series`. Here, we create hooks which are specific to `reledpar`.

```

775 \unless\ifnocritical@%
776 \newhookcommand@series{Xonlyside}%
777 \fi%
778 \unless\ifnofamiliar@%
779 \newhookcommand@series{onlysideX}%
780 \fi
781
782
783 %

```

### V.3 Init standards series (A,B,C,D,E,Z)

`\init@series@par` `\newseries@par` is called by `\newseries`. However, this last command is called before `reledpar` is loaded. Thus, we need to initiate a specific series hook for `reledpar`.

```

784 \newcommand{\init@series@par}{%
785 \def\do##1{\newseries@par{##1}}%
786 \dolistloop{\@series}%
787 }%
788 \init@series@par%
789 %

```

### V.4 Tools specific to $\text{\LaTeX}$ 's classical footnotes

As users can use classical footnotes of  $\text{\LaTeX}$  (`\footnote`) in parallel texts, we must integrate the same tools to get correct number as for `reledmac`' footnotes (V.1.2 p. 47).

```

\footnote@reading90 \newcount\footnote@reading%
\footnote@typeset91 \newcounter{footnote@typeset}%
92 %

```

## VI *Pstart numbers dumping and restoration*

While in `reledmac` the footnotes are inserted at the same time as the `\pstart... \pend` are read, in `reledpar` they are inserted when the `\Columns` or `\Pages` commands are called. Consequently, if we do nothing, the value of the `PstartL` and `PstartR` counters are not the same in the main text and in the notes. To solve this problem, we dump the values in two list (one by side) when processing `\pstart` and restore these at each `\pstart` when calling `\Columns` or `\Pages`. We also dump and restore the value of the boolean `\ifnumberpstart`.

So, first step, creating the lists. Here, “pc” means “public counters”.

```

\list@pstartL@pc93 \list@create{\list@pstartL@pc}%
\list@pstartR@pc94 \list@create{\list@pstartR@pc}%
95 %

```

Two commands to dump current pstarts. We prefer two commands to one with argument indicating the side, because the commands are short, and so we save one test (or a \csname construction).

```

\dump@pstartL@pc% \def\dump@pstartL@pc{%
\dump@pstartR@pc% \xright@appenditem{\the\c@pstartL}\to\list@pstartL@pc%
798 \global\cslet{numberpstart@L\the\l@dumpstartsL}{\ifnumberpstart}%
799 }%
800
801 \def\dump@pstartR@pc{%
802 \xright@appenditem{\the\c@pstartR}\to\list@pstartR@pc%
803 \global\cslet{numberpstart@R\the\l@dumpstartsR}{\ifnumberpstart}%
804 }%
805
806 %

```

\restore@pstartL@pc And so, the commands to restore them.

```

\restore@pstartR@pc \def\restore@pstartL@pc{%
807 \ifx\list@pstartL@pc\empty\else%
808 \gl@p\list@pstartL@pc\to\@temp%
809 \global\c@pstartL=\@temp%
810 \fi%
811 }%
812
813 \def\restore@pstartR@pc{%
814 \ifx\list@pstartR@pc\empty\else%
815 \gl@p\list@pstartR@pc\to\@temp%
816 \global\c@pstartR=\@temp%
817 \fi%
818 }%
819 %

```

## VII Parallel environments

The initial set up for parallel processing is deceptively simple.

pairs pages

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

```

820 \newenvironment{pairs}{%}
821 \l@dpairingtrue
822 \l@dpagingfalse
823 \initnumbering@quote
824 \save@familiarfootnote@number%
825 \if@ledgroup%
826 \get@familiarfootnote@number%
827 \fi%

```

```

828 \save@section@number%
829 \at@begin@pairs%
830 }{%
831 \l@dpairingfalse
832 }
833
834 %

```

**\AtBeginPairs** The `\AtBeginPairs` macro just define a `\at@begin@pairs` macro, called at the beginning of each pairs environments.

```

835 \newcommand{\AtBeginPairs}[1]{\gdef\at@begin@pairs{#1}}%
836 \def\at@begin@pairs{}%
837
838 %

```

The `pages` environment additionally sets the ‘column’ widths to the `\textwidth` (as known at the time the package is called). In this environment, there are two text in parallel on 2 pages.

```

839 \newenvironment{pages}{%
840 \l@dpairingtrue
841 \l@dpagingtrue
842 \initnumbering@quote
843 \save@familiarfootnote@number%
844 \if@ledgroup%
845 \get@familiarfootnote@number%
846 \fi%
847 \save@section@number%
848 \setlength{\Lcolwidth}{\textwidth}%
849 \setlength{\Rcolwidth}{\textwidth}%
850 }{%
851 \l@dpairingfalse
852 \l@dpagingfalse
853 }
854
855 %

```

**ifinstanzaL** These boolean tests are switched by the `\stanza` command, using either the left or right side.

**ifinstanzaR**

```

856 \newif\ifinstanzaL
857 \newif\ifinstanzaR
858 %

```

**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.

```

859 \newenvironment{Leftside}{%
860   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname\else%
861     \led@err@Leftside@PreviousNotPrinted%
862   \fi%
863   \ledRcolfalse
864   \setcounter{pstartL}{1}
865   \let\pstart\pstartL
866   \let\thepstart\thepstartL
867   \let\pend\pendL
868   \let\memorydump\memorydumpL
869   \Leftsidehook
870   \let\old@startstanza\@startstanza
871   \def\@startstanza[##1]{\global\instanzaLtrue\old@startstanza[##1]}
872 }{
873   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname%
874     \led@error@missing@numbering{Leftside}%
875   \fi%
876   \Leftsidehookend}
877 %

```

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

```

\Leftsidehookend
\Rightsidehook
\Rightsidehookend
878 \newcommand*{\Leftsidehook}{}
879 \newcommand*{\Leftsidehookend}{}
880 \newcommand*{\Rightsidehook}{}
881 \newcommand*{\Rightsidehookend}{}
882
883 %

```

**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.

```

884 \newenvironment{Rightside}{%
885   \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else%
886     \led@err@Rightside@PreviousNotPrinted%
887   \fi%
888   \ledRcoltrue
889   \let\beginnumbering\beginnumberingR
890   \let\endnumbering\endnumberingR
891   \let\pausenumbering\pausenumberingR
892   \let\resumenumbering\resumenumberingR
893   \let\memorydump\memorydumpR
894   \let\thepstart\thepstartR
895   \let\pstart\pstartR
896   \let\pend\pendR
897   \let\ledpb\ledpbR
898   \let\lednopb\lednopbR
899   \let\lineation\lineationR

```

```

900 \Rightsidehook
901 \let\old@startstanza\@startstanza
902 \def\@startstanza[##1]{\global\instanzaRtrue\old@startstanza[##1]}
903 }{f%
904 \ledRcolfalse
905 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
906 \led@error@missing@numbering{Rightside}%
907 \fi%
908 \Rightsidehookend
909 }
910
911 %

```

## VIII 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.

### VIII.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**  
**\par@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 is 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.

```

912 \newcount\num@linesR
913 \newbox\one@lineR
914 \newcount\par@lineR
915 %

```

**\pstartL** **\pstartR** `\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 is 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.

```

916 \newcounter{pstartL}
917 \renewcommand{\thepstartL}{\bfseries\@arabic\c@pstartL}. }
918 \newcounter{pstartR}
919 \renewcommand{\thepstartR}{\bfseries\@arabic\c@pstartR}. }
920
921
922 \newcommandx*{\pstartL}[1][1]{%
923   \if@nobreak%
924     \let\@oldnobreak\@nobreaktrue%
925   \else%
926     \let\@oldnobreak\@nobreakfalse%
927   \fi%
928   \@nobreaktrue%
929   \ifluatex%
930     \xdef\l@luatextextdir@L{\the\textdir}%
931     \xdef\l@luatexpardir@L{\the\pardir}%
932     \xdef\l@luatexbodydir@L{\the\bodydir}%
933   \fi%
934   \ifnumbering \else%
935     \led@err@PstartNotNumbered%
936     \beginnumbering%
937   \fi%
938   \ifnumberedpar@%
939     \led@err@PstartInPstart%
940   \pend%
941   \fi%
942 %

```

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

```

943 \ifpst@rtedL\else%
944   \list@clear{\inserts@list}%
945   \global\let\next@insert=\empty%
946   \global\pst@rtedLtrue%
947 \fi%
948 \begingroup\everypar{}%
949 %

```

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

```

950 \global\advance\l@dnumpstartsL \@ne%
951 \ifnum\l@dnumpstartsL>\l@dc@maxchunks%
952   \led@err@TooManyPstarts%
953   \global\l@dnumpstartsL=\l@dc@maxchunks%
954 \fi%
955 \global\setnamebox{\l@dLcolrawbox\the\l@dnumpstartsL}=\vbox\bgroup%
956 %

```

We set all the usual interline penalties to zero; this ensures that there will be no large interline penalties to prevent us from slicing the paragraph into pieces. These penalties

revert to the values that you set when the group for the `\vbox` ends.

```

957 \l@dzeroopenalties%
958 \ifautopar\else%
959   \ifnumberpstart%
960     \ifsidepstartnum%
961       \else%
962         \thepstartL%
963       \fi%
964     \fi%
965   \fi%
966 \hsize=\Lcolwidth%
967 \numberedpar@true%
968 \iflabelpstart\protected@edef\@currentlabel%
969   {\p@pstartL\thepstartL}\fi%
970 %

```

Dump the optional arguments

```

971 \ifstrempy{#1}%
972   {\csgdef{before@pstartL@the\l@dnumpstartsL}{\at@every@pstart}}%
973   {\csgdef{before@pstartL@the\l@dnumpstartsL}{\noindent#1}}%
974   \at@every@pstart@call%
975 %

```

Gobble following space (automatically done if there is no optional argument)

```

976 \ignorespaces%
977 %
978 }
979 %

```

The same for right side.

```

980 \newcommandx*{\pstartR}[1][1]{%
981   \if@nobreak%
982     \let\@oldnobreak\@nobreaktrue%
983   \else%
984     \let\@oldnobreak\@nobreakfalse%
985   \fi%
986   \@nobreaktrue%
987   \ifluatex%
988     \xdef\l@luatextextdir@R{\the\textdir}%
989     \xdef\l@luatexpardir@R{\the\pardir}%
990     \xdef\l@luatexbodydir@R{\the\bodydir}%
991   \fi%
992   \ifnumberingR \else%
993     \led@err@PstartNotNumbered%
994     \beginnumberingR%
995   \fi%
996   \ifnumberedpar@%
997     \led@err@PstartInPstart%

```



```

998 \pendR%
999 \fi%
1000 \ifpstart@rtedR\else%
1001 \list@clear{\inserts@listR}%
1002 \global\let\next@insertR=\empty%
1003 \global\pstart@rtedRtrue%
1004 \fi%
1005 \begingroup\everypar{}%
1006 \global\advance\l@dnumpstartsR \@ne%
1007 \ifnum\l@dnumpstartsR>\l@dc@maxchunks%
1008 \led@err@TooManyPstarts%
1009 \global\l@dnumpstartsR=\l@dc@maxchunks%
1010 \fi%
1011 \global\setnamebox{\l@dRcolrawbox\the\l@dnumpstartsR}=\vbox\bgroup%
1012 \l@dzeropenalties%
1013 \ifautopar\else%
1014 \ifnumberpstart%
1015 \ifsidepstartnum\else%
1016 \thepstartR%
1017 \fi%
1018 \fi%
1019 \fi%
1020 \hsize=\Rcolwidth%
1021 \numberedpar@true%
1022 \iflabelpstart\protected@edef\@currentlabel%
1023 {\p@pstartR\thepstartR}\fi%
1024 \ifstrempy{#1}%
1025 {\csgdef{before@pstartR@the\l@dnumpstartsR}{\at@every@pstart}}%
1026 {\csgdef{before@pstartR@the\l@dnumpstartsR}{\noindent#1}}%
1027 \at@every@pstart@call%
1028 \ignorespaces%
1029 }
1030 %

```

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

```

1031 \newcommandx*\pendL}[1][1]{%
1032 \ifnumbering \else%
1033 \led@err@PendNotNumbered%
1034 \fi%
1035 \ifnumberedpar@ \else%
1036 \led@err@PendNoPstart%
1037 \fi%
1038 %

```

We immediately call \endgraf to end the paragraph; this ensures that there will be no large interline penalties to prevent us from slicing the paragraph into pieces.

```

1039 \endgraf\global\num@lines=\prevgraf\egroup%
1040 \global\par@line=0%

```

```

1041 %
End the group that was begun in the \pstart.

1042 \endgroup%
1043 \ignorespaces%
1044 \@oldnobreak%
1045 \dump@pstartL@pc%
1046 \ifnumberpstart%
1047   \addtocounter{pstartL}{1}%
1048 \fi
1049 \parledgroup@beforenotes@save{L}%
1050 %

Dump content of the optional argument.

1051 \ifstrempy{#1}%
1052   {\csgdef{after@pendL@the\l@dnumpstartsL}{\at@every@pend}}%
1053   {\csgdef{after@pendL@the\l@dnumpstartsL}{\noindent#1}}%
1054 }
1055 %

```

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

```

1056 \newcommandx*{\pendR}[1][1]{%
1057   \ifnumberingR \else%
1058     \led@err@PendNotNumbered%
1059   \fi%
1060   \ifnumberedpar@ \else%
1061     \led@err@PendNoPstart%
1062   \fi%
1063   \endgraf\global\l@num@linesR=\prevgraf\egroup%
1064   \global\par@lineR=0%
1065   \endgroup%
1066   \ignorespaces%
1067   \@oldnobreak%
1068   \dump@pstartR@pc%
1069   \ifnumberpstart%
1070     \addtocounter{pstartR}{1}%
1071   \fi%
1072   \parledgroup@beforenotes@save{R}%
1073   \ifstrempy{#1}%
1074     {\csgdef{after@pendR@the\l@dnumpstartsR}{\at@every@pend}}%
1075     {\csgdef{after@pendR@the\l@dnumpstartsR}{\noindent#1}}%
1076 }
1077 %
1078 %

```

**\AtEveryPstartCall** The \AtEveryPstartCall argument is called when the \pstartL or \pstartR is called. That is different of \AtEveryPstart the argument of which is called when the \pstarts are printed.

```

1079 \newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
1080 \gdef\at@every@pstart@call{}%
1081 %

```

`\ifprint@last@after@pendL` Two booleans set to true, when the time is to print the last optional argument of a `\pend`.

```

\ifprint@last@after@pendR
1082 \newif\ifprint@last@after@pendL%
1083 \newif\ifprint@last@after@pendR%
1084 %

```

## VIII.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 analogously for a line of right text.

```

\l@drightbox
1085 \newbox\l@dleftbox
1086 \newbox\l@drightbox
1087
1088 %

```

`\countLline` We need to know the number of lines processed.

```

\countRline
1089 \newcount\countLline
1090 \countLline \z@
1091 \newcount\countRline
1092 \countRline \z@
1093
1094 %

```

`\@donereallinesL` We need to know the number of ‘real’ lines output (i.e., those that have been input by the user), and the total lines output (which includes any blank lines output for synchronisation).

```

\@donetotallinesL
\@donereallinesR
\@donetotallinesR
1095 \newcount\@donereallinesL
1096 \newcount\@donetotallinesL
1097 \newcount\@donereallinesR
1098 \newcount\@donetotallinesR
1099
1100 %

```

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

```

1101 \newcommand*{\do@lineL}{%
1102 \letcs{\ifnumberpstart}{numberpstart@L\the\l@dpscl}%
1103 \advance\countLline \@ne%

```

```

1104 \ifvbox\namebox{l@dLcolrawbox\the\l@dpscL}%
1105 {\vbadness=10000%
1106 \splittopskip=\z@%
1107 \do@lineLhook%
1108 \l@emptyd@ta%
1109 \global\setbox\one@line=\vsplit\namebox{l@dLcolrawbox\the\l@dpscL}%
1110 to\baselineskip}%
1111 \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startL}{}%
1112 \unvbox\one@line \global\setbox\one@line=\lastbox%
1113 \@writepageofparL%
1114 \getline@numL%
1115 \ifnum\@lock>\@ne%
1116 \inserthangingsymboltrue%
1117 \else%
1118 \inserthangingsymbolfalse%
1119 \fi%
1120 \setbox\l@dleftbox%
1121 \hb@xt@ \Lcolwidth{%
1122 \ifl@dhidenumber%
1123 \global\l@dhidenumberfalse%
1124 \f@x@l@cks%
1125 \else%
1126 \affixline@num%
1127 \fi%
1128 \xifinlist{\the\l@dpscL}{\eled@sections@@}%
1129 {%
1130 \if@firstlineofpage%
1131 \set@Xtxtbeforenotes%
1132 \global\@firstlineofpagefalse%
1133 \fi%
1134 \insert@msdata%
1135 \add@inserts%
1136 \affixside@note%
1137 }%
1138 {\print@lineL}%
1139 }%
1140 \add@penaltiesL%
1141 \global\advance\@donereallinesL\@ne%
1142 \global\advance\@donetotallinesL\@ne%
1143 \else%
1144 \iflinenumberLevenifblank
1145 \new@lineL%
1146 \l@emptyd@ta%
1147 \getline@numL%
1148 \affixline@num%
1149 \setbox\l@dleftbox \hb@xt@ \Lcolwidth{%
1150 \l@dld@ta%
1151 \hspace*{\Lcolwidth}%
1152 \ledrlfill\l@drd@ta%

```

```

1153 }%
1154 \else%
1155   \setbox\l@leftbox \hb@xt@ \Lcolwidth{\hspace*{\Lcolwidth}}%
1156   \fi%
1157   \global\advance\@donetotallinesL\@ne%
1158 \fi%
1159 }%
1160
1161
1162 %

```

`\print@lineL` `\print@lineL` is for lines without a sectioning command. See `reledmac` definition of `\print@line` for handbook.

```

1163 \def\print@lineL{%
1164   \affixstart@numL%
1165   \l@dld@ta%
1166   \if@firstlineofpage%
1167     \set@Xtxtbeforenotes%
1168     \global\@firstlineofpagefalse%
1169   \fi%
1170   \insert@msdata%
1171   \add@inserts\affixside@note%
1172   \l@dlsn@te%
1173   \hb@xt@ \Lcolwidth{\ledllfill\hb@xt@ \wd\one@line{%
1174     \do@insidelineLhook%
1175     \ifluatex%
1176       \textdir\l@luatextextdir@L%
1177     \fi%
1178     \new@lineL%
1179     \inserthangingsymbolL%
1180     \l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta%
1181   \l@drrsn@te}}%
1182
1183 %

```

`\print@eledsectionL` `\print@eledsectionL` is for line with macro code.

```

1184 \def\print@eledsectionL{%%
1185   \addtocounter{pstartL}{-1}%
1186   \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{%
1187   \ifdefstring{\@eledsectmark}{L}{\ledsectnomark}%
1188   \numdef{\temp@}{\l@dpscL-1}%
1189   \xifinlist{\temp@}{\eled@sections@}{\@nobreaktrue}{\@nobreakfalse}%
1190   \@eled@sectioningtrue%
1191   \bgroup%
1192     \ifluatex%
1193       \textdir\l@luatextextdir@L%
1194       \pardir\l@luatexpardir@L%
1195       \bodydir\l@luatexbodydir@L%

```

```

1196 \ifdefstring{\l@luatexttextdir@L}{TRT}{\@RTLtrue}{}%
1197 \fi%
1198 \csuse{eled@sectioning@the\l@dpscL}%
1199 \egroup%
1200 \@eled@sectioningfalse%
1201 \global\csundef{eled@sectioning@the\l@dpscL}%
1202 \if@RTL%
1203 \hspace{-3\paperwidth}%
1204 {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1205 \else%
1206 \hspace{3\paperwidth}%
1207 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1208 \fi%
1209 \vskip\eledsection@correcting@skip%
1210 }
1211
1212 %

```

**\dolineLhook** These high-level commands just redefine the low-level commands. They have to be used  
**\dolineRhook** be user, without `\makeatletter`.

```

\doinssideLhook 1213 \newcommand*{\dolineLhook}[1]{\gdef\dolineLhook{#1}}%
\doinssideRhook 1214 \newcommand*{\dolineRhook}[1]{\gdef\dolineRhook{#1}}%
1215 \newcommand*{\doinssideLhook}[1]{\gdef\doinssideLhook{#1}}%
1216 \newcommand*{\doinssideRhook}[1]{\gdef\doinssideRhook{#1}}%
1217
1218 %

```

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

```

\dolineLhook 1219 \newcommand*{\do@lineLhook}{%
\dolineRhook 1220 \newcommand*{\do@lineRhook}{%
\do@insidelineLhook 1221 \newcommand*{\do@insidelineLhook}{%
\do@insidelineRhook 1222 \newcommand*{\do@insidelineRhook}{%
1223
1224 %

```

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

```

1225 \newcommand*{\do@lineR}{%
1226 \let\linenumrepL\linenumrep%
1227 \let\sublinenumrepL\sublinenumrep%
1228 \let\linenumrepR\linenumrepR%
1229 \let\sublinenumrepR\sublinenumrepR%
1230 \letcs{\ifnumberpstart}{numberpstart@R\the\l@dpscR}%
1231 \ledRcol@true%
1232 \advance\countRline \@ne%
1233 \ifvbox\namebox{\l@dRcolrawbox\the\l@dpscR}%
1234 {\vbadness=10000%

```

```

1235 \splittopskip=\z@%
1236 \do@lineRhook%
1237 \l@emptyd@ta%
1238 \global\setbox\one@lineR=\vsplit\namebox{l@dRcolrawbox\the\l@dpscR}%
1239 to\baselineskip}%
1240 \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startR}{}%
1241 \unvbox\one@lineR \global\setbox\one@lineR=\lastbox%
1242 \@writepageofparR%
1243 \getline@numR%
1244 \ifnum\@lockR>\@ne%
1245 \inserthangingsymbolRtrue%
1246 \else%
1247 \inserthangingsymbolRfalse%
1248 \fi%
1249 \setbox\l@drightbox%
1250 \hb@xt@ \Rcolwidth{%
1251 \ifl@dhiddenumber%
1252 \global\l@dhiddenumberfalse%
1253 \f@x@l@cksR%
1254 \else%
1255 \affixline@numR%
1256 \fi%
1257 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
1258 {%
1259 \if@firstlineofpageR%
1260 \set@Xtxtbeforenotes%
1261 \global\@firstlineofpageRfalse%
1262 \fi%
1263 \insert@msdata%
1264 \add@insertsR%
1265 \affixside@noteR%
1266 }%
1267 {\print@lineR}%
1268 }%
1269 \add@penaltiesR%
1270 \global\advance\@donereallinesR\@ne%
1271 \global\advance\@donetotallinesR\@ne%
1272 \else%
1273 \iflinenumberRevenifblank%
1274 \new@lineR
1275 \l@emptyd@ta%
1276 \getline@numR%
1277 \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1278 \affixline@numR%
1279 \l@dld@ta%
1280 \hspace*{\Rcolwidth}%
1281 \ledrlfill\l@drd@ta%
1282 }%
1283 \else%

```

```

1284 \setbox\l@drightbox \hb@xt@ \Rcolwidth{\hspace*{\Rcolwidth}}%
1285 \fi%
1286 \global\advance\@donetotallinesR\@ne%
1287 \fi%
1288 \ledRcol@false%
1289 \let\linenumrep\linenumrepL%
1290 \let\sublinenumrep\sublinenumrepL%
1291 }
1292
1293
1294 %

```

```

\print@lineR
\print@eledsectionR

```

### VIII.3 Line and page number computation

`\getline@numR` The `\getline@numR` macro determines the page and line numbers for the right text line we are about to send to the vertical list. The `\getline@numL` is the same for left text.

```

1295 \newcommand*\getline@numR{%
1296 \global\advance\absline@numR \@ne
1297 \do@actionsR
1298 \do@ballastR
1299 \ifledgroupnotesR\else
1300 \ifnumberline
1301 \ifsublines@
1302 \ifnum\sub@lockR<\tw@
1303 \global\advance\subline@numR \@ne
1304 \fi
1305 \else
1306 \ifnum\@lockR<\tw@
1307 \global\advance\line@numR \@ne
1308 \global\subline@numR \z@
1309 \fi
1310 \fi
1311 \fi
1312 }
1313 \newcommand*\getline@numL{%
1314 \global\advance\absline@num \@ne
1315 \do@actions
1316 \do@ballast
1317 \ifledgroupnotesL\else
1318 \ifnumberline
1319 \ifsublines@
1320 \ifnum\sub@lock<\tw@
1321 \global\advance\subline@num \@ne
1322 \fi
1323 \else
1324 \ifnum\@lock<\tw@
1325

```



```

1326         \global\advance\line@num \@ne
1327         \global\subline@num \z@
1328     \fi
1329 \fi
1330 \fi
1331 \fi
1332 }
1333
1334
1335 %

```

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

```

1336 \newcommand*{\do@ballastR}{\global\ballast@count=\z@
1337 \begingroup
1338   \advance\absline@numR \@ne
1339   \ifnum\next@actionlineR=\absline@numR
1340     \ifnum\next@actionR>-1001
1341       \global\advance\ballast@count by -\c@ballast
1342     \fi
1343   \fi
1344   \endgroup}
1345 %

```

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

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

```

1346 \newif\ifl@dskipversenumberR
1347 \newcommand*{\do@actions@fixedcodeR}{%
1348   \ifcase\l@dskipversenumberR
1349   \or% % 1001 = starting sublineation
1350     \global\sublines@true
1351   \or% % 1002 = ending sublineation
1352     \global\sublines@false
1353   \or% % 1003 = starting locking number
1354     \global\@lockR=\@ne
1355   \or% % 1004 = ending locking number
1356     \ifnum\@lockR=\tw@
1357       \global\@lockR=\thr@@
1358     \else
1359       \global\@lockR=\z@
1360     \fi
1361   \or% % 1005 = starting locking subnumber
1362     \global\sub@lockR=\@ne
1363   \or% % 1006 = ending locking subnumber
1364     \ifnum\sub@lockR=\tw@
1365

```

```

1366 \global\sub@lockR=\thr@@
1367 \else
1368 \global\sub@lockR=\z@
1369 \fi
1370 \or% % 1007 = skipping numbering
1371 \l@dskipnumbertrue
1372 \or% % 1008 = skipping numbering in stanza
1373 \l@dskipversenumberRtrue%
1374 \or% % 1009 = hiding number
1375 \l@dhiddenumbertrue%
1376 \or% % 1010 = inserting msdata
1377 \add@msdata%
1378 \else%
1379 \led@warn@BadAction
1380 \fi%
1381 }
1382
1383
1384 \newcommand*{\do@actionsR}{%
1385 \global\let\do@actions@nextR=\relax
1386 \@l@tempcntb=\absline@numR
1387 \ifnum\@l@tempcntb<\next@actionlineR\else
1388 \ifnum\next@actionR>-1001\relax
1389 \@firstlineofpageRtrue%
1390 \global\page@numR=\next@actionR
1391 \ifbypage@R
1392 \global\line@numR \z@ \global\subline@numR \z@
1393 \fi
1394 \add@msdata@firstlineofpage%
1395 \else
1396 \ifnum\next@actionR<-4999\relax % 9/05 added relax here
1397 \@l@tempcnta=-\next@actionR
1398 \advance\@l@tempcnta by -5001\relax
1399 \ifsublines@
1400 \global\subline@numR=\@l@tempcnta
1401 \else
1402 \global\line@numR=\@l@tempcnta
1403 \fi
1404 \else
1405 \@l@tempcnta=-\next@actionR
1406 \advance\@l@tempcnta by -1000\relax
1407 \do@actions@fixedcodeR
1408 \fi
1409 \fi
1410 \ifx\actionlines@listR\empty
1411 \gdef\next@actionlineR{1000000}%
1412 \else
1413 \glp\actionlines@listR\to\next@actionlineR
1414 \glp\actions@listR\to\next@actionR
1415 \global\let\do@actions@nextR=\do@actionsR

```

```

1416 \fi
1417 \fi
1418 \do@actions@nextR}
1419
1420 %

```

## VIII.4 Line number printing

`\l@dcalcnm` `\affixline@numR` is the right text version of the `\affixline@num` macro.

```

\ch@cksub@l@ckR
\ch@ck@l@ckR
\fx@l@ckR
\affixline@numR
1421 \newcommand*{\l@dcalcnm}[3]{%
1422 \ifnum #1 > #2\relax
1423 \l@l@tempcnta = #1\relax
1424 \advance\l@l@tempcnta by -#2\relax
1425 \divide\l@l@tempcnta by #3\relax
1426 \multiply\l@l@tempcnta by #3\relax
1427 \advance\l@l@tempcnta by #2\relax
1428 \else
1429 \l@l@tempcnta=#2\relax
1430 \fi}
1431
1432 \newcommand*{\ch@cksub@l@ckR}{%
1433 \ifcase\sub@lockR
1434 \or
1435 \ifnum\sublock@disp=\@ne
1436 \l@l@tempcntb \z@ \l@l@tempcnta \@ne
1437 \fi
1438 \or
1439 \ifnum\sublock@disp=\tw@
1440 \else
1441 \l@l@tempcntb \z@ \l@l@tempcnta \@ne
1442 \fi
1443 \or
1444 \ifnum\sublock@disp=\z@
1445 \l@l@tempcntb \z@ \l@l@tempcnta \@ne
1446 \fi
1447 \fi}
1448
1449 \newcommand*{\ch@ck@l@ckR}{%
1450 \ifcase\@lockR
1451 \or
1452 \ifnum\lock@disp=\@ne
1453 \l@l@tempcntb \z@ \l@l@tempcnta \@ne
1454 \fi
1455 \or
1456 \ifnum\lock@disp=\tw@
1457 \else
1458 \l@l@tempcntb \z@ \l@l@tempcnta \@ne
1459

```

```

1460 \fi
1461 \or
1462 \ifnum\lock@disp=\z@
1463 \l@dttempcntb \z@ \l@dttempcnta \@ne
1464 \fi
1465 \fi}
1466
1467 \newcommand*{\f@x@l@cksR}{%
1468 \ifcase\@lockR
1469 \or
1470 \global\@lockR \tw@
1471 \or \or
1472 \global\@lockR \z@
1473 \fi
1474 \ifcase\sub@lockR
1475 \or
1476 \global\sub@lockR \tw@
1477 \or \or
1478 \global\sub@lockR \z@
1479 \fi}
1480
1481
1482 \newcommand*{\affixline@numR}{%
1483 \ifledgroupnotesR\else\ifnumberline
1484 \ifl@dskipnumber
1485 \global\l@dskipnumberfalse
1486 \else
1487 \ifsublines@
1488 \l@dttempcntb=\subline@numR
1489 \l@dcalcnum{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR
}%
1490 \ch@cksub@lockR
1491 \else
1492 \l@dttempcntb=\line@numR
1493 \ifx\linenumberlist\empty
1494 \l@dcalcnum{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%
1495 \else
1496 \l@dttempcnta=\line@numR
1497 \edef\rem@inder{\linenumberlist,\number\line@numR,}%
1498 \edef\sc@n@list{\def\noexpand\sc@n@list
1499 ###1,\number\l@dttempcnta,###2|{\def\noexpand\rem@inder{###2}}}%
1500 \sc@n@list\expandafter\sc@n@list\rem@inder|}%
1501 \ifx\rem@inder\empty\advance\l@dttempcnta\@ne\fi
1502 \fi
1503 \ch@ck@l@ckR
1504 \fi
1505 \ifnum\l@dttempcnta=\l@dttempcntb
1506 \ifl@dskipversenumberR\else
1507 \if@twocolumn
1508 \if@firstcolumn

```

```

1509     \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1510     \else
1511     \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1512     \fi
1513   \else
1514     \@l@tempcntb=\line@marginR
1515     \ifnum\@l@tempcntb>\@ne
1516       \advance\@l@tempcntb by\page@numR
1517     \fi
1518     \ifodd\@l@tempcntb
1519       \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1520     \else
1521       \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1522     \fi
1523   \fi
1524 \fi
1525 \fi
1526 \f@x@l@cksR
1527 \fi
1528 \fi
1529 \fi}
1530 %

```

## VIII.5 Pstart number printing in side

The printing of the pstart number is like in reledmac, with two differences :

- Some commands have versions suffixed by R or L.
- The `\affixpstart@num` and `\affixpstart@numR` commands are called in the `\Pages` command. Consequently, the `pstartL` and `pstartR` counters must be reset at the beginning of this command.

```

\affixpstart@numL31
\affixpstart@numR32 \newcommand*{\affixpstart@numL}{%
  \leftpstartnumR33 \ifsidepstartnum
  \rightpstartnumR34 \if@twocolumn
    \if@firstcolumn
    \leftpstartnumL35
    \gdef\l@dld@ta{\llap{\leftpstartnumL}}}%
  \rightpstartnumL36
  \else
  \ifpstartnumR37
  \gdef\l@drd@ta{\rlap{\rightpstartnumL}}}%
1538
1539 \fi
1540 \else
1541   \@l@tempcntb=\line@margin
1542   \ifnum\@l@tempcntb>\@ne
1543     \advance\@l@tempcntb \page@num
1544   \fi
1545   \ifodd\@l@tempcntb
1546     \gdef\l@drd@ta{\rlap{\rightpstartnumL}}}%

```

```

1547     \else
1548         \gdef\l@dld@ta{\llap{\leftpstartnumL}}}%
1549     \fi
1550 \fi
1551 \fi
1552 }
1553 \newcommand*\affixpstart@numR{%
1554 \ifsidepstartnum
1555 \if@twocolumn
1556     \if@firstcolumn
1557         \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
1558     \else
1559         \gdef\l@dld@ta{\rlap{\rightpstartnumR}}}%
1560     \fi
1561 \else
1562     \@l@tempcntb=\line@marginR
1563     \ifnum\@l@tempcntb>\@ne
1564         \advance\@l@tempcntb \page@numR
1565     \fi
1566     \ifodd\@l@tempcntb
1567         \gdef\l@dld@ta{\rlap{\rightpstartnumR}}}%
1568     \else
1569         \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
1570     \fi
1571 \fi
1572 \fi
1573 }
1574
1575 \newcommand*\leftpstartnumL{%
1576 \ifpstartnum
1577 \thepstartL
1578 \kern\linenumsep\global\pstartnumfalse\fi
1579 }
1580 \newcommand*\rightpstartnumL{%
1581 \ifpstartnum\kern\linenumsep
1582 \thepstartL
1583 \global\pstartnumfalse\fi
1584 }
1585 \newif\ifpstartnumR
1586 \pstartnumRtrue
1587 \newcommand*\leftpstartnumR{%
1588 \ifpstartnumR
1589 \thepstartR
1590 \kern\linenumsep\global\pstartnumRfalse\fi
1591 }
1592 \newcommand*\rightpstartnumR{%
1593 \ifpstartnumR\kern\linenumsep
1594 \thepstartR
1595 \global\pstartnumRfalse\fi
1596 }

```

```
1597 %
```

## VIII.6 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.

```
1598 \list@create{\inserts@listR}
1599 %
```

`\add@insertsR` The right text version.

```
\add@inserts@nextR
1600 \newcommand*{\add@insertsR}{%
1601   \global\let\add@inserts@nextR=\relax
1602   \ifx\inserts@listR\empty \else
1603     \ifx\next@insertR\empty
1604       \ifx\insertlines@listR\empty
1605         \global\noteschanged@true
1606         \gdef\next@insertR{100000}%
1607       \else
1608         \gl@p\insertlines@listR\to\next@insertR
1609       \fi
1610     \fi
1611     \ifnum\next@insertR=\absline@numR
1612       \gl@p\inserts@listR\to\@insertR
1613       \@insertR
1614       \global\let\@insertR=\undefined
1615       \global\let\next@insertR=\empty
1616       \global\let\add@inserts@nextR=\add@insertsR
1617     \fi
1618   \fi
1619   \add@inserts@nextR}
1620
1621 %
```

## VIII.7 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 are working on at the moment. The count `\@l@dttempcnta` 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 does not 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. Peter Wilson thinks 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, Peter Wilson ends up with the following.

```

1622 \newcommand*{\add@penaltiesL}{%
1623 \newcommand*{\add@penaltiesR}{%
1624
1625 %

```

### VIII.8 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.

```

1626 \newcommand*{\flush@notesR}{%
1627 \xloop
1628 \ifx\inserts@listR\empty \else
1629 \glp\inserts@listR\to\@insertR
1630 \@insertR
1631 \global\let\@insertR=\undefined
1632 \repeat}
1633
1634 %

```



## IX Footnotes

### IX.1 Footnotes output specific to \Pages

`\print@Xnotes@forpages`    The `\Xonlyside` and `\onlysideX` hooks for `\Pages` allow notes to be printed either in left or right pages only. The implementation of such features is delegated to `\correct@Xfootins@box`  
`\print@notesX@forpages`    `\print@Xnotes@forpages`, which replaces `\print@Xnotes` inside `\Pages`. Here is how we proceed<sup>6</sup>:  
`\correct@footinsX@box`

- If notes are to be printed in both sides, we just proceed the usual way: print the foot starts for the series, then the foot group.
- If notes are to be printed in the left side, we do these prints only for even pages ; if notes are to be printed in the right side, we do these prints only for odd pages.
- However, that is not enough. Because the problem does not only consists in printing notes in any particular page. It is also not to put aside room for notes in the pages where we do not want to print them. To take an example: if some note in the left side is too long by 160pt to be printed in full in the left page, we do not want to put aside 160pt a space for it in the following right page.
- To solve this problem, we change the magnification factor associated with notes before going to the next page. If we start a page where no notes are supposed to be printed, the magnification counter is set to 0. The dimension associated to footnote is set to `\maxdimen`, and so we can keep all the notes we want, without any break inside. We also set the note skip to 0pt. Before starting a new page where these notes are supposed to be printed, we reset these counter and skip to their default values. (About these counter, dimension and skip, read *The TeXbook* p. 122-125).
- In the output macro of the page where notes must NOT be printed, we store the content of the footnote box produced by T<sub>E</sub>X to a temporary box.
- After going to the next page, before typesetting any thing in this page, we put the content of this temporary box the footnote insert box.

The code to print critical notes, when processing `\Pages`, called in the output routine.

```
1635 \newcommand\print@Xnotes@forpages[1]{%
1636 %
```

First case: notes are for both sides. Just print the note start and the note group

```
1637 \ifcseempty{Xonlyside@#1}{%
1638 \csuse{#1footstart}{#1}%
1639 \csuse{#1footgroup}{#1}%
1640 }%
1641 %
```

<sup>6</sup>See <http://tex.stackexchange.com/a/230332/7712>.

Second case: notes are for one side only. First test if we are in a page where they must be printed.

```

1642   {%
1643     \ifboolexpr{%
1644       ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page
1645       }})%
1646       or%
1647       (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}}))%
1648     }%

```

If we are in a page where notes must be printed, print the notes.

```

1649   {%
1650     \csuse{#1footstart}{#1}%
1651     \csuse{#1footgroup}{#1}%
1652   }%

```

Then, set to not to keep room for notes in the next page. Also set to that, in the next page, notes are not to be split, using `\maxdimen`.

```

1653     \global\count\csuse{#1footins}=\z@%
1654     \global\skip\csuse{#1footins}=\z@%
1655     \global\dimen\csuse{#1footins}=\maxdimen%
1656   }%
1657 }%

```

In case we are on a page where notes must NOT be printed. First restore expected rooms for notes on the next page. Also reset expected vertical size allowed to notes.

```

1658   {%
1659     \global\count\csuse{#1footins}=\csuse{default@#1footins}%
1660     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
1661     \bgroup%
1662       \csuse{Xnotefontsize@#1}%
1663       \global\dimen\csuse{#1footins}=\csuse{Xmaxhnotes@#1}%
1664     \egroup%
1665   }%

```

Then, save the current insert box to a temporary insert box.

```

1666     \global\setnamebox{#1footins@kept}=\box\namebox{#1footins}%
1667   }%
1668 }%

```

End of `\print@Xnotes@forpages`.

```

1669   }%
1670 }%
1671 }%

```

And now, the same for familiar footnotes.

```

1672 \newcommand\print@notesX@forpages[1]{%
1673   \ifcsemt{onlysideX@#1}{%
1674     \csuse{footstart#1}{#1}%
1675     \csuse{footgroup#1}{#1}%
1676   }%
1677   {%
1678     \ifboolexpr{%
1679       ((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page
1680       }})%
1681       or%
1682       (test {\ifcsstring{onlysideX@#1}{R}} and test{\ifnumodd{\c@page}}))%
1683     }%
1684     \csuse{footstart#1}{#1}%
1685     \csuse{footgroup#1}{#1}%
1686     \global\count\csuse{footins#1}=\z@%
1687     \global\skip\csuse{footins#1}=\z@%
1688     \global\dimen\csuse{footins#1}=\maxdimen%
1689   }%
1690   {%
1691     \global\count\csuse{footins#1}=\csuse{default@footins#1}%
1692     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
1693     \bgroup%
1694       \csuse{Xnotefontsize@#1}%
1695       \global\dimen\csuse{footins#1}=\csuse{maxhnotesX@#1}%
1696     \egroup%
1697     \global\setnamebox{footins#1@kept}=\box\namebox{footins#1}%
1698   }%
1699 }%
1700 }%
1701 }%
1702 %

```

**\insert@notes@for@onlyside** \insert@notes@for@onlyside is everytime \Pages go to the next side. It just reinsert the notes note printed on the previous side because of Xonlyside or \onlysideX setting.

```

1703 \newcommand{\insert@notes@for@onlyside}{%
1704   \def\do##1{%
1705     \unless\ifnocritical%
1706       \ifvoid\csuse{##1footins@kept}\else%
1707         \expandafter\insert\csname ##1footins\endcsname%
1708         \bgroup%
1709           \unvnamebox{##1footins@kept}%
1710         \egroup%
1711       \fi%
1712     \fi%
1713     \unless\ifnofamiliar%
1714       \ifvoid\csuse{footins##1@kept}\else%
1715         \expandafter\insert\csname footins##1\endcsname%

```

```

1716     \bgroup%
1717     \unvnamebox{footins##1@kept}%
1718     \egroup%
1719     \fi%
1720     \fi%
1721     }%
1722     \dolistloop{\@series}%
1723 }%
1724 %

```

## IX.2 Critical footnote printed in right side

`\edtext@later` and `\edtextnow` are used to print critical footnotes on the right side, while referring to the text on the left side. First, we create two counters, one for `\edtext@later` and the other for `\edtextnow`.

```

1725 \newcount\edtext@now%
1726 \newcount\edtext@later%
1727 %
1728 \newcommand{\edtextlater}[2]{%#1 lemma, #2 critical notes
1729 %

```

First, increase the `\edtext@later` counter.

```

1730 \global\advance\edtext@later by \one%
1731 %
1732 % As we are in a pseudo-\cs{edtext}, we now need to get the \cs{sameword}
data stored in the auxiliary file for this \cs{edtextlater}.
1733 \advance\edtext@level by \one%
1734 \ifcsvoid{sw@list@edtext@the\edtext@level}%
1735     {\global\let\sw@inthisedtext\empty}%
1736     {\expandafter\gl@p\csname sw@list@edtext@the\edtext@level\
endcsname\to\sw@inthisedtext}%
1737 %

```

The main feature of `\edtextlater` is to create a macro which will be called on the equivalent `\edtextnow`.

```

1738 %
1739 \csxdef{edtext@later@the\edtext@later}{%
1740 %

```

`\edtextnow` will insert a empty `\edtext`.

```

1741 \noexpand\edtext{%
1742 }%
1743 %

```

With a `\lemma` and `\linenum` defined by the current `\edtextlater`. Also with `\sameword` data gotten from the current `\edtextlater`

```

1744 {%
1745   \unexpanded{%
1746     \lemma{\no@expands #1}%
1747     \xxref{start:edtext:later:\the\edtext@later}{end:edtext:later:\the\
edtext@later}%
1748   }%
1749   \noexpand\linenum{|||||\edfont@info}%
1750   \unexpanded{\def\sw@inthisedtext}{\expandonce{\sw@inthisedtext}}}%
1751 %

```

As the `\edtextnow` is generally called on the other side than the corresponding `\edtextlater`, we need to store the side for a proper formatting of the footnote.

```

1752   \ifledRcol%
1753     \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@true}}%
1754   \else%
1755     \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@false}}%
1756   \fi%
1757 %

```

And the footnote command of this `\edtextlater`.

```

1758   \unexpanded{#2}%
1759 }%
1760 }%
1761 %

```

And now, we print the current lemma data. But we save the beginning and the starting line using the crossref mechanism. We also store information in the auxiliary file about the existence of a `\edtextlater` and, if required, about the use of a `\lemma`

```

1762 \edlabel{start:edtext:later:\the\edtext@later}%
1763 \flag@start@later%
1764 \bgroup%
1765   \def\lemma##1{%
1766     \ifledRcol%
1767       \write\linenum@outR{\string\@lemma}%
1768     \else%
1769       \write\linenum@out{\string\@lemma}%
1770     \fi%
1771   }%
1772   \renewcommand{\do}[1]{\expandafter\renewcommandx\csname ##1footnote\
endscname[2][1,usedefault]{}}%
1773   \dolistloop{\@series}%
1774   #2%
1775 \egroup%
1776 \showlemma{#1}%
1777 \edlabel{end:edtext:later:\the\edtext@later}%
1778 \flag@end@later%
1779 %

```

We decrease the counter increased at the beginning.

```

1780 \advance\@edtext@level by -\@ne%
1781 }%
1782 %

```

\edtextnow just calls the command defined as is, reading the \edtext@later list.

```

1783 \newcommand{\edtextnow}[0]{%
1784 \global\advance\edtext@now by \@ne
1785 \csuse{edtext@later@\the\edtext@now}%
1786 }%
1787 %

```

## X 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.

```

1788 \list@create{\labelref@listR}
1789
1790 %

```

**\edlabel** This command is defined only one time in reledmac, including features for reledpar.

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

```

1791 \def\l@dmake@labelsR#1|#2|#3|#4|#5{%
1792 \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
else
1793 \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
1794 \fi
1795 \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
{#1|#2|#3|#4|\@Rlineflag}%
1796 \ignorespaces}
1797 \AtBeginDocument{%
1798 \def\l@dmake@labelsR#1|#2|#3|#4|#5{%
1799 }
1800
1801 %

```

**\@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, \@nl, and the \sub@on and \sub@off commands appearing in the \linenum@out file.

It is defined on reledmac.

## XI Side notes

Regular `\marginpars` do not work inside numbered text — they do not produce any note but do put an extra unnumbered blank line into the text.

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

```
\sidenotemargin*
1802 \WithSuffix\newcommand\sidenotemargin*[1]{%
1803   \l@dgetsidenote@margin{#1}
1804   \global\sidenote@marginR=\@l@tempcntb
1805   \global\sidenote@margin=\@l@tempcntb
1806 }
1807 \newcount\sidenote@marginR
1808 \global\sidenote@margin=\@ne
1809
1810 %
```

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

```
1811 \newcommand*\affixside@noteR{%
1812   \def\sidenotecontent@{%
1813     \numgdef{\itemcount@}{0}%
1814     \def\do##1{%
1815       \ifnumequal{\itemcount@}{0}%
1816         {%
1817           \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
1818         {\appto\sidenotecontent@{\sidenotesep ##1}%
1819         }%
1820         \numgdef{\itemcount@}{\itemcount@+1}%
1821       }%
1822     \dolistloop{\l@dcsnotetext}%
1823     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
1824     \gdef\@templ@d{%
1825       \gdef\@templ@n{\l@dcsnotetext\l@dcsnotetext@1\l@dcsnotetext@r}%
1826       \ifx\@templ@d\@templ@n \else%
1827         \if@twocolumn%
1828           \if@firstcolumn%
1829             \setl@dlp@rbox{##1}{\sidenotecontent@}%
1830           \else%
1831             \setl@drp@rbox{\sidenotecontent@}%
1832           \fi%
1833         \else%
1834           \@l@tempcntb=\sidenote@marginR%
1835           \ifnum\@l@tempcntb>\@ne%
1836             \advance\@l@tempcntb by\page@numR%
1837           \fi%
1838           \ifodd\@l@tempcntb%
1839             \setl@drp@rbox{\sidenotecontent@}%
1840             \gdef\sidenotecontent@{%
```

```

1841 \numdef{\itemcount@}{0}%
1842 \dolistloop{\l@dcsnotetext@l}%
1843 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
1844 \setl@dlp@rbox{\sidenotecontent@}%
1845 \else%
1846 \setl@dlp@rbox{\sidenotecontent@}%
1847 \gdef\sidenotecontent@{}%
1848 \numdef{\itemcount@}{0}%
1849 \dolistloop{\l@dcsnotetext@r}%
1850 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
1851 \setl@drp@rbox{\sidenotecontent@}%
1852 \fi%
1853 \fi%
1854 \fi%
1855 }
1856
1857 %

```

## XII Verse

Like in `reledmac`, the insertion of `hangingsymbol` is base on `\ifinserthangingsymbol`, and, for the right side, on `\ifinserthangingsymbolR`. Both commands also include the hanging space, to be sure the `\one@line` of hanging lines has the same width that the `\one@line` of normal lines and to prevent the column separator from shifting.

```

\inserthangingsymbolL58 \newif\ifinserthangingsymbolR
\inserthangingsymbolR59 \newcommand{\inserthangingsymbolL}{%
1860 \ifinserthangingsymbol%
1861 \ifinstanzaL%
1862 \hskip \@ifundefined{sza@00}{0}{\expandafter%
1863 \noexpand\csname sza@00\endcsname}\stanzaindentbase%
1864 \@hangingsymbol%
1865 \fi%
1866 \fi%
1867 }%
1868 \newcommand{\inserthangingsymbolR}{%
1869 \ifinserthangingsymbolR%
1870 \ifinstanzaR%
1871 \hskip \@ifundefined{sza@00}{0}{\expandafter%
1872 \noexpand\csname sza@00\endcsname}\stanzaindentbase%
1873 \@hangingsymbol%
1874 \fi%
1875 \fi%
1876 }%
1877 %

```

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.



```

1878 \chardef\next=\catcode\&
1879 \catcode\&=\active
1880
1881 %

```

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

```

1882 \newenvironment{astanza}[1][\catcode\&=\active
1883 \global\stanza@count\@ne\stanza@modulo\@ne
1884 \ifnum\usernamecount{sza@00}=\z@
1885 \let\stanza@hang\relax
1886 \let\endlock\relax
1887 \else
1888 \rightskip\z@ plus 1fil\relax
1889 \fi
1890 \ifnum\usernamecount{szp@00}=\z@
1891 \let\sza@penalty\relax
1892 \fi
1893 \def&{%
1894 \endlock\mbox{}%
1895 \sza@penalty
1896 \global\advance\stanza@count\@ne
1897 \@astanza@line}%
1898 \def\&{\@stopastanza}%
1899 \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}} and test{\ifstrempy
1900 {#1}}}%
1901 {\pstart[\at@every@stanza]}%
1902 {\pstart[#1]}%
1903 \@astanza@line
1904 \let\par\relax\ignorespaces%No paragraph in verses
1905 }{}
1906
1907 %

```

`\@stopastanza` This command is called by `\&` in `astanza` environment. It allows optional arguments.

```

1908 \newcommandx{\@stopastanza}[1][1,usedefault]{%
1909 \endlock\mbox{}%
1910 \ifboolexpr{not test{\ifdefvoid{\at@every@stop@stanza}} and test{\
1911 ifstrempy{#1}}}%
1912 {\pend[\at@every@stop@stanza]}%
1913 {\pend[#1]}%
1914 }%

```

`\@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.

```

1915 \newcommand*{\@astanza@line}{%
1916   \ifnum\value{stanzaindentsrepetition}=0
1917     \parindent=\csname sza@\number\stanza@count
1918       @\endcsname\stanzaindentbase
1919   \else
1920     \parindent=\csname sza@\number\stanza@modulo
1921       @\endcsname\stanzaindentbase
1922     \managestanza@modulo
1923   \fi
1924   \endgraf
1925   \stanza@hang%
1926   \ignorespaces}
1927
1928 %

```

Lastly reset the modified category codes.

```

1929 \catcode`\&=\next
1930
1931 %

```

`\thestanzaL` And now, the left and right stanza counter.

```

\thestanzaR
1932 \newcounter{stanzaL}
1933 \newcounter{stanzaR}
1934 \renewcommand{\thestanzaL}{%
1935   \textbf{\arabic{stanzaL}}%
1936 }
1937 \renewcommand{\thestanzaR}{%
1938   \textbf{\arabic{stanzaR}}%
1939 }
1940 %
1941 %

```

## XIII Fixing babel and polyglossia

With parallel texts there is the possibility that the two sides might use different languages via `babel`. On the other hand, nor `babel` nor `polyglossia` 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
\l@dusedbabeltrue

```

```

1942 \newif\ifl@dusedbabel
1943 %

```

`\l@dchecklang`

`\bbl@set@language` In babel the macro `\bbl@set@language{⟨lang⟩}` does the work when the language `⟨lang⟩` is changed via `\selectlanguage`. Unfortunately for us, if it is given an argument in the form of a control sequence it strips off the `\` character rather than expanding the command. We need a version that accepts an argument in the form `\lang` without it stripping the `\`.

```

1944 \patchcmd{\bbl@set@language}%
1945   {\select@language{\language}}%
1946   {\edef\language{#1}\select@language{\language}}%
1947   {}%
1948   {}%
1949
1950 %

```

The rest of the setup has to be postponed until the end of the preamble when we know if babel or polyglossia have 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 similar  
`\theledlanguageL` to `\selectlanguage`.  
`\theledlanguageR`

```

1951 \newcommand*{\l@duselanguage}[1]{%
1952   \gdef\theledlanguageL{#1}%
1953   \gdef\theledlanguageR{#1}%
1954
1955 %

```

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

```

1956 \AtBeginDocument{%
1957   \ifundefined{xpg@main@language}{%
1958     \ifundefined{bbl@main@language}{%
1959 %

```

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

```

1960   \l@dusedbabelfalse
1961   }{%
1962 %

```

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.

```

1963 \l@dusedbabeltrue
1964 \let\l@doldselectlanguage\selectlanguage
1965 \let\l@doldbbl@set@language\bbl@set@language
1966 \renewcommand{\selectlanguage}[1]{%
1967   \l@doldselectlanguage{#1}%
1968   \ifledRcol \gdef\theledlanguageR{#1}%
1969   \else      \gdef\theledlanguageL{#1}%
1970   \fi}
1971 %

```

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

```

1972 \renewcommand*{\l@duselanguage}[1]{%
1973   \expandafter\l@doldselectlanguage\expandafter{#1}}
1974 %

```

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

```

1975 \gdef\theledlanguageL{\bbl@main@language}%
1976 \gdef\theledlanguageR{\bbl@main@language}%
1977 }%
1978 }
1979 %

```

If use polyglossia

```

1980 { \let\old@otherlanguage\otherlanguage%
1981   \renewcommand{\otherlanguage}[2][]{%
1982     \selectlanguage{#1}{#2}%
1983     \ifledRcol \gdef\theledlanguageR{#2}%
1984     \else      \gdef\theledlanguageL{#2}%
1985     \fi}%
1986   \renewcommand{\l@duselanguage}[1]{%
1987     \csuse{no\language@name @numbers}\select@language{#1}%
1988   }%
1989   \gdef\theledlanguageL{\xpg@main@language}%
1990   \gdef\theledlanguageR{\xpg@main@language}%
1991 %

```

That is it.

```

1992 }}
1993 %

```

## XIV 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 default is  
`\l@dc@maxchunks` 5120 chunk pairs.

```
1994 \newcount\l@dc@maxchunks
1995 \newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}
1996 \maxchunks{5120}
1997
1998 %
```

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

```
1999 \newcount\l@dnumpstartsR
2000
2001 %
```

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

`\l@pscr`

```
2002 \newcount\l@dpscl
2003 \newcount\l@dpscr
2004
2005 %
```

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

```
2006 \newcommand*{\l@dsetuprawboxes}{%
2007 \l@l@tempcntb=\l@dc@maxchunks
2008 \loop\ifnum\l@l@tempcntb>\z@
2009 \newnamebox{\l@dLcolrawbox\the\l@l@tempcntb}
2010 \newnamebox{\l@dRcolrawbox\the\l@l@tempcntb}
2011 \advance\l@l@tempcntb \m@ne
2012 \repeat}
2013
2014 %
```

`\l@dsetupmaxlinecounts` To be able to synchronise left and right texts we need to know the maximum number of text lines there are in each pair of chunks. `\l@dsetupmaxlinecounts` creates `\maxchunks` new counts called `\l@dmaxlinesinpar1`, etc., and `\l@dzeromaxlinecounts` zeroes all of them.

```
2015 \newcommand*{\l@dsetupmaxlinecounts}{%
2016 \l@l@tempcntb=\l@dc@maxchunks
2017 \loop\ifnum\l@l@tempcntb>\z@
2018 \newnamecount{\l@dmaxlinesinpar\the\l@l@tempcntb}
2019 \advance\l@l@tempcntb \m@ne
2020 \repeat}
2021 \newcommand*{\l@dzeromaxlinecounts}{%
2022 \begingroup
2023 \l@l@tempcntb=\l@dc@maxchunks
2024 \loop\ifnum\l@l@tempcntb>\z@
```

```

2025 \global\usernamecount{l@dmxlinesinpar\the\l@dttempcntb}=\z@
2026 \advance\l@dttempcntb \m@ne
2027 \repeat
2028 \endgroup}
2029
2030 %

```

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

```

2031 \AtBeginDocument{%
2032 \l@dssetuprawboxes
2033 \l@dssetupmaxlinecounts
2034 \l@dzeromaxlinecounts
2035 \l@dnumpstartsL=\z@
2036 \l@dnumpstartsR=\z@
2037 \l@dpscL=\z@
2038 \l@dpscR=\z@}
2039
2040 %

```

## XV Checking text to be processed

```

\if@pstarts \check@pstarts returns \@pstartstrue if there are any unprocessed chunks.
\@pstartstrue
\@pstartsfalse
\check@pstarts
2041 \newif\if@pstarts
2042 \newcommand*{\check@pstarts}{%
2043 \@pstartsfalse
2044 \ifnum\l@dnumpstartsL>\l@dpscL
2045 \@pstartstrue
2046 \else
2047 \ifnum\l@dnumpstartsR>\l@dpscR
2048 \@pstartstrue
2049 \fi
2050 \fi
2051 }
2052
2053 %

```

```

\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 \araw@textfalse.
\checkraw@text
2054 \newif\ifaraw@text
2055 \newcommand*{\checkraw@text}{%
2056 \araw@textfalse
2057 \ifvbox\namebox{l@dLcolrawbox\the\l@dpscL}
2058 \araw@texttrue
2059 \else

```

```

2060 \ifvbox\namebox{l@dRcolrawbox\the\l@dpscR}
2061 \araw@texttrue
2062 \fi
2063 \fi
2064 }
2065
2066 %

```

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

```

2067 \newcommand*\@writelinesinparL{%
2068 \edef\next{%
2069 \write\linenum@out{\string\@pend[\the\@donereallinesL]}}%
2070 \next
2071 \global\@donereallinesL \z@}
2072 \newcommand*\@writelinesinparR{%
2073 \edef\next{%
2074 \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}%
2075 \next
2076 \global\@donereallinesR \z@}
2077
2078 %

```

`\@writepageofparL` These write the pages where start the first line of a chunk.

```

\@writepageofparR
2079 \newcommand*\@writepageofparL[0]{%
2080 \ifnum\@donereallinesL=\z@%
2081 \edef\next{%
2082 \write\linenum@out{\string\@pstart{\the\l@dpscL}{\the\c@page}{\the\
numpagelinesL}}%
2083 }%
2084 \next%
2085 \fi%
2086 }%
2087 \newcommand*\@writepageofparR[0]{%
2088 \ifnum\@donereallinesR=\z@%
2089 \edef\next{%
2090 \write\linenum@outR{\string\@pstartR{\the\l@dpscR}{\the\c@page}{\the\
numpagelinesR}}%
2091 }%
2092 \next%
2093 \fi%
2094 }%
2095 %

```

## XVI Parallel columns

`\@eledsectionL` The parbox `\@eledsectionL` and `\@eledsectionR` will keep the sections' title.  
`\@eledsectionR`

```

2096 \newsavebox{\@eledsectionL}%
2097 \newsavebox{\@eledsectionR}%
2098 %

```

**\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.

```

2099 \newcommand*{\Columns}{%
2100   \ifl@dpairing%
2101     \led@err@Columns@InsideEnv%
2102   \fi%
2103   \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
2104     \led@err@Columns@WithoutEnv%
2105   \else%
2106     \global\l@dprintingcolumnstrue%
2107     \eledsection@correcting@skip=-\baselineskip% Correction for sections'
titles
2108     \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else
2109       \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
2110     \fi
2111 %

```

Start a group and zero counters, etc.

```

2112 \begingroup
2113   \l@dzeropenalties
2114   \endgraf\global\num@lines=\prevgraf
2115   \global\num@linesR=\prevgraf
2116   \global\par@line=\z@
2117   \global\par@lineR=\z@
2118   \global\l@dpscL=\z@
2119   \global\l@dpscR=\z@
2120   \get@familiarfootnote@number%
2121 %

```

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

```

2122 \check@pstarts
2123 \loop\if@pstarts
2124   \global\pstartnumtrue
2125   \global\pstartnumRtrue
2126 %

```

Increment `\l@dpscL` and `\l@dpscR` which here count the numbers of left and right chunks. Also restore the value of the public `pstart` counters.

```

2127   \global\advance\l@dpscL \@ne
2128   \global\advance\l@dpscR \@ne
2129   \restore@pstartL@pc%
2130   \restore@pstartR@pc%
2131 %

```



We print the optional argument of `\pstart` or the argument of `\AtEveryPstart`.

```
2132 \Columns@print@before@pstart%
2133 %
```

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

```
2134 \checkraw@text
2135 { \loop\ifaraw@text
2136 %
```

Grab the next pair of left and right text lines and output them, swapping languages if they differ, adding section title if needed.

```
2137 \l@duselanguage{\theledlanguageL}%
2138 \do@lineL
2139 \xifinlist{\the\l@dpscL}{\eled@sections@@}
2140 {%
2141 \ifdefstring{\@eledsectmark}{L}%
2142 {\csuse{eled@sectmark@the\l@dpscL}%
2143 }{}}%
2144 \global\csundef{eled@sectmark@the\l@dpscL}%
2145 \savebox{\@eledsectionL}{\parbox[t][][t]{\Lcolwidth}{\vbox
2146 {\print@eledsectionL}}}%\vbox{ }-> prevent alignment troubles with RTL
2147 language
2148 }%
2149 {}%
2150 \l@duselanguage{\theledlanguageR}%
2151 \do@lineR
2152 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}
2153 {%
2154 \ifdefstring{\@eledsectmark}{R}%
2155 {\csuse{eled@sectmark@the\l@dpscR}%
2156 }{}}%
2157 \global\csundef{eled@sectmark@the\l@dpscR}%
2158 \savebox{\@eledsectionR}{\parbox[t][][t]{\Rcolwidth}{\vbox
2159 {\print@eledsectionR}}}%\vbox{ }-> prevent alignment troubles with RTL
2160 language
2161 {}%
2162 \hb@xt@ \hsize{%
2163 \ifdefstring{\columns@position}{L}{\hfill }%
2164 \unhbox\l@leftbox%
2165 \ifhbox{\@eledsectionL}%
2166 \usebox{\@eledsectionL}%
2167 \fi%
2168 \print@columnseparator%
2169 \unhbox\l@rightbox%
2170 \ifhbox{\@eledsectionR}%
2171 \usebox{\@eledsectionR}%
2172 \fi%
2173 \ifdefstring{\columns@position}{R}{\hfill}%
```

```

2170     }%
2171     \checkraw@text
2172     \checkverseL
2173     \checkverseR
2174     \checkpb@columns
2175     \repeat}
2176 %

```

Having completed a pair of chunks, write the number of lines in each chunk to the respective section files. Increment `pstart` counters and reset line numbering if it is by `pstart`.

```

2177     \@writelinesinparL
2178     \@writelinesinparR
2179     \check@pstarts
2180     \ifbypstart@%
2181         \write\linenum@out{\string\@set[1]}
2182         \resetprevline@
2183     \fi
2184     \ifbypstart@R
2185         \write\linenum@outR{\string\@set[1]}
2186         \resetprevline@
2187     \fi
2188     \Columns@print@after@pend%
2189     \repeat
2190 %

```

Having output all chunks, make sure all notes have been output, then zero counts ready for the next set of texts. The boolean tests for stanza are switched to false.

```

2191     \flush@notes
2192     \flush@notesR
2193     \endgroup
2194 %

2195     \global\l@dpscL=\z@
2196     \global\l@dpscR=\z@
2197     \global\l@dnpstartsL=\z@
2198     \global\l@dnpstartsR=\z@
2199     \global\l@dprintingcolumnsfalse%
2200     \ignorespaces
2201     \global\instanzaLfalse%
2202     \global\instanzaRfalse%
2203     \fi}
2204
2205 %

```

`\print@columnseparator` `\print@columnseparator` prints the column separator, with surrounding spaces (as the user has set them). We use the  $\text{\TeX}$  `\ifdim` instead of `etoolbox` to avoid having `\hfill` in a `{}`, which deletes some space (but not much).

```

2206 \def\print@columnseparator{%
2207   \ifdim\beforecolumnseparator<0pt%
2208     \hfill%
2209   \else%
2210     \hspace{\beforecolumnseparator}%
2211   \fi%
2212   \columnseparator%
2213   \ifdim\aftercolumnseparator<0pt%
2214     \hfill%
2215   \else%
2216     \hspace{\beforecolumnseparator}%
2217   \fi%
2218 }%
2219 %

```

**\checkpb@columns** \checkpb@columns prevent or make pagebreaking in columns, depending of the use of \ledpb or \lednopb.

```

2220
2221 \newcommand{\checkpb@columns}{%
2222   \newif\if@pb
2223   \newif\if@nopb
2224   \IfStrEq{\led@pb@setting}{before}{
2225     \numdef{\next@absline}{\the\absline@num+1}%
2226     \numdef{\next@abslineR}{\the\absline@numR+1}%
2227     \xifinlistcs{\next@absline}{l@prev@pb}{\@pbtrue}{}%
2228     \xifinlistcs{\next@abslineR}{l@prev@pbR}{\@pbtrue}{%
2229     \xifinlistcs{\next@absline}{l@prev@nopb}{\@nopbtrue}{}%
2230     \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\@nopbtrue}{%
2231     }{}
2232     \IfStrEq{\led@pb@setting}{after}{
2233     \xifinlistcs{\the\absline@num}{l@prev@pb}{\@pbtrue}{}%
2234     \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\@pbtrue}{%
2235     \xifinlistcs{\the\absline@num}{l@prev@nopb}{\@nopbtrue}{}%
2236     \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\@nopbtrue}{%
2237     }{}
2238     \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
2239     \if@pb\pagebreak[4]\fi
2240   }
2241   %

```

**\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).

```

2242 \newcommand*{\columnseparator}{%
2243   \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}}
2244 \newdimen\columnrulewidth
2245 \columnrulewidth=\z@

```

2246

2247 %

`\columnspan` The position of the `\Columns` in a page. Default value is R. Stored in `\columns@position`.  
`\columns@position`

2248 `\newcommand*{\columnspan}[1]{%`2249 `\xdef\columns@position{#1}%`2250 `}%`2251 `\xdef\columns@position{R}%`

2252 %

`\beforecolumnseparator` `\beforecolumnseparator` and `\aftercolumnseparator` lengths are defined to -1pt.  
`\aftercolumnseparator` If user changes them to a positive length, the lengths are used to define blank spaces before / after the column separator, instead of `\hfill`.

2253 `\newlength{\beforecolumnseparator}%`2254 `\setlength{\beforecolumnseparator}{-2pt}%`

2255

2256 `\newlength{\aftercolumnseparator}%`2257 `\setlength{\aftercolumnseparator}{-2pt}%`

2258

2259 %

`setwidthliketwocolumns@L` The `\setwidth...` macros are called in `\beginnumbering` in a **non-parallel** typesetting context, to fix the width of the lines to be vertically aligned with parallel columns. They are also called at the beginning of a note's group, if some options are enabled. The `setnotepositionliketwocolumns@L` `setwidthliketwocolumns@C` `setpositionliketwocolumns@C` `setnotepositionliketwocolumns@C` `setwidthliketwocolumns@R` `setpositionliketwocolumns@R` `setnotepositionliketwocolumns@R` `\xxxfootstart` in a **non-parallel** typesetting context to fix the position of notes block.

2260 `\newcommand{\setwidthliketwocolumns@L}{%`2261 `% Temporary dimension, initially equal to the standard hsize, i.e. text width`2262 `% \begin{macrocode}`2263 `\newdimen\temp%`2264 `\temp=\hsize%`

2265 %

Hsize : Left + Right width

2266 `\hsize=\Lcolwidth%`2267 `\advance\hsize\Rcolwidth%`

2268 %

Now, calculating the remaining space

2269 `\advance\temp-\hsize%`

2270 %

And multiply the hsize by 2/3 of this space

```

2271 \multiply\temp by 2%
2272 \divide\temp by 3%
2273 \advance\hsize\temp%
2274 }%
2275
2276 \newcommand{\setpositionliketwocolumns@L}{%
2277 \renewcommand{\ledrlfill}{\hfill}%
2278 }%
2279
2280 \newcommand{\setnotespositionliketwocolumns@L}{%
2281 }%
2282
2283
2284 %
2285
2286 \newcommand{\setwidthliketwocolumns@C}{%
2287 % Temporary dimension, initially equal to the standard hsize, i.e. text
2288 % width
2289 %
2290
2291 \newdimen\temp%
2292 \temp=\hsize%
2293 % Hsize : Left + Right width
2294 %
2295
2296 \hsize=\Lcolwidth%
2297 \advance\hsize\Rcolwidth%
2298 % Now, calculating the remaining space
2299 %
2300
2301 \advance\temp-\hsize%
2302 %
2303
2304 And multiply the hsize by 1/2 of this space
2305
2306 \divide\temp by 2%
2307 \advance\hsize\temp%
2308 }%
2309
2310 \newcommand{\setpositionliketwocolumns@C}{%
2311 \doinsidelinehook{\hfill}%
2312 \renewcommand{\ledrlfill}{\hfill}%
2313 }%
2314
2315 \newcommand{\setnotespositionliketwocolumns@C}{%
2316 \newdimen\temp%
2317 \newdimen\tempa%
2318 \temp=\hsize%
2319 \tempa=\Lcolwidth%
2320 \advance\tempa\Rcolwidth%
2321 \advance\temp-\tempa%

```

```

2314 \divide\temp by 2%
2315 \leftskip=\temp%
2316 \rightskip=-\temp%
2317 }%
2318
2319 \newcommand{\setwidthliketwocolumns@R}{%
2320 %

```

Temporary dimension, initially equal to the standard hsize, i.e. text width

```

2321 \newdimen\temp%
2322 \temp=\hsize%
2323 %

```

Hsize : Left + Right width

```

2324 \hsize=\Lcolwidth%
2325 \advance\hsize\Rcolwidth%
2326 %

```

Now, calculating the remaining space

```

2327 \advance\temp-\hsize%
2328 %

```

And multiply the hsize by 2/3 of this space

```

2329 \multiply\temp by 2%
2330 \divide\temp by 3%
2331 \advance\hsize\temp%
2332 }%
2333
2334 \newcommand{\setpositionliketwocolumns@R}{%
2335 \doinsidelinehook{\hfill}%
2336 }%
2337
2338 \newcommand{\setnotespositionliketwocolumns@R}{%
2339 \newdimen\temp%
2340 \newdimen\tempa%
2341 \temp=\hsize%
2342 \tempa=\Lcolwidth%
2343 \advance\tempa\Rcolwidth%
2344 \advance\temp-\tempa%
2345 \divide\temp by 2%
2346 \leftskip=\temp%
2347 \rightskip=-\temp%
2348 }%
2349
2350 %

```

`\Columns@print@before@pstart` The `\Columns@print@before@pstart` and `\Columns@print@after@pend` print the content of the optional argument of `\pstart` / `\pend`. If this content is not empty, it also print the separator.

```

2351 \newcommand{\Columns@print@before@pstart}{%
2352   \ifboolexpr{%
2353     test{\ifcsstring{before@pstartL@the\l@dpscl}{\at@every@pstart}}%
2354     and test {\ifcsstring{before@pstartR@the\l@dpscr}{\at@every@pstart}}%
2355     and test {\ifdefempty{\at@every@pstart}}}%
2356     {%
2357     {%
2358       \hb@xt@ \hsize{%
2359         \ifdefstring{\columns@position}{L}{\hfill }%
2360         \par\parbox[t] [] [t]{\Lcolwidth}{%
2361           \csuse{before@pstartL@the\l@dpscl}%
2362         }%
2363         \print@columnseparator%
2364         \parbox[t] [] [t]{\Rcolwidth}{%
2365           \set@sectcountR%
2366           \csuse{before@pstartR@the\l@dpscr}%
2367         }%
2368         \ifdefstring{\columns@position}{R}{\hfill}%
2369       }%
2370     }%
2371     \global\csundef{before@pstartL@the\l@dpscl}%
2372     \global\csundef{before@pstartR@the\l@dpscr}%
2373   }%
2374 \newcommand{\Columns@print@after@end}{%
2375   \ifboolexpr{%
2376     test{\ifcsstring{after@endL@the\l@dpscl}{\at@every@end}}%
2377     and test {\ifcsstring{after@endR@the\l@dpscr}{\at@every@end}}%
2378     and test {\ifdefempty{\at@every@end}}}%
2379     {%
2380     {%
2381       \hb@xt@ \hsize{%
2382         \ifdefstring{\columns@position}{L}{\hfill }%
2383         \parbox[t] [] [t]{\Lcolwidth}{%
2384           \csuse{after@endL@the\l@dpscl}%
2385         }%
2386         \print@columnseparator%
2387         \parbox[t] [] [t]{\Rcolwidth}{%
2388           \set@sectcountR%
2389           \csuse{after@endR@the\l@dpscr}%
2390         }%
2391         \ifdefstring{\columns@position}{R}{\hfill}%
2392       }%
2393     }%
2394     \global\csundef{after@endL@the\l@dpscl}%
2395     \global\csundef{after@endR@the\l@dpscr}%
2396   }%
2397   %

```

## XVII Parallel pages

This is considerably more complicated than parallel columns.

### XVII.1 Specific counters

<code>\numpagelinesL</code>	Counts for the number of lines on a left or right page, and the smaller of the number of
<code>\numpagelinesR</code>	lines on a pair of facing pages.
<code>\l@dminpagelines</code>	
2398	<code>\newcount\numpagelinesL</code>
2399	<code>\newcount\numpagelinesR</code>
2400	<code>\newcount\l@dminpagelines</code>
2401	
2402	<code>%</code>

### XVII.2 Main macro

`\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.

```

2403 \newcommand*{\Pages}[1][1,usedefault]{%
2404   \ifl@dpairing%
2405     \led@err@Pages@InsideEnv%
2406   \fi%
2407   \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
2408     \led@err@Pages@WithoutEnv%
2409   \else%
2410     \ifstrequal{#1}{mainmatter}{\Pages@mainmattertrue}{\Pages@mainmatterfalse}
2411   }%
2412   \eledsection@correcting@skip=-2\baselineskip% line correcting for section
2413   titles.
2414   \parledgroup@notespacing@set@correction%
2415   \typeout{}%
2416   \typeout{***** PAGES *****}%
2417   \ifnum\l@dnumpestartsL=\l@dnumpestartsR\else%
2418     \led@err@BadLeftRightPstarts{\the\l@dnumpestartsL}{\the\l@dnumpestartsR}%
2419   \fi%
2420   %
2421   %
2422   %

```

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

```

2419 \cleartol@devenpage%
2420 \global\l@dprintingpagetrue%
2421 \begingroup%
2422 %

```

As `\Pages` must be called outside of the pages environment, we have to redefine the `\Lcolwidth` and `\Rcolwidth` lengths, to prevent false overfull hboxes.



```

2423 \setlength{\Lcolwidth}{\textwidth}%
2424 \setlength{\Rcolwidth}{\textwidth}%
2425 %
2426 \l@dzeroopenalties%
2427 \endgraf\global\num@lines=\prevgraf%
2428 \global\num@linesR=\prevgraf%
2429 \global\par@line=\z@%
2430 \global\par@lineR=\z@%
2431 \global\l@dpscL=\z@%
2432 \global\l@dpscR=\z@%
2433 \writtenlinesLfalse%
2434 \writtenlinesRfalse%
2435 \get@familiarfootnote@number%
2436 %

```

The footnotes are printed in a different way from expected in `reledmac`, as we may want to print the notes on one side only.

```

2437 \let\print@Xnotes\print@Xnotes@forpages%
2438 \let\print@notesX\print@notesX@forpages%
2439 %

```

Check if there are chunks to be processed.

```

2440 \check@pstarts%
2441 \loop\if@pstarts%
2442 %

```

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

```

2443 \global\advance\l@dpscL \@ne%
2444 \global\advance\l@dpscR \@ne%
2445 %

```

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

```

2446 \getlinesfromparlistL%
2447 \getlinesfromparlistR%
2448 \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}%
2449 {\usernamecount{l@dmaxlinesinpar\the\l@dpscL}}%
2450 \check@pstarts%
2451 \repeat%
2452 %

```

Zero the counts again, ready for the next bit.

```

2453 \global\l@dpscL=\z@%
2454 \global\l@dpscR=\z@%
2455 %

```

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

```

2456 \getlinesfrompagelistL%
2457 \getlinesfrompagelistR%
2458 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
2459 {\l@dminpagelines}%
2460 %

```

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.

```

2461 \check@pstarts%
2462 \if@pstarts%
2463 %

```

Increment the chunk counts to get the first pair. Restore also the value of public pstart counters.

```

2464 \global\advance\l@dpscL \@ne%
2465 \global\advance\l@dpscR \@ne%
2466 \restore@pstartL@pc%
2467 \restore@pstartR@pc%
2468 %

```

We have not processed any lines from these chunks yet, so zero the respective line counts.

```

2469 \global\@donereallinesL=\z@%
2470 \global\@donetotallinesL=\z@%
2471 \global\@donereallinesR=\z@%
2472 \global\@donetotallinesR=\z@%
2473 %

```

Start a loop over the boxes (chunks).

```

2474 \checkraw@text%
2475 %
2476 % \begingroup
2477 { \loop\ifaraw@text%
2478 %

```

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

```

2479 \checkpageL%
2480 \l@duselanguage{\theledlanguageL}%
2481 { \loop\ifl@dsamepage%
2482 %

```

Process the next (left) text line, adding it to the page. Eventually, adds the optional argument of pstart.

```

2483 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}%
2484 \csuse{before@pstartL@the\l@dpscL}%
2485 \global\csundef{before@pstartL@the\l@dpscL}%
2486 \do@lineL%
2487 \xifinlist{the\l@dpscL}{\eled@sections@@}

```

```

2488         {\print@eledsectionL}%
2489         {}%
2490         \advance\numpagelinesL \@ne%
2491 %

```

When using `shiftedpstarts` option, a `\l@dleftbox` with a null height is not printed. That means we do not insert blank lines at the end of a left chunk lower than the corresponding right chunk. However, a `\l@dleftbox` with a null height will advance the `\pagetotal` in any case. Because if we do not do this, the `\checkpageL` could let `\ifl@pagefull` to false, and consequently a `\@lopL` equal to 1000 could be written in the numbered file, even if all the lines actually needed for the current page have been printed. `l@dleftbox`

```

2492         \ifshiftedpstarts%
2493             \ifdim\ht\l@dleftbox>0pt%
2494                 \parledgroup@correction@notespacing{L}%
2495                 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2496             \else%
2497                 \unless\ifadvancedshiftedpstarts%
2498                     \dimen0=\pagetotal%
2499                     \advance\dimen0 by \baselineskip%
2500                     \global\pagetotal=\dimen0%
2501                 \else%
2502                     \ifnomaxlines%
2503                         \numdef{\@tmp}{\the\l@dpscL+1}%
2504                         \ifcsdef{minpage@pstart@\@tmp}{%
2505 minpage@pstart@\@tmp}%
2506                             {\dimen0=\pagetotal%
2507                             \advance\dimen0 by \baselineskip%
2508                             \global\pagetotal=\dimen0%
2509                             }%
2510                             {}%
2511                         }{}%
2512                     \fi%
2513                 \fi%
2514             \fi%
2515         \else%
2516             \parledgroup@correction@notespacing{L}%
2517             \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2518         \fi%
2519 %

```

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. Check if we have to print the optional argument of the last pend. Check if the page is full. Check if the verse is split in two subsequent pages. Check there is any forced page breaks. Reset the verse skipnumber boolean

```

2520         \get@nextboxL%
2521         \global\l@dskipversenumberfalse%
2522         \ifprint@last@after@pendL%

```

```

2523         \csuse{after@pendL@the\l@dpscL}%
2524         \global\csundef{after@pendL@the\l@dpscL}%
2525         \fi%
2526         \checkpageL%
2527         \checkverseL%
2528         \checkpbL%
2529         \repeat%
2530 %

```

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.

```

2531         \ifl@dpagefull%
2532         \@writelinesonpageL{\the\numpagelinesL}%
2533         \else%
2534         \@writelinesonpageL{1000}%
2535         \fi%
2536 %

```

Reset to zero the left-page line count, clear the page to get onto the facing (odd, right) page, and reinitialize the accumulated dimension of interline correction for notes in parallel ledgroup.

```

2537         \numpagelinesL \z@%
2538         \parledgroup@correction@notespacing@init%
2539         \clearl@dleftpage }%
2540 %

```

Now do the same for the right text.

```

2541         \checkpageR%
2542         \l@duselanguage{\theledlanguageR}%
2543 {
2544     \loop\ifl@dsamepage%
2545     \set@sectcountR%
2546     \ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{}%
2547     \csuse{before@pstartR@the\l@dpscR}%
2548     \global\csundef{before@pstartR@the\l@dpscR}%
2549     \do@lineR%
2550     \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
2551     {\print@eledsectionR}%
2552     {}%
2553     \advance\numpagelinesR \@ne%
2554     \ifshiftedpstarts%
2555         \ifdim\ht\l@drightbox>0pt%
2556         \parledgroup@correction@notespacing{R}%
2557         \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2558         \else%
2559         \unless\ifadvancedshiftedpstarts%
2560         \dimen0=\pagetotal%
2561         \advance\dimen0 by \baselineskip%

```

```

2561         \global\pagetotal=\dimen0%
2562     \else%
2563         \ifnomaxlines%
2564             \numdef{\@tmp}{\the\l@dpscR+1}%
2565             \ifcsdef{minpage@pstart@\@tmp}{%
2566                 \ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}%
2567                     {\dimen0=\pagetotal%
2568                     \advance\dimen0 by \baselineskip%
2569                     \global\pagetotal=\dimen0%
2570                     }%
2571                     {}%
2572                     }{}%
2573                 \fi%
2574             \fi%
2575         \fi%
2576     \else%
2577         \parledgroup@correction@notespacing{R}%
2578         \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2579     \fi%
2580     \get@nextboxR%
2581     \global\l@dskipversenumberRfalse%
2582     \ifprint@last@after@pendR%
2583         \csuse{after@pendR@\the\l@dpscR}%
2584         \global\csundef{after@pendR@\the\l@dpscR}%
2585     \fi%
2586     \checkpageR%
2587     \checkverseR%
2588     \checkpbR%
2589     \repeat%
2590     \ifl@dpagefull%
2591         \@writelinesonpageR{\the\numpagelinesR}%
2592     \else%
2593         \@writelinesonpageR{1000}%
2594     \fi%
2595     \numpagelinesR=\z@%
2596     \parledgroup@correction@notespacing@init%
2597 %

```

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

```

2598     \clearl@drighthouse}%
2599 %

```

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.

```

2600     \checkraw@text%
2601     \ifaraw@text%
2602         \getlinesfrompagelistL%
2603         \getlinesfrompagelistR%
2604         \l@dcalc@minoftwo{\cs@linesonpageL}{\cs@linesonpageR}%

```

```

2605             {\l@dminpagelines}%
2606         \fi%
2607     \repeat}%
2608 %

```

We have now output the text from all the chunks.

```

2609     \fi%
2610 %

```

Make sure that there are no inserts hanging around.

```

2611     \flush@notes%
2612     \flush@notesR%
2613     \endgroup%
2614 %

```

Zero counts ready for the next set of left/right text chunks. The boolean tests for stanza are switched to false.

```

2615     \global\l@dpscL=\z%
2616     \global\l@dpscR=\z%
2617     \global\l@dnumstartsL=\z%
2618     \global\l@dnumstartsR=\z%
2619     \global\instanzaLfalse%
2620     \global\instanzaRfalse%
2621     \global\l@dprintingpagesfalse%
2622 %

```

Check the consistency of \edtext@later and \edtext@now

```

2623     \ifnum\edtext@later=\edtext@now%
2624     \else%
2625         \led@error@edtext@later@now%
2626     \fi%
2627     \global\edtext@later=\z%
2628     \global\edtext@now=\z%
2629 %

```

Prevent final notes from overlapping the line number

```

2630     \finish@Pages@notes%
2631     \ignorespaces\fi}
2632
2633
2634 %

```

### XVII.3 Ensure all notes are printed at the end of parallel pages

**\finish@Pages@notes** This macro ensures that all long notes are printed at the end of \Pages typesetting, and that there are no more long notes left for the next pages.

```

2635 \newcommand{\finish@Pages@notes}{%
2636     \def\do##1{%
2637 %

```

First, declare footnote box if there was no previous declared. E.g. if familiar or critical notes were disabled by `reledmac`'s options.

```

2638 \ifnocritical@%
2639 \global\newnamebox{##1footins}%
2640 \fi
2641 \ifnofamiliar@%
2642 \global\newnamebox{footins##1}%
2643 \fi
2644 %

```

And now, add a `\newpage` if there is no more footnote to print.

```

2645 \ifvoid\csuse{##1footins}%
2646 \ifvoid\csuse{footins##1}\else%
2647 \newpage\null%
2648 \listbreak%
2649 \fi%
2650 \else%
2651 \newpage\null%
2652 \listbreak%
2653 \fi%
2654 }%
2655 \dolistloop{@series}%
2656 }%
2657 %

```

## XVII.4 Struts

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

```

\ledstrutR
2658 \newcommand*{\ledstrutL}{}
2659 \newcommand*{\ledstrutR}{}
2660
2661 %

```

## XVII.5 Page clearing

`\cleartoevenpage` `\cleartoevenpage`, which is defined in the memoir class, is like `\clear(double)page` except that we end up on an even page. `\cleartol@evenpage` is similar except that it first checks to see if it is already on an empty page.

```

2662 \providecommand{\cleartoevenpage}[1][\@empty]{%
2663 \clearpage
2664 \ifodd\c@page\hbox{##1}\clearpage\fi}
2665
2666 \newcommand*{\cleartol@evenpage}{%
2667 \ifdim\pagetotal<\topskip% on an empty page
2668 \else
2669 \clearpage

```

```

2670 \Pages@mainmatter%
2671 \fi
2672 \ifodd\c@page%
2673 \ifprevpgnotnumbered%
2674 \addtocounter{par@page}{-1}%
2675 \ifdef{\prevpgstyle}{\thispagestyle{\prevpgstyle}}{}%
2676 \fi%
2677 \hbox{}\clearpage%
2678 \fi%
2679 }%
2680 %

```

`\clearl@dleftpage` and `\clearl@drighthouse` get us onto an odd and even page, respectively, checking that we end up on the subsequent page. Both commands use `\newpage` and not `\clearpage`. Because `\clearpage` prints all footnotes before the next page, even if it has to add new empty pages, while `\newpage` does not. And as we want notes started in the left page continue in the right page and *vice-versa*, we must use `\newpage` and not `\clearpage`

```

2681 \newcommand*\clearl@dleftpage{%
2682 \ifdim\pagetotal=0pt\hbox{}\fi%
2683 \newpage%
2684 \insert@notes@for@onlyside%
2685 \ifodd\c@page\else
2686 \led@err@LeftOnRightPage
2687 \hbox{}%
2688 \cleardoublepage
2689 \fi}
2690
2691 \newcommand*\clearl@drighthouse{%
2692 \ifdim\pagetotal=0pt\hbox{}\fi%
2693 \newpage%
2694 \insert@notes@for@onlyside%
2695 \ifodd\c@page
2696 \led@err@RightOnLeftPage
2697 \hbox{}%
2698 \cleartoevenpage
2699 \fi}
2700
2701 %

```

## XVII.6 Lines managing

`\getlinesfromparlistL` gets the next entry from the `\linesinpar@listL` and puts it into `\cs@linesinparL`; if the list is empty, it sets `\cs@linesinparL` to 0. Similarly for `\getlinesfromparlistR`

```

2702 \newcommand*\getlinesfromparlistL{%
2703 \ifx\linesinpar@listL\empty

```



```

2704 \gdef\@cs@linesinparL{0}%
2705 \else
2706 \gl@p\linesinpar@listL\to\@cs@linesinparL
2707 \fi}
2708 \newcommand*\getlinesfromparlistR{%
2709 \ifx\linesinpar@listR\empty
2710 \gdef\@cs@linesinparR{0}%
2711 \else
2712 \gl@p\linesinpar@listR\to\@cs@linesinparR
2713 \fi}
2714
2715 %

```

`\getlinesfrompagelistL` `\getlinesfrompagelistL` gets the next entry from the `\linesonpage@listL` and puts it into `\@cs@linesonpageL`; if the list is empty, it sets `\@cs@linesonpageL` to 1000. Similarly for `\getlinesfrompagelistR`.

```

2716 \newcommand*\getlinesfrompagelistL{%
2717 \ifx\linesonpage@listL\empty
2718 \gdef\@cs@linesonpageL{1000}%
2719 \else
2720 \gl@p\linesonpage@listL\to\@cs@linesonpageL
2721 \fi}
2722 \newcommand*\getlinesfrompagelistR{%
2723 \ifx\linesonpage@listR\empty
2724 \gdef\@cs@linesonpageR{1000}%
2725 \else
2726 \gl@p\linesonpage@listR\to\@cs@linesonpageR
2727 \fi}
2728
2729 %

```

`\@writelinesonpageL` These macros output the number of lines on a page to the section file in the form of `\@lopL` or `\@lopR` macros.

```

2730 \newcommand*\@writelinesonpageL{[1]{%
2731 \edef\next{\write\linenum@out{\string\@lopL{#1}}}%
2732 \next}
2733 \newcommand*\@writelinesonpageR{[1]{%
2734 \edef\next{\write\linenum@outR{\string\@lopR{#1}}}%
2735 \next}
2736
2737 %

```

`\l@dcalc@maxoftwo` `\l@dcalc@maxoftwo{<num>}{<num>}{<count>}` sets `<count>` to the maximum of the two `<num>`.

Similarly `\l@dcalc@minoftwo{<num>}{<num>}{<count>}` sets `<count>` to the minimum of the two `<num>`.

```

2738 \newcommand*{\l@dcalc@maxoftwo}[3]{%
2739   \ifnum #2>#1\relax
2740     #3=#2\relax
2741   \else
2742     #3=#1\relax
2743   \fi}
2744 \newcommand*{\l@dcalc@minoftwo}[3]{%
2745   \ifnum #2<#1\relax
2746     #3=#2\relax
2747   \else
2748     #3=#1\relax
2749   \fi}
2750
2751 %

```

### XVII.7 Page break managing

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

```

\checkpageL
\checkpageL
2752 \newif\ifl@dsamepage
2753 \l@dsamepagetrue
2754 \newif\ifl@dpagetrue
2755
2756 \newcommand*{\checkpageL}{%
2757   \l@dpagetrue
2758   \l@dsamepagetrue
2759   \check@goal
2760   \ifdim\pagetotal<\ledthegoal
2761     \ifnum\numpagelinesL<\l@dminpagelines
2762       \else
2763         \ifnomaxlines%
2764         \else%
2765           \l@dsamepagefalse%
2766           \l@dpagetruefalse%
2767         \fi%
2768       \fi
2769     \else
2770       \l@dsamepagefalse
2771       \l@dpagetrue
2772     \fi%
2773   \ifprint@last@after@pendL%
2774     \l@dpagetruefalse%
2775     \l@dsamepagefalse%
2776   \print@last@after@pendLfalse%

```

```

2777 \fi%
2778 }%
2779
2780 \newcommand*\checkpageR{%
2781 \l@dpagetrue
2782 \l@dsamepagetrue
2783 \check@goal
2784 \ifdim\pagetotal<\ledthegoal
2785 \ifnum\numpagelinesR<\l@dminpagelines
2786 \else
2787 \ifnomaxlines%
2788 \else%
2789 \l@dsamepagefalse%
2790 \l@dpagetruefalse%
2791 \fi%
2792 \fi
2793 \else
2794 \l@dsamepagefalse
2795 \l@dpagetrue
2796 \fi%
2797 \ifprint@last@after@pendR%
2798 \l@dpagetruefalse%
2799 \l@dsamepagefalse%
2800 \print@last@after@pendRfalse%
2801 \fi%
2802 }%
2803
2804 %

```

**\checkpBL** \checkpBL and \checkpBR are called after each line is printed, and after the page is checked. These commands correct page breaks depending on \ledpb and \lednopb.

```

2805 \newcommand*\checkpBL{
2806 \IfStrEq{\ledpb@setting}{after}{
2807 \xifinlistcs{\the\absline@num}{\l@prev@pb}{\l@dpagetrue\l@dsamepagefalse}{
2808 \xifinlistcs{\the\absline@num}{\l@prev@nopb}{\l@dpagetruefalse\l@dsamepagetrue}{
2809 }{
2810 \IfStrEq{\ledpb@setting}{before}{
2811 \numdef{\next@absline}{\the\absline@num+1}
2812 \xifinlistcs{\next@absline}{\l@prev@pb}{\l@dpagetrue\l@dsamepagefalse}{
2813 \xifinlistcs{\next@absline}{\l@prev@nopb}{\l@dpagetruefalse\l@dsamepagetrue}{
2814 }{
2815 }
2816
2817 \newcommand*\checkpBR{
2818 \IfStrEq{\ledpb@setting}{after}{

```

```

2819 \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\l@dpagfulltrue\
l@dsamepagefalse}{\}
2820 \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\l@dpagfullfalse\
l@dsamepagetrue}{\}
2821 }{\}
2822 \IfStrEq{\led@pb@setting}{before}{
2823 \numdef{\next@abslineR}{\the\absline@numR+1}
2824 \xifinlistcs{\next@abslineR}{l@prev@pbR}{\l@dpagfulltrue\
l@dsamepagefalse}{\}
2825 \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\l@dpagfullfalse\
l@dsamepagetrue}{\}
2826 }{\}
2827 }
2828 %

```

**\checkverseL** \checkverseL and \checkverseR are called after each line is printed. They prevent page break inside line of verse.

**\checkverseR**

```

2829 \newcommand{\checkverseL}{
2830 \ifinstanzaL
2831 \iflednopbinverse
2832 \ifinserthangingsymbol
2833 \numgdef{\prev@abslineverse}{\the\absline@num-1}
2834 \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{\}
2835 \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesL<3\ledpbnum{\
prev@abslineverse}\fi}{\}
2836 \fi
2837 \fi
2838 \fi
2839 }
2840 \newcommand{\checkverseR}{
2841 \ifinstanzaR
2842 \iflednopbinverse
2843 \ifinserthangingsymbolR
2844 \numgdef{\prev@abslineverse}{\the\absline@numR-1}
2845 \IfStrEq{\led@pb@setting}{after}{\lednopbnumR{\prev@abslineverse}}{\}
2846 \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesR<3\ledpbnumR{\
prev@abslineverse}\fi}{\}
2847 \fi
2848 \fi
2849 \fi
2850 }
2851 %

```

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

**\ledthegoal**

**\goalfraction**

**\check@goal**

```

2852 \newdimen\ledthegoal
2853 \ifshiftedpstarts

```

```

2854 \newcommand*{\@goalfraction}{0.95}
2855 \else
2856 \newcommand*{\@goalfraction}{0.9}
2857 \fi
2858
2859 \newcommand*{\check@goal}{%
2860 \ledthegoal=\@goalfraction\pagegoal}
2861 \newcommand{\setgoalfraction}[1]{%
2862 \xdef\@goalfraction{#1}%
2863 }
2864 %

```

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

```

\ifwrittenlinesL
2865 \newif\ifwrittenlinesL
2866 \newif\ifwrittenlinesR
2867
2868 %

```

## XVII.8 Getting boxes content

`\if@getnextbox` The `\if@getnextbox` boolean is switched to true if we can get the next chunk in a page after finished previous chunk. That is:

- If we use the `nosyncpstarts` option, in any case
- If we do not use it, only when the number or real or blank line of the current chunk is equal or greater to the maximum number of line in the current pair of chunks.

```

2869 \newif\if@getnextbox%
2870 %

```

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

```

2871 \newcommand*{\get@nextboxL}{%
2872 \ifvbox\namebox{1@dLcolrawbox\the\l@dpscl}% box is not empty
2873 %

```

The current box is not empty; do nothing.

```

2874 \else% box is empty
2875 %

```

The box is empty. By default, we can get the next box

```

2876 \@getnextboxtrue%Should be local, but be cautious
2877 %

```

But not when sufficient lines for this page have been generated (except when we don't do any synchronization whatsoever). output.

```

2878 \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscl}>\@donetotallinesL
2879 \parledgroup@notes@endL%
2880 \unless\ifnosyncpstarts%
2881 \getnextboxfalse%
2882 %

```

If we use the nomaxlines option, we will start at new page, but we take count of the lines to be typeset for the actual right chunk on the right page, before starting new chunk on the left page.

```

2883 \ifnomaxlines%
2884 \ifdim\pagetotal<\ledthegoal%
2885 \numdef{\@tmp}{\l@dpscl+1}%
2886 \ifcsdef{afterlines@pstart@\@tmp R}{%
2887 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp R}}
2888 %
2889 {}%
2890 {\ifcsdef{minpage@pstart@\@tmp}%
2891 {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2892 {\ifnum\numpagelinesL=\l@dminpagelines%
2893 \getnextboxtrue%
2894 \fi%
2895 }%
2896 {\@getnextboxtrue}}%
2897 {\@getnextboxtrue}}%
2898 }%
2899 {}%
2900 \fi%
2901 \fi%
2902 \fi%
2903 \else%
2904 \ifnomaxlines%
2905 \numdef{\@tmp}{\the\l@dpscl+1}%
2906 \ifcsdef{minpage@pstart@\@tmp}{%
2907 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2908 {\ifdimgreater{\pagetotal}{\ledthegoal}%
2909 {\@getnextboxtrue}%
2910 {\@getnextboxfalse}%
2911 }%
2912 {\@getnextboxtrue}%
2913 }-}%
2914 \fi%
2915 \fi%
2916 %

```

Sufficient lines have been output.

```

2917 \if@getnextbox%
2918 \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscl}=\@donetotallinesL
2919 \parledgroup@notes@endL

```

```

2920 \fi
2921 \ifwrittenlinesL\else
2922 %

```

Write out the number of lines done, and set the boolean so this is only done once.

```

2923 \@writelinesinparL
2924 \writtenlinesLtrue
2925 \fi
2926 \ifnum\l@dnumstartL>\l@dpsclL
2927 %

```

There are still unprocessed boxes. Recalculate the maximum number of lines needed, and move onto the next box (by incrementing \l@dpscl). If needed, restart the line numbering.

```

2928 \writtenlinesLfalse
2929 \ifbypstart@
2930 \global\line@num=0%
2931 \resetprevline@%
2932 \fi
2933 % Add the content of the optional argument of the previous \protect\cs{pend
2934 }.
2935 % \begin{macrocode}
2936 \csuse{after@pendL@the\l@dpscl}%
2937 \global\csundef{after@pendL@the\l@dpscl}%
2938 %

```

Check the number of lines

```

2938 \l@dcalc@maxoftwo{\the\usernamecount{l@dmaxlinesinpar\the\l@dpscl}}%
2939 {\the\@donetotallinesL}%
2940 {\usernamecount{l@dmaxlinesinpar\the\l@dpscl}}}%
2941 \global\@donetotallinesL \z@
2942 %

```

Go to the next pstart

```

2943 \global\advance\l@dpscl \@ne
2944 \global\pstartnumtrue%
2945 \restore@pstartL@pc%
2946 %

```

Add notes of parallel ledgroup.

```

2947 \parledgroup@notes@endL
2948 \parledgroup@correction@notespadding@final{L}
2949 \else
2950 %

```

```

2951 \print@last@after@pendLtrue%
2952 \fi
2953 \fi
2954 \fi}
2955 %

```

```

2956 \newcommand*{\get@nextboxR}{%
2957   \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}% box is not empty
2958   \else% box is empty
2959     \@getnextboxtrue%
2960     \ifnum\usenamecount{1@dmaxlinesinpar\the\l@dpscR}>\@donetotallinesR
2961       \parledgroup@notes@endR
2962       \unless\ifnosyncpstarts%
2963         \@getnextboxfalse%
2964         \ifnomaxlines%
2965           \ifdim\pagetotal<\ledthegoal%
2966             \numdef{\@tmp}{\l@dpscR+1}%
2967             \ifcsdef{afterlines@pstart@\@tmp L}{%
2968               \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp L}}%
2969               {}%
2970               {\ifcsdef{minpage@pstart@\@tmp}%
2971                 {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2972                   {\ifnum\numpagelinesR=\l@dminpagelines%
2973                     \@getnextboxtrue%
2974                     \fi%
2975                     }%
2976                     {\@getnextboxtrue}}%
2977                     {\@getnextboxtrue}}%
2978                     }%%
2979                     }%
2980                     {}%
2981                     \fi%
2982                     \fi%
2983                     \fi%
2984                   \else%
2985                     \ifnomaxlines%
2986                     \numdef{\@tmp}{\the\l@dpscR+1}%
2987                     \ifcsdef{minpage@pstart@\@tmp}{%
2988                     \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2989                     {\ifdimgreater{\pagetotal}{\ledthegoal}%
2990                     {\@getnextboxtrue}%
2991                     {\@getnextboxfalse}%
2992                     }%
2993                     {\@getnextboxtrue}%
2994                     }{}
2995                     \fi%
2996                   \fi%
2997                 \if@getnextbox%
2998                   \ifnum\usenamecount{1@dmaxlinesinpar\the\l@dpscR}=\@donetotallinesR
2999                     \parledgroup@notes@endR
3000                     \fi
3001                     \ifwrittenlinesR\else
3002                     \@writelinesinparR
3003                     \writtenlinesRtrue
3004                     \fi

```



```

3005 \ifnum\l@dnumpstartsR>\l@dpscR
3006 \writtenlinesRfalse
3007 \ifbypstartR
3008 \global\line@numR=0%
3009 \resetprevline%
3010 \fi
3011 \csuse{after@pendR@the\l@dpscR}%
3012 \global\csundef{after@pendR@the\l@dpscR}%
3013 \l@dcalc@maxoftwo{\the\usernamecount{l@maxlinesinpar\the\l@dpscR}}%
3014 {\the\@donetotallinesR}%
3015 {\usernamecount{l@maxlinesinpar\the\l@dpscR}}}%
3016 \global\@donetotallinesR \z@
3017 \global\advance\l@dpscR \@ne
3018 \global\pstartnumRtrue%
3019 \restore@pstartR@pc%
3020 \parledgroup@notes@endR
3021 \parledgroup@correction@notes@spacing@final{R}
3022 \else
3023 \print@last@after@pendRtrue%
3024 \fi
3025 \fi
3026 \fi}
3027
3028 %

```

## XVIII Page numbering

### XVIII.1 Global options

The `sameparallelpagenu` option allows the same page number on both left and right side. The `prevpgnotnumbered` option allows an empty (not numbered) right-side page before `\Pages`.

We cannot implement these two options by changing the value of the page counter, since its value is used by many  $\TeX$  features to determine whether a page is left (even-numbered) or right (odd-numbered). Consequently, we have to do it by patching `\thepage`, in order to use the value of the `par@page` counter instead of value of page counter.

This counter will be increased in a patched version of the  $\TeX$ 's `\@outputpage` macro, as is the page counter in this macro. However, this increase will take account of the options.

`\par@patch@thepage` `\par@patch@thepage` patches `\thepage` in order to use the value of `par@page` counter and not the value of `par@page`. It must be called after any redefinition of `\thepage`. That is why we insert it at the end of the  $\TeX$  macro `\pagenumbering`, which is called by some `\xxxmatter` commands. In cases when we are using the memoir class, we insert it at the end of `\@mempnum`. When using `\pagenumbering`, we also need to restart `par@page` counter. Consequently, we have wrapped `\par@patch@thepage` and

counter restart in `\par@patch@pagenumbering`. We also call `\par@patch@thepage` it at the beginning of the document.

```

3029
3030 \newcommand{\par@patch@thepage}{%
3031   \ifboolexpr{%
3032     bool{sameparallelpagenumber}%
3033     or bool{prevpgnotnumbered}%
3034   }%
3035   {%
3036     \patchcmd{\thepage}%
3037       {page}{par@page}%
3038       {}%
3039       {\led@error@fail@patch@thepage}%
3040   }{}%
3041 }%
3042
3043 \newcommand{\par@patch@pagenumbering}{%
3044   \ifboolexpr{%
3045     bool{sameparallelpagenumber}%
3046     or bool{prevpgnotnumbered}%
3047   }%
3048   {%
3049     \setcounter{par@page}{1}%
3050   }%
3051   {}%
3052   \par@patch@thepage%
3053 }%
3054
3055 \ifl@dmemoir%
3056   \apptocmd{\@mempnum}%
3057     {\par@patch@pagenumbering}%
3058     {}%
3059     {\led@error@fail@patch@@mempnum}%
3060
3061 \else%
3062   \apptocmd{\pagenumbering}%
3063     {\par@patch@pagenumbering}%
3064     {}%
3065     {\led@error@fail@patch@pagenumbering}%
3066 \fi%
3067
3068 \AtBeginDocument{\par@patch@thepage}%
3069 %

```

`\@outputpage` As its name says, `\@outputpage` is a  $\text{\LaTeX}$ 's macro called in the output routine. It is this macro which increases the page counter.. We patch it in order to increase, conditionally, the `par@page` counter.

```

3070 \AtBeginDocument{%

```

```

3071 \apptocmd{\@outputpage}{%
3072   \ifsameparallelpagenumber%
3073     \ifl@dprintingpages%
3074       \ifodd\c@page\else%
3075         \stepcounter{par@page}%
3076       \fi%
3077     \else%
3078       \stepcounter{par@page}%
3079     \fi%
3080   \else%
3081     \stepcounter{par@page}%
3082   \fi%
3083 }%
3084 {}%
3085 {\led@error@fail@patch@\@outputpage}%
3086 }
3087 %

```

`\thepar@page` And now, initialize `par@page` counter.

```

3088 \newcounter{par@page}%
3089 \setcounter{par@page}{1}%
3090 %

```

## XVIII.2 *mainmatter* option of `\Pages`

The optional argument of `\Pages` could be equal to `mainmatter`. In this case the boolean `\ifPages@mainmatter` is set to true, and some special things are done in `\Pages@mainmatter`, called by `\cleartol@devenpage`.

```

\ifPages@mainmatter 3091 \newif\ifPages@mainmatter
\Pages@mainmatter 3092 \newcommand{\Pages@mainmatter}{%
3093   \ifPages@mainmatter%
3094     \pagenumbering{arabic}%
3095     \addtocounter{page}{1}%
3096     \addtocounter{par@page}{-1}%
3097     \patchcmd{\thepage}{page}{par@page}{}{}%
3098   \fi%
3099 }
3100 %

```

## XIX Sections' titles' commands

As switching from left to right pages does not clear the page since v1.13.0, but only creates new pages, no `\vbox{}` is inserted, and consequently parallel chapters are misaligned.

So we patch the `\chapter` command in order to prevent this problem.

```

\chapter 3101 \pretocmd{\chapter}{%
3102   \ifl@dprintingpages%
3103   \vbox{}%
3104   \fi%
3105   }%
3106   {}%
3107   {}%
3108   %

```

**\eledsectnotoc** \eledsectnotoc just saves its content \@eledsectnotoc, which will be tested where sectioning commands will be printed.

```

3109 \newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}}
3110 \eledsectnotoc{R}
3111 %

```

**\eledsectmark** \eledsectmark just saves its content \@eledsectmark, which will be tested where sectioning commands will be printed.

```

3112 \newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}}
3113 \eledsectmark{L}
3114 %

```

**\eledsection@correcting@skip** Because the vertical correction needed after inserting a title in parallel depends whether we are in parallel columns or parallel pages, we stock its length in \eledsection@correcting@skip.

```

3115 \newskip\eledsection@correcting@skip
3116 %

```

**\eled@sectioningR@out** We save the sectioning commands of the right side in the \eled@sectioningR@out file.

```

3117 \newwrite\eled@sectioningR@out
3118 %

```

## XX Page break/no page break, depending on the specific line

We need to adapt the macro of the homonym section of eledmac to eledpar.

**\prev@pbR** The \l@prev@pbR macro is a etoolbox's list, which contains the lines in which page breaks occur (before or after). The \l@prev@nopbR macro is a etoolbox list, which contains the lines in which NO page breaks occur (before or after).

```

3119 \def\l@prev@pbR{}
3120 \def\l@prev@nopbR{}
3121 %

```

`\ledpbR` The `\ledpbR` macro writes the call to `\led@pbR` in line-list file. The `\ledpbnumR` macro writes the call to `\led@pbnumR` in line-list file. The `\lednopbR` macro writes the call to `\led@nopbR` in line-list file. The `\lednopbnumR` macro writes the call to `\led@nopbnumR` in line-list file.

```

3122 \newcommand{\ledpbR}{\write\linenum@outR{\string\led@pbR}}
3123 \newcommand{\ledpbnumR}[1]{\write\linenum@outR{\string\led@pbnumR{#1}}}
3124 \newcommand{\lednopbR}{\write\linenum@outR{\string\led@nopbR}}
3125 \newcommand{\lednopbnumR}[1]{\write\linenum@outR{\string\led@nopbnumR{#1}}}
3126 %

```

`\led@pbR` The `\led@pbR` add the absolute line number in the `\prev@pbR` list. The `\led@pbnumR` add the argument in the `\prev@pbR` list. The `\led@nopbR` add the absolute line number in the `\prev@nopbR` list. The `\led@nopbnumR` add the argument in the `\prev@nopbR` list.

```

3127 \newcommand{\led@pbR}{\listxadd{\l@prev@pbR}{\the\absline@numR}}
3128 \newcommand{\led@pbnumR}[1]{\listxadd{\l@prev@pbR}{#1}}
3129 \newcommand{\led@nopbR}{\listxadd{\l@prev@nopbR}{\the\absline@numR}}
3130 \newcommand{\led@nopbnumR}[1]{\listxadd{\l@prev@nopbR}{#1}}
3131 %

```

## XXI Parallel ledgroup

`\parledgroup@` The marks `\parledgroup@` contains information about the beginnings and endings of notes in a parallel ledgroup. `\parledgroup@series` contains the footnote series. `\parledgroup@type` contains the type of the footnote: critical (Xfootnote) or familiar (footnoteX).

```

3132 \newmarks\parledgroup@
3133 \newmarks\parledgroup@series
3134 \newmarks\parledgroup@type
3135 %

```

`\parledgroup@notes@startL` `\parledgroup@notes@startL` and `\parledgroup@notes@startR` are used to mark the beginning of a note series in a parallel ledgroup.

```

3136 \newcommand{\parledgroup@notes@startL}{%
3137   \ifnum\usenamecount{1@dmxlinesinpar\the\l@dpscL}>0%
3138   \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
bhookeXnoteX@\splitfirstmarks\parledgroup@series}}{}%
3139   \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
bhookeXnote@\splitfirstmarks\parledgroup@series}}{}%
3140   \fi%
3141   \global\ledgroupnotesL@true%
3142   \insert@noterule@ledgroup{L}%
3143 }
3144 \newcommand{\parledgroup@notes@startR}{%
3145   \ifnum\usenamecount{1@dmxlinesinpar\the\l@dpscR}>0%

```

```

3146 \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
      bhooknoteX@\splitfirstmarks\parledgroup@series}}{ }%
3147 \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
      bhookXnote@\splitfirstmarks\parledgroup@series}}{ }%
3148 \fi%
3149 \global\ledgroupnotesR@true%
3150 \insert@noterule@ledgroup{R}%
3151 }
3152 %

```

`\parledgroup@notes@startL` `\parledgroup@notes@endL` and `\parledgroup@notes@endR` are used to mark the end of a note series in a parallel ledgroup.

```

3153 \newcommand{\parledgroup@notes@endL}{%
3154   \global\ledgroupnotesL@false%
3155 }
3156 \newcommand{\parledgroup@notes@endR}{%
3157   \global\ledgroupnotesR@false%
3158 }
3159 %

```

`\insert@noterule@ledgroup` A `\vskip` is not used when the boxes are constructed. So we insert it before ledgroup note series when parallel lines are constructed. This is the goal of `\insert@noterule@ledgroup`

```

3160 \newcommand{\insert@noterule@ledgroup}[1]{
3161   \IfStrEq{\splitbotmarks\parledgroup@}{begin}{%
3162     \IfStrEq{\splitbotmarks\parledgroup@type}{Xfootnote}{
3163       \csuse{ifledgroupnotes#1@}
3164       \vskip\skip\csuse{mp\splitbotmarks\parledgroup@series footins}
3165       \csuse{\splitbotmarks\parledgroup@series footnoterule}
3166       \fi
3167     }
3168   }
3169   \IfStrEq{\splitbotmarks\parledgroup@type}{footnoteX}{
3170     \csuse{ifledgroupnotes#1@}
3171     \vskip\skip\csuse{mpfootins\splitbotmarks\parledgroup@series}
3172     \csuse{footnoterule\splitbotmarks\parledgroup@series}
3173     \fi
3174   }{ }
3175 }
3176 {}
3177 }
3178 %

```

`\@parledgroupnotespadding` `\@parledgroupnotespadding` can be redefined by the user to change the interline spacing of ledgroup notes.

```

3179 \newcommand{\setparledgroupnotespadding}[1]{\gdef\@parledgroupnotespadding
      {#1}}
3180 \newcommand{\@parledgroupnotespadding}{ }
3181 %

```

`up@notespacing@correction` `\parledgroup@notespacing@correction` is the difference between a normal line skip and a line skip in a note. It is set by `\parledgroup@notespacing@set@correction`, called at the beginning of `\Pages`.

```

3182 \dimdef{\parledgroup@notespacing@correction}{0pt}
3183 \newcommand{\parledgroup@notespacing@set@correction}{%
3184   {\@getfirstseries\csuse{Xnotefontsize@\@firstseries}}%We suppose all the
series has the same footnote size setup
3185   \@parledgroupnotespacing\dimgdef{\temp@spacing}{\baselineskip}}%
3186   \dimgdef{\parledgroup@notespacing@correction}{\baselineskip-\temp@spacing}
}%
3187 }
3188 %

```

`rrrection@notespacing@init` `\parledgroup@correction@notespacing@init` sets the value of accumulated corrections of note spacing to 0 pt. It is called at the beginning of each pages AND at the end of each ledgroup.

```

3189 \newcommand{\parledgroup@correction@notespacing@init}{
3190   \dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}
3191   \dimdef{\parledgroup@notespacing@correction@modulo}{0pt}
3192 }
3193 \parledgroup@correction@notespacing@init
3194 %

```

`rection@notespacing@final` `\parledgroup@correction@notespacing@final` adds the total space deleted because of correction for notes, in a parallel ledgroup. It also adds the space needed by the other side spaces between note rules and notes. It is called after the print of each `pstart/pend`.

```

3195 \newcommand{\parledgroup@correction@notespacing@final}[1]{
3196   \ifparledgroup
3197     \vspace{\parledgroup@notespacing@correction@accumulated}
3198     \parledgroup@correction@notespacing@init%
3199     \ifstrequal{#1}{L}{
3200       \numdef{\@checking}{\the\l@dpscl-1}
3201     }{
3202       \numdef{\@checking}{\the\l@dpscr-1}
3203     }
3204     \dimdef{\@beforenotes@current@diff}{\csuse{@parledgroup@beforenotes@
@checking L}-\csuse{@parledgroup@beforenotes@
@checking R}}%
3205     \ifstrequal{#1}{L}%
3206     {% Left
3207       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{-\
@beforenotes@current@diff}}%
3208     }%
3209     {% Right
3210       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{\
@beforenotes@current@diff}}{}
3211     }%

```

```

3212 \fi
3213 }
3214 %

```

`\parledgroup@correction@notespacing` `\parledgroup@correction@notespacing` is used before each printed line. If it is a line of notes in parallel ledgroup, the space `\parledgroup@notespacing@correction` is decreased, to make interline space correct. The decreased space is added to `\parledgroup@notespacing` and `\parledgroup@notespacing@correction@modulo`. If `\parledgroup@notespacing@correction` is equal or greater than `\baselineskip`:

- It is decreased by `\baselineskip`.
- The total of line number in the current page is decreased by one.

For example, suppose an normal interline of 24 pt and interline for note of 12 pt. That means that the two lines of notes take the place of one normal line. For every two lines of notes, the line total for the current place is decreased by one.

```

3215 \newcommand{\parledgroup@correction@notespacing}[1]{%
3216   \csuse{ifledgroupnotes#1@}%
3217   \vspace{-\parledgroup@notespacing@correction}%
3218   \dimdef{\parledgroup@notespacing@correction@accumulated}{\
parledgroup@notespacing@correction@accumulated+\
parledgroup@notespacing@correction}%
3219   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo+\
parledgroup@notespacing@correction}%
3220   \ifdimless{\parledgroup@notespacing@correction@modulo}{\baselineskip}
3221   {\advance\numpagelinesL -\@ne%
   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo-\baselineskip}%
3222   }% mean greater than equal
3223   \fi%
3224 }
3225 %

```

`\parledgroup@beforenotesL` `\parledgroup@beforenotesL` and `\parledgroup@beforenotesR` store the total of space before notes in the current parallel ledgroup.

```

3226 \dimdef\parledgroup@beforenotesL{0pt}
3227 \dimdef\parledgroup@beforenotesR{0pt}
3228 %

```

`\parledgroup@beforenotes@save` The macro `\parledgroup@beforenotes@save` dumps the space before notes of the current parallel ledgroup in a macro named with the current pstart number.

```

3229 \newcommand{\parledgroup@beforenotes@save}[1]{
3230   \ifparledgroup
3231   \csdimdef{@parledgroup@beforenotes@\the\csuse{1@dnumpstarts#1}#1}{\
csuse{\parledgroup@beforenotes#1}}

```



```

3232 \csdimgdef{parledgroup@beforenotes#1}{0pt}
3233 \fi
3234 }
3235 %

```

## XXII Compatibility with eledmac

Here, we define some command for the eledmac-compat option.

```

3236 \ifeledmaccompat%
3237
3238
3239 \unless\ifnocritical@
3240 \let\onlyXside\Xonlyside
3241 \fi
3242 \fi
3243 %

```

## XXIII The End

</code>

## Appendix A Some things to do when changing version

### Appendix A.1 Migration to eledpar 1.4.3

Version 1.4.3 corrects a bug added in version 0.12, which made hanging verse always flush right, despite the value of the first element in the `\setstanzaindent` command.

However, if you want to return to automatic flushright margins for verses with hanging indents, you have to redefine the `\hangingsymbol` command.

```
\renewcommand{\hangingsymbol}{\protect\hfill}
```

See the following two examples:

With standard `\hangingsymbol`:

A very long verse should sometimes be hanging. The position of the hanging verse is fixed.

With the modification of the `hangingsymbol`:

A very long verse should sometimes be hanging. And we can see that a hanging verse is flush right.

### Appendix A.2 Migration from eledpar to reledpar

As for migration from eledmac to reledmac:

- One option has been removed because it is deprecated.
- Some of the customizations previously made by `\renewcommand` have been replaced with commands.
- Some command names have been changed in order to have a more logical and uniform pattern.

#### Appendix A.2.1 Deprecated options

The `shiftedverses` option has been removed. Use the general `shiftedpstart` option instead.

#### Appendix A.2.2 `\renewcommand` replaced with command

Many uses of `\renewcommand` have been replaced with uses of specific commands. Please read the handbook about these particular commands.

<i>Deprecated <code>\renewcommand</code></i>	<i>Replaced with</i>
<code>\goalfraction</code>	<code>\setgoalfraction</code>
<code>\parledgroupnotespacing</code>	<code>\setparledgroupnotespacing</code>
<code>\Rlineflag</code>	<code>\setRlineflag</code>

### Appendix A.2.3 Commands the names of which have changed

In order to ease the migration from eledpar to reledpar, you may load reledmac with eledmac-compat option. However, it is advised to change the command names.

<i>Old command</i>	<i>New command</i>
<code>\onlyXside</code>	<code>\Xonlyside</code>

## Appendix A.3 Migration to reledpar 2.2.0

The *astanza* can take now an option argument. Consequently, if the first line of verse in a *astanza* environment starts with brackets [], you must precede them with a `\relax`. If you do not do it, the content of the brackets will be considered as an optional argument of the *astanza* environment.

## Appendix A.4 Migration to reledpar 2.3.0

The line number style (alphabetic, numeric, etc.) for the notes of the right side are now defined by the value you set to `\linenumberstyleR` or `\linenumberstyle*`, and not by the value you set to `\linenumberstyle` which is kept for left side.

The same is true for sub-line number styles and `\sublinenumberstyleR` or `\sublinenumberstyle*`, which are distinct from `\sublinenumberstyle`.

Consequently, if you have changed line number representation in footnotes with `\linenumberstyle` and `\sublinenumberstyle`, check your settings for these control sequences.

## Appendix A.5 Migration to reledpar 2.4.0

We have fixed a bug which which misaligned left and right sides when a line contained a dotted letter.

We have tested and saw no problem with this correction, but if you see a difference in alignment between version 2.3.0 and 2.4.0, please contact us.

## Appendix A.6 Migration to reledpar 2.5.0

If you use either `\stanza` or *astanza* environment, please read Appendix A.12 p. 343.

## Appendix A.7 Migration to reledpar 2.6.0

`\printlinenumR` was deleted. Use `\Xlineflag` instead.

## Appendix A.8 Migration to reledpar 2.6.1

If you use `perpage` package to control footnote numbering, please read the handbook on 5.3.3 p. 13.

## 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 and Maïeul Rouquette. *eledmac A presumptuous attempt to port EDMAC, TABMAC and EDSTANZA to LaTeX*. December 2004. (Available from CTAN in macros/latex/contrib/eledmac)

## Index

### Symbols

<code>\@adv</code>	1
<code>\@astanza@line</code>	1
<code>\@cs@linesinparL</code>	1
<code>\@cs@linesinparR</code>	1
<code>\@cs@linesonpageL</code>	1
<code>\@cs@linesonpageR</code>	1
<code>\@donereallinesL</code>	1
<code>\@donereallinesR</code>	1
<code>\@donetotallinesL</code>	1
<code>\@donetotallinesR</code>	1
<code>\@eledsectionL</code>	1
<code>\@eledsectionR</code>	1
<code>\@lab</code>	1
<code>\@lopL</code>	1
<code>\@lopR</code>	1
<code>\@nl</code>	1
<code>\@nl@regR</code>	1
<code>\@outputpage</code>	1
<code>\@par@sync@option</code>	1
<code>\@par@this@sync@option</code>	1
<code>\@parledgroupnotespacing</code>	1
<code>\@pend</code>	1
<code>\@pendR</code>	1
<code>\@pstart</code>	1
<code>\@pstartR</code>	1
<code>\@pstartsfalse</code>	1
<code>\@pstartstrue</code>	1
<code>\@ref</code>	1
<code>\@ref@reg@parseR</code>	1
<code>\@ref@regR</code>	1
<code>\@set</code>	1
<code>\@stopastanza</code>	1

\@writelinesinparL	1
\@writelinesinparR	1
\@writelinesonpageL	1
\@writelinesonpageR	1
\@writepageofparL	1
\@writepageofparR	1
CLASSmemoir	113
COMMAND\+	49
COMMAND\@Rlineflag	78, 148
COMMAND\@adv	40, 147
COMMAND\@cs@linesinparL	104
COMMAND\@cs@linesonpageL	105
COMMAND\@eledsectionL	87
COMMAND\@eledsectionR	87
COMMAND\@eledsectmark	116
COMMAND\@eledsectnotoc	116
COMMAND\@goalfraction	12, 108
COMMAND\@l@dtmptcnta	71
COMMAND\@lab	38, 78, 147
COMMAND\@lopL	44, 99, 105
COMMAND\@lopR	44, 105
COMMAND\@mempnum	113
COMMAND\@namedef	27
COMMAND\@namuse	27
COMMAND\@nl	38, 45, 78, 147
COMMAND\@nl@regR	38
COMMAND\@outputpage	113, 114
COMMAND\@page	78
COMMAND\@par@sync@option	38
COMMAND\@parledgroupnotespacing	118
COMMAND\@pend	43
COMMAND\@pendR	43
COMMAND\@pstart	44
COMMAND\@pstartstrue	86
COMMAND\@ref	41–43, 45, 147
COMMAND\@ref@later	42
COMMAND\@ref@reg@parseargR	42
COMMAND\@ref@regR	42
COMMAND\@set	40, 147
COMMAND\@sw	42
COMMAND\AtBeginPairs	8, 52, 145, 149
COMMAND\AtEveryPend	145–147
COMMAND\AtEveryPstart	2, 18, 58, 89, 145–147
COMMAND\AtEveryPstartCall	2, 18, 58, 146
COMMAND\AtEveryStanza	149
COMMAND\AtEveryStopStanza	149
COMMAND\Clear the right lines for \@read@linelist	38
COMMAND\Columns	7, 8, 17, 18, 50, 88, 92, 142–144, 146, 148
COMMAND\Columns@print@after@pend	94
COMMAND\Columns@print@before@pstart	94

COMMAND\lcolwidth	8, 10, 96
COMMAND\Leftsidehook	142
COMMAND\Leftsidehookend	142
COMMAND\MakePerPage	13
COMMAND\Pages	4, 7, 10, 12, 17, 18, 49, 50, 69, 73, 75, 96, 102, 113, 115, 119, 142, 144, 146–148
COMMAND\Pages@mainmatter	115
COMMAND\Pairs	49
COMMAND\Rcolwidth	8, 10, 96
COMMAND\Rightsidehook	142
COMMAND\Rightsidehookend	142
COMMAND\Rlineflag	122
COMMAND\Xendlineflag	15, 148
COMMAND\Xlineflag	15, 123, 148
COMMAND\Xmaxhnotes	13
COMMAND\Xnoteswidthliketwocolumns	9, 145
COMMAND\Xonlyside	14, 73, 123, 149
COMMAND\Xtxtbeforenotes	149
COMMAND\&	18, 19
COMMAND\absline@numR	37
COMMAND\add@penalties	71
COMMAND\add@penaltiesL	71
COMMAND\advanceline	40, 46, 147
COMMAND\affixline@num	67
COMMAND\affixline@numR	67, 142, 143
COMMAND\affixpstart@num	69
COMMAND\affixpstart@numR	69
COMMAND\affixside@note	79
COMMAND\aftercolumnseparator	9, 92, 144
COMMAND\araw@textfalse	86
COMMAND\araw@texttrue	86
COMMAND\at@begin@pairs	52
COMMAND\autopar	18
COMMAND\ballast@count	71
COMMAND\baselineskip	91, 120
COMMAND\bbl@set@language	83, 147
COMMAND\beforecolumnseparator	9, 92, 144
COMMAND\begin	18, 19
COMMAND\beginnumbering	15–17, 28, 38, 92, 143, 144, 146
COMMAND\beginnumberingR	33, 45
COMMAND\bf	143
COMMAND\bfseries	143
COMMAND\brokenpenalty	71
COMMAND\chapter	115, 142
COMMAND\check@goal	108
COMMAND\check@pstarts	86
COMMAND\checkpageL	99, 106
COMMAND\checkpb@columns	91
COMMAND\checkpbL	107
COMMAND\checkpbR	107
COMMAND\checkraw@text	86

COMMAND\checkverseL	108
COMMAND\checkverseR	108
COMMAND\clear(double)page	103
COMMAND\clearl@dleftpage	104
COMMAND\clearl@drighthpage	104
COMMAND\clearpage	104, 146
COMMAND\cleartoevenpage	103
COMMAND\cleartol@devenpage	103, 115
COMMAND\columnrulewidth	8, 91
COMMAND\columns@position	92
COMMAND\columnseparator	8, 9
COMMAND\columnsposition	9, 144
COMMAND\correct@Xfootins@box	146
COMMAND\correct@footinsX@box	146
COMMAND\critext	145
COMMAND\csname	51
COMMAND\displaywidowpenalty	71
COMMAND\do@actions	65
COMMAND\do@actions@fixedcode	142
COMMAND\do@actions@nextR	65
COMMAND\do@actionsR	65, 142
COMMAND\do@ballast	71
COMMAND\do@ballastR	65
COMMAND\do@insidelineLhook	144
COMMAND\do@insidelineRhook	144
COMMAND\do@line	59
COMMAND\do@line(L/R)	62
COMMAND\do@lineL	59, 71, 142, 143
COMMAND\do@lineLhook	142
COMMAND\do@lineR	62, 142–144
COMMAND\do@lineRhook	142
COMMAND\do@lockoff	147
COMMAND\do@lockoffR	41
COMMAND\do@lockon	147
COMMAND\do@lockonR	40
COMMAND\doinsidelineLhook	145
COMMAND\doinsidelineRhook	145
COMMAND\dolineLhook	145
COMMAND\dolineRhook	145
COMMAND\edindex	145
COMMAND\edlabel	143, 145
COMMAND\edtext	15, 41, 45, 46, 76, 145
COMMAND\edtext@later	76, 78, 102
COMMAND\edtext@now	102
COMMAND\edtextlater	15, 76, 77, 149
COMMAND\edtextnow	15, 76–78, 149
COMMAND\eled@sectioningR@out	116
COMMAND\eledchapter	146
COMMAND\eledsection	145–147
COMMAND\eledsection@correcting@skip	116

COMMAND\eledsectmark	21, 116
COMMAND\eledsectnotoc	20, 116
COMMAND\eledxxx	145
COMMAND\end	18
COMMAND\endgraf	57
COMMAND\endlock	46, 147
COMMAND\endnumbering	9, 15, 17, 29, 146
COMMAND\endsub	46, 147
COMMAND\endumbering	16
COMMAND\expandafter	48
COMMAND\extensionchars	28
COMMAND\firstlinenum	16, 144, 147, 148
COMMAND\firstsublinenum	144, 147, 148
COMMAND\fix@page	39, 147
COMMAND\flag@end	45, 142, 145
COMMAND\flag@start	45, 145
COMMAND\flush@notesR	72
COMMAND\footnote	50
COMMAND\footnoteX	47, 48
COMMAND\footnoteX@reading	148
COMMAND\footnoteXmk	14
COMMAND\footnoteXnomk	14, 48
COMMAND\frontmatter	12, 21
COMMAND\get@nextboxL	143
COMMAND\get@nextboxR	143
COMMAND\getline@numL	64
COMMAND\getline@numR	64
COMMAND\getlinesfrompagelistL	105
COMMAND\getlinesfrompagelistR	105
COMMAND\getlinesfromparlistL	104
COMMAND\getlinesfromparlistR	104
COMMAND\gl@p	48
COMMAND\goalfraction	122
COMMAND\hangingsymbol	122, 142
COMMAND\hfill	90, 92
COMMAND\hidenumbering	16, 147
COMMAND\if@getnextbox	109
COMMAND\ifPages@mainmatter	115
COMMAND\ifbypage@	147
COMMAND\ifbypstart@R	147
COMMAND\ifdim	90
COMMAND\ifinserthangingsymbol	80
COMMAND\ifinserthangingsymbolR	80
COMMAND\ifl@dpagefull	106
COMMAND\ifl@dpaging	25, 145
COMMAND\ifl@dpairing	25, 142
COMMAND\ifl@dsamelang	145
COMMAND\ifl@dsamepage	106
COMMAND\ifl@pagefull	99
COMMAND\ifledRcol	25



COMMAND\iflinenumberLevenifblank	36
COMMAND\iflinenumberRevenifblank	36
COMMAND\iflledRcol	143
COMMAND\ifnumberedpar@	54
COMMAND\ifnumberingR	143
COMMAND\ifnumberpstart	50
COMMAND\ifpst@rtedL	28, 29, 55, 142
COMMAND\ifpst@rtedR	28
COMMAND\ifsublines@	40
COMMAND\insert@countR	41
COMMAND\insert@noterule@ledgroup	118
COMMAND\insert@notes@for@onlyside	75
COMMAND\insertlines@list	41
COMMAND\insertlines@listR	42
COMMAND\inserts@list	54
COMMAND\inserts@listR	71
COMMAND\l@d@nums	46
COMMAND\l@d@set	40, 46, 147
COMMAND\l@dLcolrawbox	54
COMMAND\l@dLcolrawbox1	85
COMMAND\l@dLcolrawbox2	85
COMMAND\l@dRcolrawbox	54
COMMAND\l@dcalc@maxoftwo	105
COMMAND\l@dcalc@minoftwo	105
COMMAND\l@dchecklang	142, 144
COMMAND\l@dcsnote	144
COMMAND\l@dleftbox	59, 99, 146
COMMAND\l@dlinenumR	36, 142
COMMAND\l@dlsnote	144
COMMAND\l@dmake@labels	78
COMMAND\l@dmaxlinesinpar	97
COMMAND\l@dmaxlinesinpar1	85
COMMAND\l@dminpagelines	97, 142
COMMAND\l@dnumpstartsL	85, 142
COMMAND\l@dprintingcolumnstrue	145
COMMAND\l@dprintingpagestrue	145
COMMAND\l@dpscL	88, 97, 98, 111
COMMAND\l@dpscR	88, 97, 98
COMMAND\l@drsnote	144
COMMAND\l@dsetupmaxlinecounts	85
COMMAND\l@duselanguage	83, 84, 142
COMMAND\l@dzeromaxlinecounts	85
COMMAND\l@prev@nopbR	116
COMMAND\l@prev@pbR	116
COMMAND\labelstarttrue	143
COMMAND\labelref@list	78
COMMAND\labelref@listR	78
COMMAND\lang	83
COMMAND\last@page@numR	39
COMMAND\led	143

COMMAND\led@nopbR	117
COMMAND\led@nopbnumR	117
COMMAND\led@pbR	117
COMMAND\led@pbnumR	117
COMMAND\ledinnerrote	19
COMMAND\ledleftnote	19
COMMAND\lednopb	19, 91, 107
COMMAND\lednopbR	117
COMMAND\lednopbnumR	117
COMMAND\ledouterote	19
COMMAND\ledpb	91, 107
COMMAND\ledpbR	117
COMMAND\ledpbnumR	117
COMMAND\ledrightnote	19
COMMAND\ledsidenote	19
COMMAND\ledstrutL	142
COMMAND\ledstrutR	142, 148
COMMAND\ledthegoal	108
COMMAND\ledtrutL	142, 148
COMMAND\leftlinenumR	36, 142
COMMAND\lemma	76, 77
COMMAND\let	48
COMMAND\line@list@R	42
COMMAND\line@list@stuff	38, 45
COMMAND\line@margin	33
COMMAND\line@marginR	33, 142
COMMAND\line@numR	37
COMMAND\lineation	17, 146
COMMAND\lineation*	17, 33, 145
COMMAND\lineationR	17, 32, 146, 149
COMMAND\linenum	76
COMMAND\linenum@out	78
COMMAND\linenum@outR	44
COMMAND\linenumberLevenifblanktrue	17, 149
COMMAND\linenumberRevenifblank	149
COMMAND\linenumberRevenifblanktrue	17
COMMAND\linenumberstyle	17, 123
COMMAND\linenumberstyle*	123
COMMAND\linenumberstyleR	17, 123
COMMAND\linenumincrement	16, 144, 147, 148
COMMAND\linenummargin	17, 33, 142, 147–149
COMMAND\linenummargin*	17, 34, 148
COMMAND\linenummarginR	17, 34, 148
COMMAND\linenumrepR	35, 142
COMMAND\linesinpar@listL	43, 104
COMMAND\linesonpage@listL	44, 105
COMMAND\lock@off	41
COMMAND\lock@on	40
COMMAND\mainmatter	2, 12, 21, 148
COMMAND\makeatletter	62

COMMAND\maxchunks	7, 18, 85, 86
COMMAND\maxdimen	73, 74
COMMAND\maxhnotesX	13
COMMAND\memorydump	16, 31
COMMAND\n@num	146
COMMAND\new@lineL	45
COMMAND\new@lineR	45
COMMAND\newhookcommand@series	49
COMMAND\newif	146
COMMAND\newpage	103, 104, 146
COMMAND\newseries	50
COMMAND\newseries@	46
COMMAND\newseries@par	46, 49, 50
COMMAND\noeledxxx	145
COMMAND\nomark@	48
COMMAND\nomaxlines	44
COMMAND\notesXwidthliketwocolumns	9, 145
COMMAND\num@lines	71
COMMAND\num@lines(R)	54
COMMAND\numberingR	30
COMMAND\numberlinefalse	7
COMMAND\numberonlyfirstinline	143
COMMAND\numberpstartfalse	16
COMMAND\numberpstarttrue	16, 143, 147
COMMAND\one@line	54, 80
COMMAND\one@lineR	54
COMMAND\onlyXside	123
COMMAND\onlysideX	14, 73, 75, 146, 149
COMMAND\otherlanguage	147
COMMAND\page@action	40, 147
COMMAND\pagenumbering	113, 147
COMMAND\pages	12
COMMAND\pagetotal	99, 146
COMMAND\par@line	71
COMMAND\par@line(R)	54
COMMAND\par@patch@pagenumbering	114
COMMAND\par@patch@thepage	113, 114
COMMAND\par@sync@option	23
COMMAND\parledgroup@	117
COMMAND\parledgroup@beforenotes@save	120
COMMAND\parledgroup@beforenotesL	120
COMMAND\parledgroup@beforenotesR	120
COMMAND\parledgroup@correction@notespacing	120
COMMAND\parledgroup@correction@notespacing@final	119
COMMAND\parledgroup@correction@notespacing@init	119
COMMAND\parledgroup@notes@endL	118
COMMAND\parledgroup@notes@endR	118
COMMAND\parledgroup@notes@startL	117
COMMAND\parledgroup@notes@startR	117
COMMAND\parledgroup@notespacing@correction	119, 120

COMMAND\parledgroup@notespacing@correction@accumulated	120
COMMAND\parledgroup@notespacing@correction@modulo	120
COMMAND\parledgroup@notespacing@set@correction	119
COMMAND\parledgroup@series	117
COMMAND\parledgroup@type	117
COMMAND\parledgroupnotespacing	122
COMMAND\parledgrouptrue	19
COMMAND\patchcmd	147
COMMAND\pausenumbering	9, 31, 149
COMMAND\pend	3, 7, 10, 17–19, 50, 54, 57–59, 84, 94, 144, 145, 147, 148
COMMAND\pendL	144
COMMAND\pendR	144
COMMAND\pends	18
COMMAND\perpage	13
COMMAND\prev@nopbR	117
COMMAND\prev@pbR	117
COMMAND\prevpgstyle	24
COMMAND\print@Xnotes	73
COMMAND\print@Xnotes@forpages	73, 74, 146
COMMAND\print@columnseparator	90, 144
COMMAND\print@eledsectionL	61
COMMAND\print@line	61
COMMAND\print@lineL	61
COMMAND\print@notesX@forpages	146
COMMAND\printlinenumR	123
COMMAND\printlinesR	148
COMMAND\pstart	3, 7, 10, 16–19, 32, 46, 50, 54, 55, 58, 84, 89, 94, 143, 144, 146–148
COMMAND\pstartL	58, 144
COMMAND\pstartR	58, 143, 144
COMMAND\pstartinfootnote	146
COMMAND\raw@text	84
COMMAND\read@linelist	38, 125, 147
COMMAND\ref@reg	41
COMMAND\ref@regR	41, 147
COMMAND\relax	123
COMMAND\reledmac	147
COMMAND\renewcommand	122
COMMAND\resumenumbering	9, 31, 144, 149
COMMAND\resumenumberingR	33, 145
COMMAND\rightlinenumR	36, 142
COMMAND\sameword	76
COMMAND\section	142
COMMAND\section@num	28
COMMAND\selectlanguage	18, 83, 84
COMMAND\set@continuousnumberingforR	33
COMMAND\set@line	46, 147
COMMAND\set@line@action	40, 147
COMMAND\setRlineflag	17, 122
COMMAND\setgoalfraction	12, 122
COMMAND\sethangingsymbol	19

COMMAND\setline	40, 46, 147
COMMAND\setlinenum	40, 46, 147
COMMAND\setnoteposition...	92
COMMAND\setparledgroupnotespacing	122, 147
COMMAND\setposition...	92
COMMAND\setstanzaindents	9, 19, 122
COMMAND\setwidth...	92
COMMAND\sidenotemargin	19, 144
COMMAND\sidenotemargin*	19, 144
COMMAND\skipnumbering	16, 146
COMMAND\sloppy	8
COMMAND\stanza	7, 9, 16, 18, 52, 81, 123, 143
COMMAND\stanzanumtrue	19
COMMAND\startlock	46, 147
COMMAND\startsub	46, 147
COMMAND\sub@action	40, 147
COMMAND\sub@off	78
COMMAND\sub@on	78
COMMAND\subline@numR	37
COMMAND\sublinenumberstyle	17, 123
COMMAND\sublinenumberstyle*	123
COMMAND\sublinenumberstyleR	17, 123
COMMAND\sublinenumincrement	144, 147, 148
COMMAND\sublinenumrepR	35, 142
COMMAND\syntaxonly	148
COMMAND\sza@0@	19
COMMAND\textheight	13
COMMAND\textwidth	52
COMMAND\the@labelX	148
COMMAND\theledlanguageL	83, 84
COMMAND\theledlanguageR	83, 84
COMMAND\thepage	21, 113
COMMAND\thepstartL	16, 143
COMMAND\thepstartR	16, 143
COMMAND\thestanzaL	19
COMMAND\thestanzaR	19
COMMAND\ vbox	56
COMMAND\vskip	118
COMMAND\vsplit	71
COMMAND\widthliketwocolumns	9
COMMAND\widthliketwocolumnsfalse	9
COMMAND\widthliketwocolumnstrue	9
COMMAND\xflagref	148
COMMAND\xright@appenditem	48
COMMAND\xspace	22
COMMAND\xxxfootstart	92
COMMAND\xxxmatter	113
ENVIRONMENTLeftside	52
ENVIRONMENTRightside	53
ENVIRONMENTstanza	18, 19, 81, 123, 148

ENVIRONMENTcolumns	30, 148
ENVIRONMENTledgroup	6, 148
ENVIRONMENTleft	17
ENVIRONMENTpages	30, 51, 148
ENVIRONMENTpairs	51, 148
PACKAGEEDMAC	124
PACKAGEEDSTANZA	124
PACKAGEEledmac	146
PACKAGEEledpar	146
PACKAGETABMAC	124
PACKAGEbabel	18, 82–84, 148
PACKAGEedmac	124
PACKAGEeledmac	4, 85, 121, 122, 124, 144, 147
PACKAGEeledpar	5, 6, 13, 34, 122, 123, 144, 145
PACKAGEetoolbox	90, 116
PACKAGEedmac	6
PACKAGEledpar	1, 6
PACKAGEMemoir	124
PACKAGEMusixtex	144
PACKAGEperpage	2, 13, 123, 148
PACKAGEpolyglossia	18, 82–84, 149
PACKAGEreledmac	1, 3, 5–7, 9, 13, 17–20, 22, 24, 25, 28, 29, 33, 34, 37–41, 43–50, 61, 69, 78, 80, 97, 103, 122, 123, 147–149
PACKAGEreledpar	1, 3, 5–7, 9–12, 18, 19, 21–24, 32, 37, 43, 44, 46, 47, 49, 50, 78, 122, 123, 147
PACKAGEsetspace	2, 20
PACKAGESyntonly	148
PACKAGEXkeyval	22
PACKAGEXr	149

## A

\absline@numR	1
\actionlines@listR	1
\actions@listR	1
\add@inserts@nextR	1
\add@insertsR	1
\add@penaltiesL	1
\add@penaltiesR	1
\advanceline	1
\affixline@numR	1
\affixpstart@numL	1
\affixpstart@numR	1
\affixside@noteR	1
\aftercolumnseparator	1, 9
\araw@textfalse	1
\araw@texttrue	1
astanza (environment)	18
\AtBeginPairs	1, 8
\AtEveryPstartCall	1
\autopar	18

**B**

\bbl@set@language	1
\beforecolumnseparator	1, 9
\beginnumbering	15
\beginnumberingR	1

**C**

\c@firstlinenumR	1
\c@firstsublinenumR	1
\c@linenumincrementR	1
\c@sublinenumincrementR	1
\ch@ck@l@ckR	1
\ch@cksub@l@ckR	1
\chapter	1
\chapterinpages	1
\check@goal	1
\check@pstarts	1
\checkpageL	1
\checkpageR	1
\checkpb@columns	1
\checkpbL	1
\checkpbR	1
\checkraw@text	1
\checkverseL	1
\checkverseR	1
\clearl@dleftpage	1
\clearl@drightpage	1
\cleartoevenpage	1
\cleartol@devenpage	1
\columnrulewidth	1, 8
\Columns	1, 8
\columns@position	1
\Columns@print@after@pend	1
\Columns@print@before@pstart	1
\columnseparator	1, 8
\columnspan	1, 9
\correct@footinsX@box	1
\correct@Xfootins@box	1
\countLline	1
\countRline	1
\critext	1

**D**

\do@actions@fixedcodeR	1
\do@actions@nextR	1
\do@actionsR	1
\do@ballastR	1
\do@insidelineLhook	1
\do@insidelineRhook	1
\do@lineL	1

<code>\do@lineLhook</code> .....	<u>1</u>
<code>\do@lineR</code> .....	<u>1</u>
<code>\do@lineRhook</code> .....	<u>1</u>
<code>\do@lockoff</code> .....	<u>1</u>
<code>\do@lockoffR</code> .....	<u>1</u>
<code>\do@lockon</code> .....	<u>1</u>
<code>\do@lockonR</code> .....	<u>1</u>
<code>\doinsidelineLhook</code> .....	<u>1</u>
<code>\doinsidelineRhook</code> .....	<u>1</u>
<code>\dolineLhook</code> .....	<u>1</u>
<code>\dolineRhook</code> .....	<u>1</u>
<code>\dump@pstartL@pc</code> .....	<u>1</u>
<code>\dump@pstartR@pc</code> .....	<u>1</u>

**E**

<code>\edlabel</code> .....	<u>1</u>
<code>\edtext</code> .....	<u>1</u>
<code>\edtext@later</code> .....	<u>1</u>
<code>\edtext@now</code> .....	<u>1</u>
<code>\edtextlater</code> .....	<u>1</u> , 15
<code>\edtextnow</code> .....	<u>1</u> , 15
<code>\eled@sectioningR@out</code> .....	<u>1</u>
<code>\eledsection@correcting@skip</code> .....	<u>1</u>
<code>\eledsectmark</code> .....	<u>1</u> , 21
<code>\eledsectnotoc</code> .....	<u>1</u> , 20
<code>\endlock</code> .....	<u>1</u>
<code>\endnumbering</code> .....	<u>1</u> , 15
<code>\endnumberingR</code> .....	<u>1</u>
<code>\endsub</code> .....	<u>1</u>
environments:	
<code>astanza</code> .....	18
<code>Leftside</code> .....	15
<code>pages</code> .....	9
<code>pairs</code> .....	8
<code>Rightside</code> .....	15

**F**

<code>\f@x@l@cksR</code> .....	<u>1</u>
<code>\finish@Pages@notes</code> .....	<u>1</u>
<code>\first@linenum@out@Rfalse</code> .....	<u>1</u>
<code>\first@linenum@out@Rtrue</code> .....	<u>1</u>
<code>\firstlinenum</code> .....	<u>1</u> , 16
<code>\firstlinenum*</code> .....	<u>1</u> , 17
<code>\firstlinenumR</code> .....	<u>1</u> , 17
<code>\firstsublinenum</code> .....	<u>1</u> , 16
<code>\firstsublinenum*</code> .....	<u>1</u> , 17
<code>\firstsublinenumR</code> .....	<u>1</u> , 17
<code>\fix@page</code> .....	<u>1</u>
<code>\flag@end</code> .....	<u>1</u>
<code>\flag@start</code> .....	<u>1</u>



\flush@notesR .....	1
\footnote@reading .....	1
\footnote@typeset .....	1
\footnoteXmk .....	14
\footnoteXnomk .....	14

## G

\get@familiarfootnote@number .....	1
\get@nextboxL .....	1
\get@nextboxR .....	1
\getline@numR .....	1
\getlinesfrompagelistL .....	1
\getlinesfrompagelistR .....	1
\getlinesfromparlistL .....	1
\getlinesfromparlistR .....	1
\goalfraction .....	1

## H

\hidenumbering .....	16
----------------------	----

## I

\if@getnextbox .....	1
\if@pstarts .....	1
\ifaraw@text .....	1
\iffirst@linenum@out@R .....	1
\ifinstanzaL .....	1
\ifinstanzaR .....	1
\ifl@dpagfull .....	1
\ifl@dpaging .....	1
\ifl@dpairing .....	1
\ifl@dsamepage .....	1
\ifl@dusedbabel .....	1
\ifledRcol .....	1
\iflinenumberLevenifblank .....	1
\iflinenumberRevenifblank .....	1
\ifnomaxlines .....	1
\ifnosyncpstarts .....	1
\ifPages@mainmatter .....	1
\ifprevpgnotnumbered .....	1
\ifprint@last@after@pendL .....	1
\ifprint@last@after@pendR .....	1
\ifpst@rtedL .....	1
\ifpst@rtedR .....	1
\ifpstartnumR .....	1
\ifsameparallelpagenumber .....	1
\ifshiftedpstarts .....	1
\ifwidthliketwocolumns .....	1
\ifwrittenlinesL .....	1
\init@series@par .....	1
\initnumbering@sectcountR .....	1

<code>\insert@countR</code> .....	1
<code>\insert@noterule@ledgroup</code> .....	1
<code>\insert@notes@for@onlyside</code> .....	1
<code>\inserthangingsymbolL</code> .....	1
<code>\inserthangingsymbolR</code> .....	1
<code>\insertlines@listR</code> .....	1
<code>\inserts@listR</code> .....	1

## L

<code>\l@d@set</code> .....	1
<code>\l@dc@maxchunks</code> .....	1
<code>\l@dcalc@maxoftwo</code> .....	1
<code>\l@dcalc@minoftwo</code> .....	1
<code>\l@dcalcnun</code> .....	1
<code>\l@dchecklang</code> .....	1
<code>\l@dleftbox</code> .....	1
<code>\l@dlinenumR</code> .....	1
<code>\l@dmake@labelsR</code> .....	1
<code>\l@dminpagelines</code> .....	1
<code>\l@dnumpstartsL</code> .....	1
<code>\l@dnumpstartsR</code> .....	1
<code>\l@dpagefullfalse</code> .....	1
<code>\l@dpagefulltrue</code> .....	1
<code>\l@drightbox</code> .....	1
<code>\l@dsamepagefalse</code> .....	1
<code>\l@dsamepagetrue</code> .....	1
<code>\l@dsetupmaxlinecounts</code> .....	1
<code>\l@dsetuprawboxes</code> .....	1
<code>\l@dskipversenumberR</code> .....	1
<code>\l@dusedbabelfalse</code> .....	1
<code>\l@dusedbabeltrue</code> .....	1
<code>\l@duselanguage</code> .....	1
<code>\l@dzeromaxlinecounts</code> .....	1
<code>\l@pscL</code> .....	1
<code>\l@pscR</code> .....	1
<code>\labelref@listR</code> .....	1
<code>\last@page@numR</code> .....	1
<code>\Lcolwidth</code> .....	1, 8, 10
<code>\led@err@BadLeftRightPstarts</code> .....	1
<code>\led@err@Columns@InsideEnv</code> .....	1
<code>\led@err@Columns@WithoutEnv</code> .....	1
<code>\led@err@LeftOnRightPage</code> .....	1
<code>\led@err@Leftside@PreviousNotPrinted</code> .....	1
<code>\led@err@Pages@InsideEnv</code> .....	1
<code>\led@err@Pages@WithoutEnv</code> .....	1
<code>\led@err@RightOnLeftPage</code> .....	1
<code>\led@err@Rightside@PreviousNotPrinted</code> .....	1
<code>\led@err@TooManyPstarts</code> .....	1
<code>\led@error@edtext@later@now</code> .....	1
<code>\led@error@fail@patch@memppnum</code> .....	1

<code>\led@error@fail@patch@@outputpage</code>	1
<code>\led@error@fail@patch@pagenumbering</code>	1
<code>\led@error@fail@patch@thepage</code>	1
<code>\led@error@missing@numbering</code>	1
<code>\led@error@note@called@onleftside</code>	1
<code>\led@error@note@called@onrightside</code>	1
<code>\led@nopbnumR</code>	1
<code>\led@nopbR</code>	1
<code>\led@pbnumR</code>	1
<code>\led@pbR</code>	1
<code>\led@warn@ChangeSyncOption</code>	1
<code>\led@warn@setting@in@rightside</code>	1
<code>\lednopbnum</code>	1
<code>\lednopbnumR</code>	1
<code>\ledpbnumR</code>	1
<code>\ledpbR</code>	1
<code>\ledstrutL</code>	1
<code>\ledstrutR</code>	1
<code>\ledthegoal</code>	1
<code>\leftlinenumR</code>	1
<code>\leftpstartnumL</code>	1
<code>\leftpstartnumR</code>	1
Leftside (environment)	15
<code>\Leftsidehook</code>	1
<code>\Leftsidehookend</code>	1
<code>\line@list@stuffR</code>	1
<code>\line@listR</code>	1
<code>\line@marginR</code>	1
<code>\line@numR</code>	1
<code>\lineation*</code>	1, 17
<code>\lineationR</code>	1, 17
<code>\linenum@outR</code>	1
<code>\linenumberLevenifblanktrue</code>	17
<code>\linenumberRevenifblanktrue</code>	17
<code>\linenumberstyle*</code>	1, 17
<code>\linenumberstyleR</code>	1, 17
<code>\linenumincrement</code>	1, 16
<code>\linenumincrement*</code>	1, 17
<code>\linenumincrementR</code>	1, 17
<code>\linenummargin</code>	1
<code>\linenummargin*</code>	1, 17
<code>\linenummarginR</code>	1, 17
<code>\linenumrepR</code>	1
<code>\linesinpar@listL</code>	1
<code>\linesinpar@listR</code>	1
<code>\list@clearing@regR</code>	1
<code>\list@pstartL@pc</code>	1
<code>\list@pstartR@pc</code>	1
<code>\lock@off</code>	1

## M

\maxchunks .....	1, 7
\maxlinesinpar@list .....	1
\memorydump .....	16
\memorydumpL .....	1
\memorydumpR .....	1

## N

\n@num .....	1
\namebox .....	1
\new@lineL .....	1
\new@lineR .....	1
\newnamebox .....	1
\newnamecount .....	1
\newseries@par .....	1
\notesXwidthliketwocolumns .....	9
\num@linesR .....	1
\numberpstartfalse .....	16
\numberpstarttrue .....	16
\numpagelinesL .....	1
\numpagelinesR .....	1

## O

\one@lineR .....	1
\onlysideX .....	14
optionadvancedshiftedpstarts .....	10, 11
optionauxdir .....	149
optioncontinuousnumberingwithcolumns .....	9, 149
optionnomaxlines .....	10, 11, 23
optionnosyncpstarts .....	12, 23, 109
optionshiftedpstarts .....	6, 11, 23
optionwidthliketwocolumns .....	9

## P

\page@action .....	1
\page@numR .....	1
\Pages .....	1, 10
pages (environment) .....	9
\Pages@mainmatter .....	1
pairs (environment) .....	8
\par@lineR .....	1
\par@patch@pagenumbering .....	1
\par@patch@thepage .....	1
\parledgroup@ .....	1
\parledgroup@beforenotes@save .....	1
\parledgroup@beforenotesL .....	1
\parledgroup@beforenotesR .....	1
\parledgroup@correction@notespacing .....	1
\parledgroup@correction@notespacing@final .....	1
\parledgroup@correction@notespacing@init .....	1

<code>\parledgroup@notes@startL</code> .....	1
<code>\parledgroup@notes@startR</code> .....	1
<code>\parledgroup@notes@spacing@correction</code> .....	1
<code>\parledgroup@notes@spacing@set@correction</code> .....	1
<code>\parledgroupseries@</code> .....	1
<code>\parledgrouptype@</code> .....	1
<code>\pausenumberingR</code> .....	1
<code>\pend</code> .....	17
<code>\pendL</code> .....	1
<code>\pendR</code> .....	1
<code>\prev@nopbR</code> .....	1
<code>\prev@pbR</code> .....	1
<code>\prevpgstyle</code> .....	1
<code>\print@columnseparator</code> .....	1
<code>\print@eledsectionL</code> .....	1
<code>\print@eledsectionR</code> .....	1
<code>\print@lineL</code> .....	1
<code>\print@lineR</code> .....	1
<code>\print@notesX@forpages</code> .....	1
<code>\print@Xnotes@forpages</code> .....	1
<code>\pstart</code> .....	17
<code>\pstartL</code> .....	1
<code>\pstartR</code> .....	1

## R

<code>\Rcolwidth</code> .....	1, 8, 10
<code>\read@linelist</code> .....	1
<code>\reledpar@error</code> .....	1
<code>\reledpar@warning</code> .....	1
<code>\restore@pstartL@pc</code> .....	1
<code>\restore@pstartR@pc</code> .....	1
<code>\resumenumberingR</code> .....	1
<code>\rightlinenumR</code> .....	1
<code>\rightpstartnumL</code> .....	1
<code>\rightpstartnumR</code> .....	1
<code>Rightside (environment)</code> .....	15
<code>\Rightsidehook</code> .....	1
<code>\Rightsidehookend</code> .....	1
<code>\Rlineflag</code> .....	1

## S

<code>\save@familiarfootnote@number</code> .....	1
<code>\save@section@number</code> .....	1
<code>\section@numR</code> .....	1
<code>\selectlanguage</code> .....	1
<code>\set@continuousnumberingforR</code> .....	1
<code>\set@line</code> .....	1
<code>\set@line@action</code> .....	1
<code>\set@sectcountR</code> .....	1
<code>\setgoalfraction</code> .....	1, 12

<code>\sethangingsymbol</code> .....	19
<code>\setline</code> .....	1
<code>\setlinenum</code> .....	1
<code>\setnamebox</code> .....	1
<code>\setnotepositionliketwocolumns@C</code> .....	1
<code>\setnotepositionliketwocolumns@L</code> .....	1
<code>\setnotepositionliketwocolumns@R</code> .....	1
<code>\setpositionliketwocolumns@C</code> .....	1
<code>\setpositionliketwocolumns@L</code> .....	1
<code>\setpositionliketwocolumns@R</code> .....	1
<code>\setRlineflag</code> .....	17
<code>\setwidthliketwocolumns@C</code> .....	1
<code>\setwidthliketwocolumns@L</code> .....	1
<code>\setwidthliketwocolumns@R</code> .....	1
<code>\sidenote@marginR</code> .....	1
<code>\sidenotemargin*</code> .....	1
<code>\skip@lockoff</code> .....	1
<code>\skipnumbering</code> .....	1, 16
<code>\startlock</code> .....	1
<code>\startsub</code> .....	1
<code>\sub@action</code> .....	1
<code>\subline@numR</code> .....	1
<code>\sublinenumberstyle*</code> .....	1, 17
<code>\sublinenumberstyleR</code> .....	1, 17
<code>\sublinenumincrement</code> .....	1, 16
<code>\sublinenumincrement*</code> .....	1, 17
<code>\sublinenumincrementR</code> .....	1, 17
<code>\sublinenumrepR</code> .....	1
<b>T</b>	
<code>\theledlanguageL</code> .....	1
<code>\theledlanguageR</code> .....	1
<code>\thepar@page</code> .....	1
<code>\thepstartL</code> .....	16
<code>\thepstartR</code> .....	16
<code>\thestanzaL</code> .....	1, 19
<code>\thestanzaR</code> .....	1, 19
<b>U</b>	
<code>\unhnamebox</code> .....	1
<code>\unvnamebox</code> .....	1
<code>\usenamecount</code> .....	1
<b>W</b>	
<code>\widthliketwocolumns</code> .....	9
<b>X</b>	
<code>\Xendlineflag</code> .....	15
<code>\Xlineflag</code> .....	15
<code>\Xnoteswidthliketwocolumns</code> .....	9
<code>\Xonlyside</code> .....	14

## Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
\Columns: Added \l@dchecklang and \l@duselanguage to \Columns	89
\Pages: Added \l@duselanguage to \Pages	98
General: Added section of babel related code	82
Fix babel problems	1
v0.3.0.	
\Pages: Added \ledstrutL to \Pages	98
Added \ledstrutR to \Pages	100
\Rightsidehookend: Added \Leftsidehook, \Leftsidehookend, \Rightsidehook and \Rightsidehookend	53
\affixline@numR: Changed \affixline@numR to match new eledmac	67
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR	65
\do@lineL: Added \do@lineLhook to \do@lineL	59
Simplified \do@lineL by using macros for some common code	59
\do@lineR: Changed \do@lineR similarly to \do@lineL	62
\flag@end: Removed extraneous spaces from \flag@end	45
\ifledRcol: Moved \ifl@dpairing to eledmac	25
\ifpst@rtedR: Moved \ifpst@rtedL to eledmac	28
\l@dlinenumR: Simplified \leftlinenumR and \rightlinenumR by introducing \l@dlinenumR	36
\l@dnumpstartsR: Moved \l@dnumpstartsL to eledmac	85
\ledstrutR: Added \ledtrutL and \ledstrutR	103
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR	35
General: Added \do@lineLhook and \do@lineRhook	62
Added hooks into Leftside environment	52
Reorganize for ledarab	1
v0.3.a.	
\line@marginR: Do not just set \line@marginR in \linenummargin	33
General: Minor \linenummargin fix	1
v0.3.b.	
\Pages: Added \l@dminpagelines calculation for succeeding page pairs	101
General: Improved parallel page balancing	1
v0.3.c.	
General: Compatibilty with Polyglossia	1
v0.4.0.	
General: No more ledparpatch. All patches are now in the main file.	1
v0.5.0.	
General: Corrections about \section and other titles in numbered sections	1
v0.6.0.	
General: Be able to us \chapter in parallel pages.	1
v0.7.0.	
General: Option ‘shiftedverses’ which make there is no blank between two parallel verses with inequal length.	1
v0.8.0.	
General: Possibility to have a symbol on each hanging of verses, like in the french typography. Redefine the commande \hangingsymbol to define the character.	1

v0.9.0.	
\ifledRcol: Moved \iflledRcol and \ifnumberingR to eledmac	25
General: Possibility to number \pstart.	16
Possibility to number the pstart with the commands \numberpstarttrue.	1
v0.9.1.	
General: The numbering of the pstarts restarts on each \beginnumbering.	1
v0.9.2.	
General: Debug : with \Columns, the hanging indentation now runs on the left columns and the hanging symbol is shown only when \stanza is used.	1
v0.9.3.	
General: \thepstartL and \thepstartR use now \bfseries and not \bf, which is deprecated and makes conflicts with memoir class.	1
v0.10.0.	
General: \edlabel commands on the right side are now correctly indicated.	1
\edlabel commands which start a paragraph are now put in the right place.	1
v0.11.0.	
\Columns: Line numbering by pstart.	90
\affixline@numR: Changed \affixline@numR to allow to disable line numbering (like in eledmac 0.15).	67
\get@nextboxR: Change \get@nextboxL and \get@nextboxR to allow to disable line numbering (like in eledmac 0.15).	109
Pstart number can be printed in side	111
\inserthangingsymbolR: Prevent the column separator for hanging verse from shifting	80
General: Change \do@lineL and \do@lineR to allow line numbering by pstart (like in eledmac 0.15).	59
Lineation can be by pstart (like in eledmac 0.15).	32
New management of hangingsymbol insertion, preventing undesirable insertions.	80
v0.12.0.	
General: New management of hangingsymbol insertion, preventing undesirable insertions.	80
v1.0.0.	
General: Compatibility with eledmac. Change name to eledpar.	1
Debug in lineation by pstart	32
v1.0.1.	
General: Correction on \numberonlyfirstinline with lineation by pstart or by page.	1
v1.1.0.	
\pstartR: Add \labelpstarttrue (from eledmac).	54
General: Shiftedverses becomes shiftedpstarts.	1
v1.1.1.	
\pstartR: Correct \pstartR bug introduced by 1.1.	54
v1.1.2.	
\affixside@noteR: Remove spurious space between line number and line content	79
v1.2.0.	
General: Support for \led<section> commands in parallel texts.	1
v1.2.1.	
\set@sectcountR: For the right section, the counter is defined only once.	30
v1.3.0.	
\edtext: Manage RTL language.	46
v1.3.2.	
General: Debug with some classes.	1



v1.3.3.	
General: Debugging the left notes of the right column. . . . .	79
v1.3.4.	
General: Allow use of commands in sidenotes, as introduced by eledmac 1.0. . . . .	79
v1.4.0.	
General: Added <code>\do@insidelineLhook</code> and <code>\do@insidelineRhook</code> . . . . .	62
v1.4.1.	
General: Enable the use of <code>stanzaindentsrepetition</code> within <code>astanza</code> environment. . . . .	81
v1.4.3.	
<code>\inserthangingsymbolR</code> : Hanging verse is no longer automatically flush right. . . . .	80
<code>\pendL</code> : Spurious spaces in <code>\pendL</code> . . . . .	57
<code>\pendR</code> : Spurious spaces in <code>\pstartR</code> . . . . .	58
<code>\pstartR</code> : Spurious spaces in <code>\pstartL</code> and <code>\pstartR</code> . . . . .	54
General: Corrects a false hanging verse when a verse is exactly the length of a line. . . . .	1
v1.5.0.	
<code>\sublinenumincrement*</code> : Add starred version of <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> to change both Left and Rightside. . . . .	34
General: Add, as in eledmac, features to manage page breaks. . . . .	1
v1.6.0.	
General: Add tool and documentation for parallel ledgroups . . . . .	19
v1.7.0.	
General: Add, as in eledmac, features to make crossrefs with <code>pstart</code> numbers. . . . .	1
v1.8.0.	
<code>\Columns</code> : Modify <code>\Columns</code> to enable to add section's title. . . . .	88
Suppress <code>\l@dchecklang</code> from <code>\Columns</code> . . . . .	89
<code>\Pages</code> : Modify <code>\Pages</code> to enable to add section's title. . . . .	96
<code>\l@dchecklang</code> : Suppress <code>\l@dchecklang</code> which did not work and was not logical, because both columns could have the same language but not the main language of the document. . . . .	83
<code>\pendL</code> : As in eledmac, <code>\pendL</code> can have an optional argument. . . . .	57
<code>\pendR</code> : As in eledmac, <code>\pendR</code> can have an optional argument. . . . .	58
<code>\print@columnseparator</code> : Move some code of <code>\Columns</code> to <code>\print@columnseparator</code> . . . . .	90
<code>\pstartR</code> : As in eledmac, <code>\pendL</code> and <code>\pendR</code> can have an optional argument. . . . .	54
<code>\sidenotemargin*</code> : <code>\sidenotemargin</code> is now directly defined in eledmac to be able to manage eledpar. . . . .	79
Add <code>\sidenotemargin*</code> . . . . .	79
<code>\theledlanguageR</code> : Correct left/right language setting with polyglossia. . . . .	84
General: <code>\beginnumbering</code> is defined only on eledmac, not on eledpar. . . . .	28
<code>\l@dlsnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in eledmac. . . . .	79
Add <code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code> . . . . .	9
Add <code>\columnspostion</code> . . . . .	9
Add, as in eledmac, new system of sectioning commands. . . . .	1
Add, as in eledmac, option to insert something after <code>\pends</code> / verses. . . . .	1
Add, as in eledmac, option to insert something between <code>\pstarts</code> / verse. . . . .	1
Change <code>\do@lineR</code> and <code>\do@lineR</code> to allow new sectioning commands. . . . .	59
Compatibility with <code>musixtex</code> . . . . .	1
Debug eledmac sectioning command after using <code>\resumenumbering</code> . . . . .	1
New sectioning commands, as in eledmac. . . . .	20

Suppress <code>\ifl@dsamelang</code> which did not work and was not logical, because both columns could have the same language but not the main language of the document.	83
v1.8.1.	
<code>\do@lineL</code> : Fix a bug with critical notes at the beginning of a page, (maybe added by v1.8.0) (?).	59
<code>\do@lineR</code> : Fix a bug with critical notes at the beginning of a page, added by v1.8.0 (?).	62
v1.8.2.	
<code>\flag@end</code> : <code>\flag@start</code> and <code>\flag@end</code> are now defined only one time for <code>eledmac</code> and <code>eledpar</code>	45
<code>\lineation*</code> : Add <code>\lineation*</code>	33
<code>\reledpar@error</code> : Errors specific to <code>eledpar</code> send to <code>eledpar</code> handbook	25
General: Debug <code>\eledxxx</code> with some paper sizes	1
Debug left and side note (bugs added by 1.8.0)	1
v1.8.3.	
<code>\Pages</code> : Debug blank pages when using optional argument in the last <code>\pend</code> .	96
<code>\doinsidelineRhook</code> : Added <code>\dolineLhook</code> , <code>\dolineRhook</code> , <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code>	62
<code>\resumenumberingR</code> : Debug <code>\resumenumberingR</code>	31
General: Add <code>\noeledxxx</code> , as in <code>eledmac</code>	1
v1.9.0.	
<code>\ifwidthliketwocolumns</code> : Added <code>widthliketwocolumns</code> option	24
<code>\theledlanguageR</code> : Debug left/right language switching with <code>polyglossia</code> . Do not write in .aux file when setting left/right lines.	84
General: Add <code>\AtBeginPairs</code> macro.	8
Compatibility with <code>\Xnoteswidthliketwocolumns</code> and <code>\notesXwidthliketwocolumns</code>	1
v1.9.1.	
<code>\ifledRcol</code> : Moved <code>\ifl@dpaging</code> to <code>eledmac</code>	25
v1.10.0.	
<code>\Pages</code> : Debug wrong pages splitting when no optional argument is used in last <code>\pend</code> (bug was added in v1.8.3).	96
Debug wrong parallel pages synchronization when an <code>\edtext</code> falls across two pages.	96
General: Compatibility with <code>\AtEveryPstart</code> and <code>\AtEveryPend</code>	1
Restore critical notes in <code>\eledsection</code> in parallel columns (this bug was added in 1.8.2).	1
v1.10.1.	
<code>\line@list@stuffR</code> : Revert modification of 1.4.2, which makes bugs with numbering. Leave vertical mode to solve spurious space before <code>minipage</code> .	45
v1.11.0.	
<code>\edtext</code> : <code>\critext</code> and <code>\edtext</code> are now defined only in <code>eledmac</code> .	46
General: Compatibility of standard footnotes with some <code>biblatex</code> styles.	1
v1.12.0.	
<code>\Columns</code> : Add <code>\l@dprintingcolumnstrue</code>	88
<code>\Pages</code> : Add <code>\l@dprintingpagestrue</code>	96
<code>\edlabel</code> : <code>\edlabel</code> and <code>\edindex</code> works now with <code>hyperref</code> when using <code>eledpar</code> .	78
<code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code>	78
<code>\print@eledsectionL</code> : Compatibility with <code>Lua<math>\TeX</math></code> RTL languages.	61
<code>\print@eledsectionR</code> : Compatibility with <code>Lua<math>\TeX</math></code> RTL languages.	64
<code>\print@lineL</code> : Compatibility with <code>Lua<math>\TeX</math></code> RTL languages.	61
General: Compatibility with <code>Lua<math>\TeX</math></code> RTL languages.	1

v1.12.1.	
\print@eledsectionL: Fixes bug with Lua $\TeX$ RTL \eledsection. . . . .	61
v1.13.0.	
\Pages: Prevent false overfull hboxes when using \Pages outside of pages environment. . . . .	96
When using shiftedpstarts option, a \l@dleftbox with a null height will advance the	
\pagetotal in any case. . . . .	96
\clearl@drighthouse: Use \newpage instead of \clearpage. . . . .	104
\ifledRcol: Remove false boolean settings which are not needed. . . . .	25
General: Enable the use of optional argument of & in astanza environment. . . . .	81
Fix bug in shiftedpstarts when size difference between pstarts is very important. . . . .	1
With parallel pages, long notes can now flow from the Left to the right side and from	
the Right to the left side. . . . .	1
v1.13.1.	
\Pages: Prevent false empty page after \Pages (bug added in 1.13.0) . . . . .	96
\correct@footinsX@box: Call \correct@footinsX@box and	
\correct@Xfootins@box directly in \print@notesX@forpages and	
\print@Xnotes@forpages. . . . .	73
Correct \correct@footinsX@box and \correct@Xfootins@box . . . . .	73
v1.14.0.	
General: Fix bug with line number position when using \eledsection and similar	
commands for RTL texts with Lua $\TeX$ . . . . .	1
The \newifs are not followed by boolean values set to false, because it is the $\TeX$	
default setting. . . . .	1
v1.15.0.	
\do@actions@nextR: Add actions 1008 and 1009 . . . . .	65
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting	
when a verse is hanging . . . . .	80
\lineationR: As \lineation, \lineationR automatically set the	
\pstartinfootnote. . . . .	32
\n@num: \n@num defined only one time for both Eledmac and Eledpar. . . . .	41
\skipnumbering: \skipnumbering defined only one time for both Eledmac and	
Eledpar . . . . .	46
General: Add \AtEveryPstartCall. . . . .	1
Add sameparallelpagenumber option. . . . .	12
Fix vertical spurious space before right \eledchapter (bug added in v1.13.0). . . . .	1
Prevent vertical space when using \AtEveryPstart or \AtEveryPend with a	
command which prints nothing . . . . .	1
v1.16.0.	
\newseries@par: Fix bug with \onlysideX. . . . .	47
General: Error message when calling \Pages inside 'pages' environment and \Columns	
inside 'pairs' environment. . . . .	1
Error message when starting a Leftside/a Rightside while the previous one has not	
been yet typeset. . . . .	1
Error message when using \beginnumbering... \endnumbering without \pstart. . . . .	1
Fix bug with nofamiliar / nocritical option of eledmac. . . . .	1
New package option sameparallelpagenumber to have the same page number for both	
left and right side. . . . .	1
v1.16.1.	
General: Write information about line-list file version in the correct file. . . . .	1

v1.16.2.	
General: Fix bug when adding empty lines before a <code>\pend</code> in combination with some specific penalties setting. . . . .	1
v1.17.0.	
General: Add compatibility of optional argument of <code>\pstart/\pend</code> and <code>\AtEveryPstart/\AtEveryPend</code> with two columns mode. . . . .	1
v1.21.0.	
General: Add <code>\hidenummering</code> . . . . .	16
v2.0.0.	
<code>\@adv</code> : <code>\@adv</code> defined only in <code>reledmac</code> . . . . .	40
<code>\@lab</code> : <code>\@lab</code> defined only in <code>eledmac</code> . . . . .	78
<code>\@ref@regR</code> : <code>\@ref</code> defined only in <code>reledmac</code> , code specific to right side moved in <code>\ref@regR</code> . . . . .	41
<code>\@set</code> : <code>\@set</code> defined only in <code>reledmac</code> . . . . .	40
<code>\advanceline</code> : <code>\advanceline</code> defined only in <code>reledmac</code> . . . . .	46
<code>\bbl@set@language</code> : Patch <code>\bbl@set@language</code> instead of redefining it . . . . .	83
<code>\do@lockonR</code> : <code>\do@lockon</code> defined only in <code>reledmac</code> . . . . .	40
<code>\endlock</code> : <code>\startlock</code> and <code>\endlock</code> defined only in <code>reledmac</code> . . . . .	46
<code>\endsub</code> : <code>\startsub</code> and <code>\endsub</code> defined only in <code>reledmac</code> . . . . .	46
<code>\fix@page</code> : <code>\fix@page</code> is defined only once in <code>reledmac</code> . . . . .	39
<code>\l@d@set</code> : <code>\l@d@set</code> defined only in <code>reledmac</code> . . . . .	40
<code>\line@marginR</code> : <code>\linenummargin</code> now defined only once time in <code>reledmac</code> . . . . .	33
<code>\page@action</code> : <code>\page@action</code> defined only in <code>reledmac</code> . . . . .	40
<code>\read@linelist</code> : <code>\read@linelist</code> is defined only once time in <code>\reledmac</code> . . . . .	38
<code>\set@line</code> : <code>\set@line</code> defined only in <code>reledmac</code> . . . . .	46
<code>\set@line@action</code> : <code>\set@line@action</code> defined only in <code>reledmac</code> . . . . .	40
<code>\setline</code> : <code>\setline</code> defined only in <code>reledmac</code> . . . . .	46
<code>\setlinenum</code> : <code>\setlinenum</code> defined only in <code>reledmac</code> . . . . .	46
<code>\skip@lockoff</code> : <code>\do@lockoff</code> defined only in <code>reledmac</code> . . . . .	41
<code>\sub@action</code> : <code>\sub@action</code> defined only in <code>reledmac</code> . . . . .	40
<code>\sublinenumincrement*</code> : <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> are now defined only in <code>reledmac</code> . . . . .	34
<code>\theledlanguageR</code> : Patch <code>\otherlanguage</code> instead of redefining it. . . . .	84
General: <code>\@nl</code> is now defined only in <code>reledmac</code> . . . . .	38
<code>\ifbypage@</code> and <code>\ifbypstart@R</code> defined in <code>eledmac</code> . . . . .	32
Fix some bugs with ‘sameparallelpagenumber’ option. . . . .	1
Many code refactored and moved to <code>reledmac</code> . . . . .	1
Package’s name becomes <code>reledpar</code> . . . . .	1
Totally new implementation of ‘sameparallelpagenumber’ option. . . . .	1
<code>chapterinpages</code> : Deleting the old system of managing parallel chapter, keep only the new one with <code>\patchcmd</code> . . . . .	52
v2.1.0.	
General: Fix bug when using <code>\eledsection</code> and related on right pages when page width is short. . . . .	1
Fix bug when using <code>\pagenumbering</code> with <code>memoir</code> (bug added in v2.0.0). . . . .	1
Fix bug with <code>\setparallelgroupnotespacing</code> with the <code>shiftedpstarts</code> option. . . . .	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code> . . . . .	1
Options to custom empty right page before <code>\Pages</code> . . . . .	1

v2.2.0.	
General: <code>astanza</code> environment can take an optional argument, which will be the optional argument of <code>\pstart</code> started by this environment. . . . .	1
New tools to number stanza . . . . .	1
v2.2.1.	
General: Fix bug with optional argument of last left <code>\pend</code> . . . . .	1
v2.3.0.	
<code>\Pages</code> : Fix bug when calling <code>\Columns</code> after a <code>\Pages</code> (bug added in v1.13.0). . . . .	96
General: Change some internal codes in order to provide compatibility with $\text{\LaTeX}$ release of october 2015 . . . . .	1
Fix bug with title number in parallel columns . . . . .	1
New line setting command suffixed by R to set only the right side. . . . .	1
v2.4.0.	
<code>\ledstrutR</code> : Deleted <code>\ledstrutL</code> and <code>\ledstrutR</code> . . . . .	103
Fix bug with dotted letter . . . . .	103
General: New way of (not) synchronizing the parallel pages. . . . .	1
Option to switch to <code>\mainmatter</code> when calling <code>\Pages</code> . . . . .	1
v2.5.0.	
General: Disable empty lines as paragraph in <code>astanza</code> . . . . .	1
Fix bug introduced in v1.15.0 which made hanging indentation in verse not work anymore. . . . .	1
New commands <code>\linenummarginR</code> and <code>\linenummargin*</code> . . . . .	1
v2.5.1.	
General: Fix spurious space when using optional argument of <code>astanza</code> environment (introduced in v2.5.0). . . . .	1
v2.5.2.	
General: Fix bug introduced in v2.5.0 with <code>\linenummargin</code> , <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> . . . . .	1
v2.6.0.	
<code>\l@dmake@labelsR</code> : <code>\@Rlineflag</code> is not stored directly after the line number, but as a fifth argument of <code>\the@labelX</code> . Can be retrieved by <code>\xflagref</code> . . . . .	78
General: <code>\Xlineflag</code> and <code>\Xendlineflag</code> added . . . . .	1
<code>\printlinesR</code> deleted . . . . .	1
Error message when calling <code>\Pages</code> or <code>\Columns</code> without previous pages or pairs environment. . . . .	1
Fix bug with footnote numbering when using the same series of familiar footnotes on both sides. . . . .	1
Fix bug with right side title number when using title commands before pages or columns environments. . . . .	1
Fix compatibility with <code>babel</code> (broken in v2.0.0). . . . .	1
No error messages about ends of left / right page when using the <code>\syntaxonly</code> command of the <code>syntonly</code> package. . . . .	1
v2.6.1.	
General: Fix bug, introduced in v2.6.0, with footnote numbering when using <code>perpage</code> package. . . . .	1
v2.6.2.	
<code>\newseries@par</code> : The $\text{\TeX}$ counter <code>\footnoteX@reading</code> is defined in <code>reledmac</code> . . .	47
General: Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.6.0). . . . .	1

Fix bug (added in v2.6.0) with footnote numbering in parallel typesetting while using polyglossia with specific numbering systems (like Greek). . . . .	1
v2.6.3.	
General: Fix spurious dot when using \linenummargin on right side (introduced in v2.5.0). . . . .	1
v2.7.0.	
General: reledmac cross-referencing can take advantage of xr package. . . . .	1
v2.7.1.	
General: Fix bug added in reledmac 2.8.2, when typesetting parallel text just after a sectioning command . . . . .	1
v2.8.0.	
General: Allow continuing line numbering between normal text and parallel text, using \pausenumbering and \resumenumbering and the continuousnumberingwithcolumns options. . . . .	1
Add \linenumberLevenifblanktrue and \linenumberRevenifblank commands . . . . .	1
Fix bug when the right line number style is not the same to the left line number style . . . . .	1
v2.9.0.	
General: Add \AtEveryStanza and \AtEveryStopStanza. . . . .	1
More specific error messages. . . . .	1
v2.9.1.	
General: Prevent \Xtxtbeforenotes hook from causing notes to go beyond the bottom margin . . . . .	1
v2.10.0.	
\do@actions@nextR: Add action 1010 . . . . .	65
General: Add new tools to make apparatuses of manuscripts . . . . .	1
v2.11.0.	
\correct@footinsX@box: Clarification in the handbook about the use of \Xonlyside and onlysideX and error message if misuse. . . . .	73
New implementation of \Xonlyside and \onlysideX hooks, prevent trouble with vertical spacing. . . . .	73
General: Compatibility with reledmac's auxdir option. . . . .	1
v2.12.0.	
General: Add \edtextlater and \edtextnow. . . . .	1, 76
Fix bug with \AtBeginPairs. . . . .	1
Fix bug with \lineationR. . . . .	1