

# 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](https://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 v2.11.0, last revised 2016/06/16.

<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 <code>\Pages</code>	12
5.2.6 Notes about <code>\mainmatter</code>	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 <code>perpage</code> 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 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 <code>\AtEveryPstart</code> and <code>\AtEveryPstartCall</code>	17
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 <code>setspace</code> package	20
<b>10 Sectioning commands</b>	<b>20</b>
<b>11 Notes about page number</b>	<b>20</b>
<b>I Implementation overview</b>	<b>21</b>

<b>II Preliminaries</b>	<b>21</b>
II.1 Package's meta-data . . . . .	21
II.2 Package's requirement . . . . .	21
II.3 Package's options . . . . .	21
II.4 Package's options . . . . .	22
II.4.1 Synchronization's options . . . . .	22
II.4.2 Other options . . . . .	23
II.5 Determining side and category of parallel processing . . . . .	23
II.6 Text's width . . . . .	24
II.7 Messages . . . . .	24
II.8 Naming macros . . . . .	26
<b>III Sectioning commands</b>	<b>27</b>
<b>IV Line counting</b>	<b>31</b>
IV.1 Setting lineation reset . . . . .	31
IV.2 Setting line number margin . . . . .	32
IV.3 Setting lineation start and step . . . . .	33
IV.4 Setting line flag . . . . .	34
IV.5 Setting line number style . . . . .	34
IV.6 Print marginal line number . . . . .	35
IV.7 Line-number counters and lists . . . . .	35
IV.7.1 Correspond to those in <code>reledmac</code> for regular or left text . . . . .	35
IV.7.2 Specific to <code>reledpar</code> . . . . .	36
IV.8 Reading the line-list file . . . . .	36
IV.9 Commands within the line-list file . . . . .	37
IV.10 Writing to the line-list file . . . . .	43
<b>V Marking text for notes</b>	<b>45</b>
V.1 Specific hooks and commands for notes . . . . .	45
V.1.1 Notes to be printed on one side only . . . . .	45
V.2 Tools specific to familiar footnotes . . . . .	45
V.2.1 Managing correct number . . . . .	46
V.2.2 Familiar footnotes without marks . . . . .	46
V.2.3 Get correct footnote number . . . . .	48
V.3 Create hooks . . . . .	48
V.4 Init standards series (A,B,C,D,E,Z) . . . . .	48
V.5 Tools specific to $\LaTeX$ 's classical footnotes . . . . .	49
<b>VI Pstart numbers dumping and restoration</b>	<b>49</b>
<b>VII Parallel environments</b>	<b>50</b>

<b>VIII Paragraph decomposition and reassembly</b>	<b>52</b>
VIII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code> . . . . .	53
VIII.2 Processing one line . . . . .	57
VIII.3 Line and page number computation . . . . .	62
VIII.4 Line number printing . . . . .	65
VIII.5 Pstart number printing in side . . . . .	68
VIII.6 Add insertions to the vertical list . . . . .	69
VIII.7 Penalties . . . . .	70
VIII.8 Printing leftover notes . . . . .	71
<b>IX Footnotes</b>	<b>71</b>
IX.1 Footnotes output specific to <code>\Pages</code> . . . . .	71
<b>X Cross referencing</b>	<b>74</b>
<b>XI Side notes</b>	<b>75</b>
<b>XII Familiar footnotes</b>	<b>76</b>
<b>XIII Verse</b>	<b>77</b>
<b>XIV Fixing babel and polyglossia</b>	<b>79</b>
<b>XV Counts and boxes for parallel texts</b>	<b>81</b>
<b>XVI Checking text to be processed</b>	<b>83</b>
<b>XVII Parallel columns</b>	<b>84</b>
<b>XVIII Parallel pages</b>	<b>92</b>
XVIII.1 Specific counters . . . . .	92
XVIII.2 Main macro . . . . .	92
XVIII.3 Ensure all notes be printed at the end of parallel pages . . . . .	99
XVIII.4 Struts . . . . .	99
XVIII.5 Page clearing . . . . .	100
XVIII.6 Lines managing . . . . .	101
XVIII.7 Page break managing . . . . .	102
XVIII.8 Getting boxes content . . . . .	105
<b>XIX Page numbering</b>	<b>109</b>
XIX.1 Global options . . . . .	109
XIX.2 mainmatter option of <code>\Pages</code> . . . . .	111
<b>XX Sections' titles' commands</b>	<b>112</b>
<b>XXI Page break/no page break, depending on the specific line</b>	<b>113</b>
<b>XXII Parallel ledgroup</b>	<b>113</b>

<b>XXIII Compatibility with eledmac</b>	<b>117</b>
<b>XXIV The End</b>	<b>117</b>
<b>Appendix A Some things to do when changing version</b>	<b>118</b>
Appendix A.1 Migration to eledpar 1.4.3 . . . . .	118
Appendix A.2 Migration from eledpar to reledpar . . . . .	118
Appendix A.2.1 Deprecated options . . . . .	118
Appendix A.2.2 \renewcommandreplaced with command . . . . .	118
Appendix A.2.3 Commands the names of which have changed . . . . .	119
Appendix A.3 Migration to reledpar 2.2.0 . . . . .	119
Appendix A.4 Migration to reledpar 2.3.0 . . . . .	119
Appendix A.5 Migration to reledpar 2.4.0 . . . . .	119
Appendix A.6 Migration to reledpar 2.5.0 . . . . .	119
Appendix A.7 Migration to reledpar 2.6.0 . . . . .	119
Appendix A.8 Migration to reledpar 2.6.1 . . . . .	119
<b>References</b>	<b>120</b>
<b>Index</b>	<b>120</b>
<b>Change History</b>	<b>139</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 XXIV); 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>

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

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

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

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

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

`\beforecolumnseparator`  
`\aftercolumnseparator` By default, the spaces around column separator are the same as the space:

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

### 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 `sameparallelpagenumber` 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. 20.

### 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. 20.

`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. 43). 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.

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 Using line flag

`\Xlineflag` Use `\Xlineflag[⟨s⟩]` to add right line flag (6.3 p. 16) 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 TeX’s memory of previous texts and any associated notes, allowing longer apparent streams of parallel texts. The command should be applied to both left and right texts, and after making sure that all previous notes have been output. For example, along the lines of:

```
\begin{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.**

```

  \firstlinenum*
  \linenumincrement*
  \firstsublinenum*
\sublinenumincrement*
  \firstlinenumR
  \linenumincrementR
  \firstsublinenumR
\sublinenumincrementR
  \lineationR
  \lineation*
  \linenumberstyleR
\sublinenumberstyleR
  \linenumberstyle*
\sublinenumberstyle*
  \linenummarginR
  \linenummargin*
  \setRlineflag

```

The starred versions change both left and right numbering schemes.

The suffixed version change the right side, without regard to the position they are called.

`\lineationR` macro is the equivalent of reledmac `\lineation` macro for the right side.

`\lineation*` macro is the equivalent of reledmac `\lineation` macro for both sides.

`\linenumberstyleR` is the equivalent of reledmac `\linenumberstyle` for right text. `\sublinenumberstyleR` is the equivalent of reledmac `\sublinenumberstyle` right text. The starred version are for both side.

`\linenummarginR{<margin>}` sets the line margin for right side. `\linenummargin*{<margin>}` sets for both side. `<margin>` can be, as for reledmac’s `\linenummargin` 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 `\setRlineflag{flag}`. Use `\setRlineflag{}` to empty it.

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 `\linenumberLevenifblanktrue` to enable it on the left side, and `\linenumberRevenifblanktrue` to enable it on right side.

## 6.4 Chunks

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

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

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

`\autopar` The `\autopar` macro can be used, instead of manually inserting `\pstart... \pends`. 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`  
`\thestanzaR` 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
\end{Rightside}
\end{pages}
```

```

\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 `sameparallepage number` 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

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

`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  $\TeX$  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  $\LaTeX 2\epsilon$ . 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}[2016/06/16 v2.11.0 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.

```
\ifshiftedpstarts 12 \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.

```
\ifadvancedshiftedpstarts 18 \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.

```
\ifnomaxlines 25 \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   \nomaxlinestruetrue%
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}{\
43   continuousnumberingwithcolumnstrue}%
44 %

```

`\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 %
```

```
\led@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 %
```

```

ftside@PreviousNotPrinted82 \newcommand*\led@err@Leftside@PreviousNotPrinted}{%
htside@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}{%
ed@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 %

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

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

r@note@called@onrightside105 \newcommand\led@error@note@called@onrightside}[1]{%
r@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 %

```

```

\led@error@fail@patch@@mempnum12 \newcommand{\led@error@fail@patch@@mempnum}{%
113 \reledpar@error{Fail to patch \string\@mempnum\space command.}\@ehc}%
114 }%
115 %

```

```

\led@error@fail@patch@@outputpage16 \newcommand{\led@error@fail@patch@@outputpage}{%
117 \reledpar@error{Fail to patch \string\@outputpage\space command.}\@ehc}%
118 }%
119 %

```

```

\led@warn@ChangeSyncOption20 \newcommand*{\led@warn@ChangeSyncOption}[1]{%
121 \reledpar@warning{You have changed synchronization's options since last
run. We have not read line-list file #1. Please run LaTeX again.}%
122 }%
123 %

```

```

\led@warn@setting@in@rightside24 \newcommand{\led@warn@setting@in@rightside}[1]{%
125 \reledpar@warning{You use #1 inside rightside environment.\MessageBreak%
126 Such behavior is deprecated.\MessageBreak%
127 Use instead #1R or #1* in your preamble.}%
128 }
129 %

```

```

\led@error@missing@numbering30 \newcommand{\led@error@missing@numbering}[1]{%
131 \reledpar@error{Missing \string\...pstart\string\pend\space inside `#1`
environment}\@ehc}%
132 }%
133 %

```

## 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.

```

\newnamebox A set of macros for creating and using ‘named’ boxes; the macros are called after the
\setnamebox regular box macros, but including the string ‘name’.
\unhnamebox
\unvnamebox134 \providecommand*{\newnamebox}[1]{%
\namebox135 \expandafter\newbox\csname #1\endcsname}
136 \providecommand*{\setnamebox}[1]{%
137 \expandafter\setbox\csname #1\endcsname}
138 \providecommand*{\unhnamebox}[1]{%
139 \expandafter\unhbox\csname #1\endcsname}
140 \providecommand*{\unvnamebox}[1]{%

```

```

141 \expandafter\unvbox\csname #1\endcsname}
142 \providecommand*{\namebox}[1]{%
143     \csname #1\endcsname}
144
145 %

```

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

```

\usernamecount
146 \providecommand*{\newnamecount}[1]{%
147     \expandafter\newcount\csname #1\endcsname}
148 \providecommand*{\usernamecount}[1]{%
149     \csname #1\endcsname}
150
151 %

```

### 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>.nrR`. The `\extensionchars` applies to the right side files just as it does to the normal files.

```

152 \newcount\section@numR
153 \section@numR=\z@
154 %

```

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

```

155 \pst@rtedLfalse
156 \newif\ifpst@rtedR
157
158 %

```

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

```

159 \newcommand*{\beginnumberingR}{%
160     \ifnumberingR
161         \led@err@NumberingStarted
162         \endnumberingR
163     \fi
164     \global\l@dnumpstartsR \z@
165     \global\pst@rtedRfalse
166     \global\numberingRtrue
167     \global\advance\section@numR \@ne

```

```

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

```

`\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.

```

193 \def\endnumberingR{%
194 \ifnumberingR
195 \global\numberingRfalse
196 \normal@pars
197 \ifnum\l@dnumpstartsR=0%
198 \led@err@NumberingWithoutPstart%
199 \fi%
200 \ifl@dpairing
201 \global\pst@rtedRfalse
202 \else
203 \ifx\insertlines@listR\empty\else
204 \global\noteschanged@true
205 \fi
206 \ifx\line@listR\empty\else
207 \global\noteschanged@true

```

```

208     \fi
209     \fi
210     \ifnoteschanged@
211     \led@mess@NotesChanged
212     \fi
213 \else
214     \led@err@NumberingNotStarted
215 \fi
216 \endgroup
217 \if@noeled@sec\else%
218     \immediate\closeout\eled@sectioningR@out%
219 \fi%
220 }
221
222 %

```

`\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`

```

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

```

```

253 \let\c@subsubsection\c@subsubsectionR%
254 }%
255 %

```

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

```

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

```

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

```

275 \newcommand*\memorydumpL{%
276   \endnumbering
277   \numberingtrue
278   \global\pst@rtedLtrue
279   \global\advance\section@num \@ne
280   \led@mess@SectionContinued{\the\section@num}%
281   \line@list@stuff{\jobname.\extensionchars\the\section@num}%
282   \l@dend@stuff}
283
284 \newcommand*\memorydumpR{%
285   \endnumberingR
286   \numberingRtrue
287   \global\pst@rtedRtrue
288   \global\advance\section@numR \@ne
289   \led@mess@SectionContinued{\the\section@numR R}%
290   \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
291   \l@dend@stuff}
292
293 %

```

## 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`.

```

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

```

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

```

328 \newcommand{\set@continuousnumberingforR}{%
329   \ifcontinuousnumberingwithcolumns%
330     \ifnum\line@numR<\line@num%
331       \expandafter\setlinenum\expandafter{\the\line@num}%
332     \fi%
333     \ifnum\last@page@num>\last@page@numR%
334       \global\last@page@numR=\last@page@num%
335     \fi%
336   \fi%
337 }
338 %

```

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

```

339 \WithSuffix\newcommand\lineation*[1]{%
340   \lineation{#1}%
341   \lineationR{#1}%
342 }%
343 %

```

## IV.2 Setting line number margin

`\linenummargin` `\line@margin` 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`.

```

344 \newcount\line@marginR
345 %

```

By default put right text numbers at the right.

```

346 \line@marginR=\@ne
347
348 %

```

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

```

349 \newcommand{\linenummarginR}[1]{%
350   \l@dgetline@margin{#1}%
351   \ifnum\@l@dttempcntb>\m@ne%
352     \global\line@marginR=\@l@dttempcntb%
353   \fi%
354 }

```

```

355 \WithSuffix\newcommand\linenummargin*[1]{%
356   \l@getline@margin{#1}%
357   \ifnum\l@dttempcntb>\m@ne%
358     \global\line@marginR=\l@dttempcntb%
359     \global\line@margin=\l@dttempcntb%
360   \fi%
361 }
362 %

```

### IV.3 Setting lineation start and step

`\c@firstlinenumR` and `\c@linenumincrementR` 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.

```

363 \newcounter{firstlinenumR}
364   \setcounter{firstlinenumR}{5}
365 \newcounter{linenumincrementR}
366   \setcounter{linenumincrementR}{5}
367 %

```

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

```

368 \newcounter{firstsublinenumR}
369   \setcounter{firstsublinenumR}{5}
370 \newcounter{sublinenumincrementR}
371   \setcounter{sublinenumincrementR}{5}
372
373 %

```

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

```

\firstsublinenum
\sublinenumincrement
\firstlinenum*
\linenumincrement*
\firstsublinenum*
\sublinenumincrement*
374 \WithSuffix\newcommand\firstlinenum*[1]{%
375   \setcounter{firstlinenumR}{#1}%
376   \setcounter{firstlinenum}{#1}%
377 }
378 \WithSuffix\newcommand\linenumincrement*[1]{%
379   \setcounter{linenumincrementR}{#1}%
380   \setcounter{linenumincrement}{#1}%
381 }
382 \WithSuffix\newcommand\firstsublinenum*[1]{%
383   \setcounter{firstsublinenumR}{#1}%
384   \setcounter{firstsublinenum}{#1}%
385 }
386 \WithSuffix\newcommand\sublinenumincrement*[1]{%

```

```

387 \setcounter{sublinenumincrementR}{#1}%
388 \setcounter{sublinenumincrement}{#1}%
389 }
390 %

```

`\firstlinenumR` And the 'R' suffixed version.

```

\linenumincrementR
\firstsublinenumR
\sublinenumincrementR
391 \newcommand\firstlinenumR[1]{%
392 \setcounter{firstlinenumR}{#1}%
393 }
394 \newcommand\linenumincrementR[1]{%
395 \setcounter{linenumincrementR}{#1}%
396 }
397 \newcommand\firstsublinenumR[1]{%
398 \setcounter{subfirstlinenumR}{#1}%
399 }
400 \newcommand\sublinenumincrementR[1]{%
401 \setcounter{sublinenumincrementR}{#1}%
402 }
403 %

```

#### IV.4 Setting line flag

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

```

404 \newcommand{\setRlineflag}[1]{%
405 \gdef\Rlineflag{#1}%
406 }
407 \setRlineflag{R}
408 %

```

#### IV.5 Setting line number style

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

```

409 \newcommand*\linenumrepR[1]{\@arabic{#1}}
410 \newcommand*\sublinenumrepR[1]{\@arabic{#1}}
411
412 %

```

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

```

413 \newcommand*\linenumberstyleR[1]{%
414 \def\linenumrepR##1{\@nameuse{#1}##1}}
415 \newcommand*\sublinenumberstyleR[1]{%
416 \def\sublinenumrepR##1{\@nameuse{#1}##1}}
417 %

```

```

\linenumberstyle* And for both side.
\sublinenumberstyle*
418 \WithSuffix\newcommand\linenumberstyle*[1]{%
419   \linenumberstyle{#1}%
420   \linenumberstyleR{#1}%
421 }%
422
423 \WithSuffix\newcommand\sublinenumberstyle*[1]{%
424   \sublinenumberstyle{#1}%
425   \sublinenumberstyleR{#1}%
426 }%
427 %
428 %

```

## IV.6 Print marginal line number

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

```

429 \newif\iflinenumberLevenifblank
430 \newif\iflinenumberRevenifblank
431 %

```

`\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`.

```

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

```

## 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.

```
447 \newcount\line@numR
448 \newcount\subline@numR
449 \newcount\absline@numR
450
451 %
```

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

```
452 \list@create{\line@listR}
453 \list@create{\insertlines@listR}
454 \list@create{\actionlines@listR}
455 \list@create{\actions@listR}
456
457 %
```

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

```
458 \newcount\page@numR
459
460 %
```

#### IV.7.2 Specific to `reledpar`

`\linesinpar@listL`  
`\linesinpar@listR`  
`\maxlinesinpar@list` 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.

```
461 \list@create{\linesinpar@listL}
462 \list@create{\linesinpar@listR}
463 \list@create{\maxlinesinpar@list}
464
465 %
```

#### IV.8 Reading the line-list file

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

```
466 \newcommand{\list@clearing@regR}{%
467   \list@clear{\line@listR}%
468   \list@clear{\insertlines@listR}%
```

```

469 \list@clear{\actionlines@listR}%
470 \list@clear{\actions@listR}%
471 \list@clear{\linesinpar@listR}%
472 \list@clear{\linesonpage@listR}
473 }
474 %

```

`\@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.

```

475 \newcommand{\@par@sync@option}[1]{%
476 \IfStrEq{#1}{\@par@this@sync@option}%
477 {}%
478 {\ifledRcol%
479 \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
480 %
481 \else%
482 \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
483 %
484 \fi%
485 \endinput%
486 }%

```

`\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.

```

487 \newcommand{\@nl@regR}{%
488 \ifx\l@dchset@num\relax \else
489 \advance\absline@numR \@ne
490 \set@line@action
491 \let\l@dchset@num\relax
492 \advance\absline@numR \m@ne
493 \advance\line@numR \m@ne% % do we need this?
494 \fi
495 \advance\absline@numR \@ne

```

```

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

```

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

`\fix@page` `\fix@page`, defined by `reledmac`.

```

533 \newcount\last@page@numR
534 \last@page@numR=-10000
535
536 %

```

`\@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`.

```

537 \newcount\@lockR
538 \newcount\sub@lockR
539
540 \newcommand*{\do@lockonR}{%
541   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
542   \ifsublines@
543     \xright@appenditem{-1005}\to\actions@listR
544     \ifnum\sub@lockR=\z@
545       \sub@lockR \@ne
546     \else
547       \ifnum\sub@lockR=\thr@@
548         \sub@lockR \@ne
549       \fi
550     \fi
551   \else
552     \xright@appenditem{-1003}\to\actions@listR
553     \ifnum\@lockR=\z@
554       \@lockR \@ne
555     \else
556       \ifnum\@lockR=\thr@@
557         \@lockR \@ne
558       \fi
559     \fi
560   \fi}
561
562 %

```

- \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`.
- \do@lockoff** `\do@lockoffR`.
- \skip@lockoff** `\skip@lockoff`.

```

563
564
565 \newcommand{\do@lockoffR}{%
566   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
567   \ifsublines@
568     \xright@appenditem{-1006}\to\actions@listR
569     \ifnum\sub@lockR=\tw@
570       \sub@lockR \thr@@
571     \else
572       \sub@lockR \z@
573     \fi
574   \else
575     \xright@appenditem{-1004}\to\actions@listR
576     \ifnum\@lockR=\tw@
577       \@lockR \thr@@
578     \else
579       \@lockR \z@
580     \fi
581   \fi}
582
583
584 %

```

`\n@num`

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

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

```

585 \newcount\insert@countR
586 %

```

- #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.

```

587 \newcommand*{\@ref@regR}[2]{%
588   \global\advance\edtext@level by 1%
589   \global\insert@countR=#1\relax
590   \loop\ifnum\insert@countR>\z@
591     \xright@appenditem{\the\absline@numR}\to\insertlines@listR

```

```

592 \global\advance\insert@countR \m@ne
593 \repeat
594 %

```

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.

```

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

```

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

```

610 \xright@appenditem%
611 {\the\page@numR|\the\line@numR|}%
612 \ifsublines@ \the\subline@numR \else 0\fi|}%
613 \the\endpage@num|\the\endline@num|}%
614 \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@listR
615 %

```

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

```

616 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
@edtext@level\endcsname}%
617 %

```

Declare and init boolean for lemma in this level.

```

618 \providebool{lemmacommand@\the\@edtext@level}%
619 \boolfalse{lemmacommand@\the\@edtext@level}%
620 %

```

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

```

621 #2
622 % Now, we store the list of \protect\cs{@sw} of this current \protect\cs{
edtext} as an element of

```

```

623 % the global list of list of \protect\cs{sw} for a \protect\cs{edtext}
depth.
624 % \begin{macrocode}
625 \ifnum\@edtext@level>0%
626 \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtextR@the\@edtext@level\endcsname}}%
627 \ifcsundef{sw@list@edtextR@the\@edtext@level}{\
create@this@edtext@level}{}%
628 \letcs{\@tmp}{sw@list@edtextR@the\@edtext@level}%
629 \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}%
630 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
631 \global\cslet{sw@list@edtextR@the\@edtext@level}{\@tmp}%
632 \fi%
633 %

Decrease edtext level counter.

634 \global\advance\@edtext@level by -1%
635 }
636 %

```

`\@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`.

```

637 \renewcommand*{\@pend}[1]{%
638 \ifbypstart\global\line@num=0\fi%
639 \xright@appenditem{#1}\to\linesinpar@listL}
640 \renewcommand*{\@pendR}[1]{%
641 \ifbypstartR\global\line@numR=0\fi
642 \xright@appenditem{#1}\to\linesinpar@listR}
643 %
644 %

```

`\@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

```

645 \newcommand{\@pstart}[3]{%
646 \ifcsdef{minpage@pstart@#1}%
647 {\ifnumgreater{#2}{\csuse{minpage@pstart@#1}}%
648 {\csnumgdef{minpage@pstart@#1}{#2}}%
649 {}%
650 }%
651 {\csnumgdef{minpage@pstart@#1}{#2}}
652 \csnumgdef{afterlines@pstart@#1L}{#3}%
653 }%
654 %
655 \newcommand{\@pstartR}[3]{%
656 \numdef{\@tmp}{#2-1}%Because we have not to know in which page the pstart
starts, but in which pair of facing page

```

```

657 \ifcsdef{minpage@pstart@#1}%
658   {\ifnumgreater{\@tmp}{\csuse{minpage@pstart@#1}}%
659     {\csnumgdef{minpage@pstart@#1}{\@tmp}}%
660     }%
661   }%
662   {\csnumgdef{minpage@pstart@#1}{\@tmp}}
663   \csnumgdef{afterlines@pstart@#1R}{#3}%
664 }%
665 %

```

`\@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`.

```

666 \renewcommand*{\@lopL}[1]{%
667   \xright@appenditem{#1}\to\linesonpage@listL}
668 \renewcommand*{\@lopR}[1]{%
669   \xright@appenditem{#1}\to\linesonpage@listR}
670
671 %

```

## 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`.

```

672 \newwrite\linenum@outR
673 %

```

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

```

\first@linenum@out@Rtrue
\first@linenum@out@Rfalse
674 \newif\iffirst@linenum@out@R
675   \first@linenum@out@Rtrue
676 %

```

`\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.

```

677 \newcommand*{\line@list@stuffR}[1]{%
678   \read@linelist{#1}%
679   \iffirst@linenum@out@R
680     \immediate\closeout\linenum@outR
681     \global\first@linenum@out@Rfalse
682     \immediate\openout\linenum@outR=\l@auxdir#1%

```

```

683 \immediate\write\linenum@outR{\string\line@list@version{\
this@line@list@version}}}%
684 \ifl@dpaging%
685 \immediate\write\linenum@outR{\string\@par@sync@option{\
@par@this@sync@option}}}%
686 \fi%
687 \else
688 \if@minipage%
689 \leavevmode%
690 \fi%
691 \closeout\linenum@outR%
692 \openout\linenum@outR=\l@auxdir#1%
693 \fi}
694
695 %

```

**\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.

```

696 \newcommand*\new@lineL}{%
697 \write\linenum@out{\string\@nl[\the\c@page][\thepage]}}
698 %

```

**\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.

```

699 \newcommand*\new@lineR}{%
700 \write\linenum@outR{\string\@nl[\the\c@page][\thepage]}}
701 %

```

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

**\startsub** **\endsub** `\startsub` and `\endsub` turn sub-lineation on and off, by writing appropriate instructions to the line-list file. There are both defined in `reledmac`.

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

**\setline** You can use `\setline{<num>}` in running text (i.e., within `\pstart . . . \pend`) to set the current visible line-number to a specified positive value. It is defined in `reledmac`.

**\setlinenum** You can use `\setlinenum{<num>}` before a `\pstart` to set the visible line-number to a specified positive value. It writes a `\l@d@set` command to the line-list file. It is defined in `reledmac`.

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

`\skipnumbering`

## 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.

`\critext`  
`\edtext`  
`\set@line`

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

### 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.

```
\newseries@par702 \newcommand{\newseries@par}[1]{%
703 %
```

#### 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 temporary the footnote not printed. We check the `nofamiliar` and `nocritical` `reledmac` options.

```
704 \unless\ifnofamiliar@%
705 \csgdef{onlysideX@#1}{}%
706 \newnamebox{footins#1@kept}%
707 \fi%
708 \unless\ifnocritical@%
709 \csgdef{Xonlyside@#1}{}%
710 \newnamebox{#1footins@kept}%
711 \fi%
712 %
```

### V.2 Tools specific to familiar footnotes

```
713 \unless\ifnofamiliar@%
714 %
```

### V.2.1 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  $\LaTeX$  reads first all the notes on one side, then all the notes on the other side. Then, however,  $\LaTeX$  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  $\LaTeX$  counter `footnoteX`. This the only one manipulated by user, and the only one finally printed.
2. A  $\TeX$  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  $\LaTeX$  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.

```
715 \newcounter{footnote#1@typeset}%
716 %
```

### V.2.2 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.

```
717 \expandafter\list@create\csname footnote#1@mk\endcsname%
718 %
```

Then, declare the `\footnoteXnomk` command.

```
719 \expandafter\newcommand\csname footnote#1nomk\endcsname[1]{%
720 %
```

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

```
721 \toggletrue{nomk@}%
722 \csuse{footnote#1}{##1}%
723 \togglefalse{nomk@}%
724 %
```

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`.

```

725     \letcs{\@tmp}{footnote#1@mk}%
726     \numdef\@tmpa{\csuse{c@footnote#1}}%
727     \global\xright@appenditem{\@tmpa}\to\@tmp%
728     \global\cslet{footnote#1@mk}{\@tmp}%
729   }%
730 %

```

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

```

731     \expandafter\newcommand\csname footnote#1mk\endcsname{%
732 %

```

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

```

733     \letcs{\@tmp}{footnote#1@mk}%
734     \gl@p\@tmp\to\@tmpa%
735     \global\cslet{footnote#1@mk}{\@tmp}%
736 %

```

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

```

737     \letcs{\old@footnote}{c@footnote#1}%
738     \setcounter{footnote#1}{\@tmpa}%
739 %

```

Define the footnote mark and print it

```

740     \protected@csxdef{\thefnmark#1}{\csuse{thefootnote#1}}%
741     \csuse{\@footnotemark#1}%
742 %

```

Restore previous footnote counter and finally add space.

```

743     \setcounter{footnote#1}{\old@footnote}%
744     \xspace%
745   }%
746 %

```

End of tools specific to familiar notes.

```

747   \fi
748 %

```

End of `\newseries@par`.

```

749 }%
750 %

```

### V.2.3 Get correct footnote number

`\get@familiarfootnote@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.

```

751 \newcommand{\save@familiarfootnote@number}{%
752   \unless\ifnofamiliar@%
753     \def\do##1{\csxdef{saved@footnote##1}{\the\csname c@footnote##1\
endcsname}}%
754     \dolistloop{\@series}%
755     \fi%
756   \xdef\saved@footnote{\the\c@footnote}%
757 }
758 \newcommand{\get@familiarfootnote@number}{%
759   \unless\ifnofamiliar@%
760     \def\do##1{\setcounter{footnote##1@typeset}{\csuse{saved@footnote##1}}}
%
761     \dolistloop{\@series}%
762     \fi%
763     \setcounter{footnote@typeset}{\saved@footnote}%
764 }
765 %

```

### V.3 Create hooks

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

```

766 \unless\ifnocritical@%
767   \newhookcommand@series{Xonlyside}%
768   \fi%
769 \unless\ifnofamiliar@%
770   \newhookcommand@series{onlysideX}%
771   \fi
772
773
774 %

```

### V.4 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`.

```

775 \newcommand{\init@series@par}{%
776   \def\do##1{\newseries@par{##1}}%
777   \dolistloop{\@series}%
778 }%
779 \init@series@par%
780 %

```

## V.5 Tools specific to L<sup>A</sup>T<sub>E</sub>X's classical footnotes

As users can use classical footnotes of L<sup>A</sup>T<sub>E</sub>X (`\footnote`) in parallel texts, we must integrate the same tools to get correct number as for `reledmac`' footnotes (V.2.1 p. 46).

```
\footnote@reading81 \newcount\footnote@reading%
\footnote@typeset82 \newcounter{footnote@typeset}%
783 %
```

## 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@pc84 \list@create{\list@pstartL@pc}%
\list@pstartR@pc85 \list@create{\list@pstartR@pc}%
786 %
```

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@pc87 \def\dump@pstartL@pc{%
\dump@pstartR@pc88 \xright@appenditem{\the\c@pstartL}\to\list@pstartL@pc%
789 \global\cslet{numberpstart@L\the\l@dumpstartsL}{\ifnumberpstart}%
790 }%
791
792 \def\dump@pstartR@pc{%
793 \xright@appenditem{\the\c@pstartR}\to\list@pstartR@pc%
794 \global\cslet{numberpstart@R\the\l@dumpstartsR}{\ifnumberpstart}%
795 }%
796
797 %
```

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

```
\restore@pstartR@pc
798 \def\restore@pstartL@pc{%
799 \ifx\list@pstartL@pc\empty\else%
800 \gl@p\list@pstartL@pc\to\@temp%
801 \global\c@pstartL=\@temp%
802 \fi%
```

```

803 }%
804 \def\restore@pstartR@pc{%
805   \ifx\list@pstartR@pc\empty\else%
806     \gl@p\list@pstartR@pc\to\@temp%
807     \global\c@pstartR=\@temp%
808   \fi%
809 }%
810 %

```

## 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.

```

811 \newenvironment{pairs}{%
812   \l@dpairingtrue
813   \l@dpagingfalse
814   \initnumbering@quote
815   \save@familiarfootnote@number%
816   \if@ledgroup%
817     \get@familiarfootnote@number%
818   \fi%
819   \save@section@number%
820   \at@begin@pairs%
821 }{%
822   \l@dpairingfalse
823 }
824
825 %

```

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

```

826 \newcommand{\AtBeginPairs}[1]{\xdef\at@begin@pairs{#1}}%
827 \def\at@begin@pairs{}%
828
829 %

```

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.

```

830 \newenvironment{pages}{%
831   \l@dpairingtrue
832   \l@dpagingtrue
833   \initnumbering@quote

```

```

834 \save@familiarfootnote@number%
835 \if@ledgroup%
836   \get@familiarfootnote@number%
837 \fi%
838 \save@section@number%
839 \setlength{\Lcolwidth}{\textwidth}%
840 \setlength{\Rcolwidth}{\textwidth}%
841 }{%
842   \l@dpairingfalse
843   \l@dpagingfalse
844 }
845
846 %

```

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

```

847 \newif\ifinstanzaL
848 \newif\ifinstanzaR
849 %

```

**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.

```

850 \newenvironment{Leftside}{%
851   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname\else%
852     \led@err@Leftside@PreviousNotPrinted%
853 \fi%
854 \ledRcolfalse
855 \setcounter{pstartL}{1}
856 \let\pstart\pstartL
857 \let\thepstart\thepstartL
858 \let\pend\pendL
859 \let\memorydump\memorydumpL
860 \Leftsidehook
861 \let\old@startstanza\@startstanza
862 \def\@startstanza[##1]{\global\instanzaLtrue\old@startstanza[##1]}
863 }{
864   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname%
865     \led@error@missing@numbering{Leftside}%
866 \fi%
867   \Leftsidehookend}
868 %

```

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

`\Leftsidehookend`

`\Rightsidehook`

`\Rightsidehookend`

```

869 \newcommand*\Leftsidehook{}
870 \newcommand*\Leftsidehookend{}
871 \newcommand*\Rightsidehook{}
872 \newcommand*\Rightsidehookend{}
873
874 %

```

**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.

```

875 \newenvironment{Rightside}{%
876   \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else%
877   \led@err@Rightside@PreviousNotPrinted%
878   \fi%
879   \ledRcoltrue
880   \let\beginnumbering\beginnumberingR
881   \let\endnumbering\endnumberingR
882   \let\pausenumbering\pausenumberingR
883   \let\resumenumbering\resumenumberingR
884   \let\memorydump\memorydumpR
885   \let\thepstart\thepstartR
886   \let\pstart\pstartR
887   \let\pend\pendR
888   \let\ledpb\ledpbR
889   \let\lednopb\lednopbR
890   \let\lineation\lineationR
891   \Rightsidehook
892   \let\old@startstanza\@startstanza
893   \def\@startstanza[##1]{\global\instanzaRtrue\old@startstanza[##1]}
894 }{%
895   \ledRcolfalse
896   \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
897   \led@error@missing@numbering{Rightside}%
898   \fi%
899   \Rightsidehookend
900 }
901
902 %

```

## 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.

```
903 \newcount\num@linesR
904 \newbox\one@lineR
905 \newcount\par@lineR
906 %
```

`\pstartL` `\pstart` starts the paragraph by clearing the `\inserts@list` list and other relevant variables, and then arranges for the subsequent text to go into the appropriate box. `\pstartR` 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.

```
907
908 \newcounter{pstartL}
909 \renewcommand{\thepstartL}{\bfseries\@arabic\c@pstartL}. }
910 \newcounter{pstartR}
911 \renewcommand{\thepstartR}{\bfseries\@arabic\c@pstartR}. }
912
913 \newcommandx*{\pstartL}[1][1]{%
914   \if@nobreak%
915     \let\@oldnobreak\@nobreaktrue%
916   \else%
917     \let\@oldnobreak\@nobreakfalse%
918   \fi%
919   \@nobreaktrue%
920   \ifluatex%
921     \xdef\l@luatextextdir@L{\the\textdir}%
922     \xdef\l@luatexpardir@L{\the\pardir}%
923     \xdef\l@luatexbodydir@L{\the\bodydir}%
924   \fi%
925   \ifnumbering \else%
926     \led@err@PstartNotNumbered%
927     \beginnumbering%
928   \fi%
929   \ifnumberedpar@%
930     \led@err@PstartInPstart%
```

```

931 \pend%
932 \fi%
933 %

```

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

```

934 \ifpstartL\else%
935 \list@clear{\inserts@list}%
936 \global\let\next@insert=\empty%
937 \global\pstartLtrue%
938 \fi%
939 \begingroup\everypar{}%
940 %

```

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.

```

941 \global\advance\l@dnumpstartsL \one%
942 \ifnum\l@dnumpstartsL>\l@dc@maxchunks%
943 \led@err@TooManyPstarts%
944 \global\l@dnumpstartsL=\l@dc@maxchunks%
945 \fi%
946 \global\setnamebox{l@dLcolrawbox\the\l@dnumpstartsL}=\vbox\bgroup%
947 %

```

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.

```

948 \l@dzeropenalties%
949 \ifautopar\else%
950 \ifnumberpstart%
951 \ifsidepstartnum%
952 \else%
953 \thepstartL%
954 \fi%
955 \fi%
956 \fi%
957 \hsize=Lcolwidth%
958 \numberedpar@true%
959 \iflabelpstart\protected@edef\@currentlabel%
960 {\p@pstartL\thepstartL}\fi%
961 %

```

Dump the optional arguments

```

962 \ifstrempy{#1}%
963 {\csgdef{before@pstartL@the\l@dnumpstartsL}{\at@every@pstart}}%
964 {\csgdef{before@pstartL@the\l@dnumpstartsL}{\noindent#1}}%
965 \at@every@pstart@call%
966 %

```

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

```

967 \ignorespaces%
968 %

```

```

969 }
970 %

```

The same for right side.

```

971 \newcommand*{\pstartR}[1][1]{%
972   \if@nbreak%
973     \let\@oldnbreak\@nbreaktrue%
974   \else%
975     \let\@oldnbreak\@nbreakfalse%
976   \fi%
977   \@nbreaktrue%
978   \ifluatex%
979     \xdef\l@luatextextdir@R{\the\textdir}%
980     \xdef\l@luatexpardir@R{\the\pardir}%
981     \xdef\l@luatexbodydir@R{\the\bodydir}%
982   \fi%
983   \ifnumberingR \else%
984     \led@err@PstartNotNumbered%
985     \beginnumberingR%
986   \fi%
987   \ifnumberedpar@%
988     \led@err@PstartInPstart%
989     \pendR%
990   \fi%
991   \ifpstart@rtedR\else%
992     \list@clear{\inserts@listR}%
993     \global\let\next@insertR=\empty%
994     \global\pstart@rtedRtrue%
995   \fi%
996   \begingroup\everypar{}%
997   \global\advance\l@dnumpstartsR \@ne%
998   \ifnum\l@dnumpstartsR>\l@dc@maxchunks%
999     \led@err@TooManyPstarts%
1000   \global\l@dnumpstartsR=\l@dc@maxchunks%
1001   \fi%
1002   \global\setnamebox{l@dRcolrawbox\the\l@dnumpstartsR}=\vbox\bgroup%
1003     \l@dzeropenalties%
1004     \ifautopar\else%
1005       \ifnumberpstart%
1006         \ifsidepstartnum\else%
1007           \thepstartR%
1008         \fi%
1009       \fi%
1010     \fi%
1011   \hsize=\Rcolwidth%
1012   \numberedpar@true%
1013   \iflabelpstart\protected@edef\@currentlabel%

```

```

1014     {\p@pstartR\thepstartR}\fi%
1015     \ifstrempy{#1}%
1016     {\csgdef{before@pstartR@the\l@dnumpstartsR}{\at@every@pstart}}%
1017     {\csgdef{before@pstartR@the\l@dnumpstartsR}{\noindent#1}}%
1018     \at@every@pstart@call%
1019     \ignorespaces%
1020   }
1021 %

```

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

```

1022 \newcommand*{\pendL}[1][1]{%
1023   \ifnumbering \else%
1024     \led@err@PendNotNumbered%
1025   \fi%
1026   \ifnumberedpar@ \else%
1027     \led@err@PendNoPstart%
1028   \fi%
1029 %

```

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.

```

1030   \endgraf\global\num@lines=\prevgraf\egroup%
1031   \global\par@line=0%
1032 %

```

End the group that was begun in the \pstart.

```

1033   \endgroup%
1034   \ignorespaces%
1035   \@oldnobreak%
1036   \dump@pstartL@pc%
1037   \ifnumberpstart%
1038     \addtocounter{pstartL}{1}%
1039   \fi
1040   \parledgroup@beforenotes@save{L}%
1041 %

```

Dump content of the optional argument.

```

1042   \ifstrempy{#1}%
1043     {\csgdef{after@pendL@the\l@dnumpstartsL}{\at@every@pend}}%
1044     {\csgdef{after@pendL@the\l@dnumpstartsL}{\noindent#1}}%
1045   }
1046 %

```

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

```

1047 \newcommand*{\pendR}[1][1]{%
1048   \ifnumberingR \else%

```

```

1049 \led@err@PendNotNumbered%
1050 \fi%
1051 \ifnumberedpar@ \else%
1052 \led@err@PendNoPstart%
1053 \fi%
1054 \endgraf\global\num@linesR=\prevgraf\egroup%
1055 \global\par@lineR=0%
1056 \endgroup%
1057 \ignorespaces%
1058 \@oldnobreak%
1059 \dump@pstartR@pc%
1060 \ifnumberpstart%
1061 \addtocounter{pstartR}{1}%
1062 \fi%
1063 \parledgroup@beforenotes@save{R}%
1064 \ifstreempty{#1}%
1065 {\csgdef{after@pendR@the\l@dnumpstartsR}{\at@every@pend}}%
1066 {\csgdef{after@pendR@the\l@dnumpstartsR}{\noindent#1}}%
1067 }
1068
1069 %

```

**\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.

```

1070 \newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
1071 \gdef\at@every@pstart@call{}%
1072 %

```

**\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**

```

1073 \newif\ifprint@last@after@pendL%
1074 \newif\ifprint@last@after@pendR%
1075 %

```

## 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**

```

1076 \newbox\l@dleftbox
1077 \newbox\l@drightbox
1078
1079 %

```

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

```

\countRline
1080 \newcount\countLline
1081   \countLline \z@
1082 \newcount\countRline
1083   \countRline \z@
1084
1085 %

```

`\@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`

```

1086 \newcount\@donereallinesL
1087 \newcount\@donetotallinesL
1088 \newcount\@donereallinesR
1089 \newcount\@donetotallinesR
1090
1091 %

```

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

```

1092 \newcommand*\do@lineL{%
1093   \letcs{\ifnumberpstart}{numberpstart@L\the\l@dpscl}%
1094   \advance\countLline \@ne%
1095   \ifvbox\namebox{1@dLcolrawbox\the\l@dpscl}%
1096     {\vbadness=10000%
1097      \splittopskip=\z@%
1098      \do@lineLhook%
1099      \l@emptyd@ta%
1100      \global\setbox\one@line=\vsplit\namebox{1@dLcolrawbox\the\l@dpscl}%
1101        to\baselineskip}%
1102   \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
\parledgroup@notes@startL}{}%
1103   \unvbox\one@line \global\setbox\one@line=\lastbox%
1104   \@writepageofparL%
1105   \getline@numL%
1106   \ifnum\@lock>\@ne%
1107     \inserthangingsymboltrue%
1108   \else%
1109     \inserthangingsymbolfalse%
1110   \fi%
1111   \setbox\l@dleftbox%
1112   \hb@xt@ \lcolwidth{%
1113     \ifl@dhidenumber%
1114       \global\l@dhidenumberfalse%
1115       \f@x@l@cks%
1116     \else%
1117       \affixline@num%

```

```

1118 \fi%
1119 \xifinlist{\the\l@dpscl}{\eled@sections@}%
1120 {%
1121   \if@firstlineofpage%
1122     \set@Xtxtbeforenotes%
1123     \global\@firstlineofpagefalse%
1124   \fi%
1125   \insert@msdata%
1126   \add@inserts%
1127   \affixside@note%
1128 }%
1129 {\print@lineL}%
1130 }%
1131 \add@penaltiesL%
1132 \global\advance\@donereallinesL\@ne%
1133 \global\advance\@donetotallinesL\@ne%
1134 \else%
1135   \iflinenumberLevenifblank
1136     \new@lineL%
1137     \l@emptyd@ta%
1138     \getline@numL%
1139     \affixline@num%
1140     \setbox\l@dleftbox \hb@xt@ \Lcolwidth{%
1141       \l@dld@ta%
1142       \hspace*{\Lcolwidth}%
1143       \ledrlfill\l@drd@ta%
1144     }%
1145   \else%
1146     \setbox\l@dleftbox \hb@xt@ \Lcolwidth{\hspace*{\Lcolwidth}}%
1147   \fi%
1148   \global\advance\@donetotallinesL\@ne%
1149 \fi%
1150 }%
1151
1152
1153 %

```

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

```

1154 \def\print@lineL{%
1155   \affixpstart@numL%
1156   \l@dld@ta%
1157   \if@firstlineofpage%
1158     \set@Xtxtbeforenotes%
1159     \global\@firstlineofpagefalse%
1160   \fi%
1161   \insert@msdata%
1162   \add@inserts\affixside@note%
1163   \l@dlsn@te%

```

```

1164 \hb@xt@ \Lcolwidth{\ledllfill\hb@xt@ \wd\one@line{%
1165 \do@insidelineLhook%
1166 \ifluatex%
1167 \textdir\l@luatextextdir@L%
1168 \fi%
1169 \new@lineL%
1170 \inserthangingsymbolL%
1171 \l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta%
1172 \l@drsn@te}}%
1173
1174 %

```

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

```

1175 \def\print@eledsectionL{%
1176 \addtocounter{pstartL}{-1}%
1177 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{%
1178 \ifdefstring{\@eledsectmark}{L}{\ledsectnomark}
1179 \numdef{\temp@}{\l@dpscL-1}%
1180 \xifinlist{\temp@}{\eled@sections@}{\@nobreaktrue}{\@nobreakfalse}%
1181 \@eled@sectioningtrue%
1182 \bgroup%
1183 \ifluatex%
1184 \textdir\l@luatextextdir@L%
1185 \pardir\l@luatexpardir@L%
1186 \bodydir\l@luatexbodydir@L%
1187 \ifdefstring{\l@luatextextdir@L}{TRT}{\@RTLtrue}{%
1188 \fi%
1189 \csuse{eled@sectioning@the\l@dpscL}%
1190 \egroup%
1191 \@eled@sectioningfalse%
1192 \global\csundef{eled@sectioning@the\l@dpscL}%
1193 \if@RTL%
1194 \hspace{-3\paperwidth}%
1195 {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1196 \else%
1197 \hspace{3\paperwidth}%
1198 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1199 \fi%
1200 \vskip\eledsection@correcting@skip%
1201 }
1202
1203 %

```

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

```

\doinssidelineLhook \newcommand*{\dolineLhook}[1]{\gdef\dolineLhook{#1}}%
\doinssidelineRhook \newcommand*{\dolineRhook}[1]{\gdef\dolineRhook{#1}}%
\doinssidelineLhook \newcommand*{\doinssidelineLhook}[1]{\gdef\doinssidelineLhook{#1}}%

```

```

1207 \newcommand*\doinsidelineRhook}[1]{\gdef\do@insidelineRhook{#1}}%
1208
1209 %

```

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

```

\do@lineRhook
\do@insidelineLhook \newcommand*\do@lineLhook{}
1210 \newcommand*\do@lineRhook{}
1211 \do@insidelineRhook \newcommand*\do@insidelineLhook{}
1212 \newcommand*\do@insidelineRhook{}
1213
1214
1215 %

```

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

```

1216 \newcommand*\do@lineR}{%
1217 \let\linenumrepL\linenumrep%
1218 \let\sublinenumrepL\sublinenumrep%
1219 \let\linenumrepR\linenumrepR%
1220 \let\sublinenumrepR\sublinenumrepR%
1221 \letcs{ifnumberpstart}{numberpstart@R\the\l@dpscr}%
1222 \ledRcol@true%
1223 \advance\countRline \@ne%
1224 \ifvbox\namebox{l@dRcolrawbox\the\l@dpscr}%
1225   {\vbadness=10000%
1226    \splittopskip=\z@%
1227    \do@lineRhook%
1228    \l@demtyd@ta%
1229    \global\setbox\one@lineR=\vsplit\namebox{l@dRcolrawbox\the\l@dpscr}%
1230     to\baselineskip}%
1231   \IfStrEq{splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startR}{}%
1232   \unvbox\one@lineR \global\setbox\one@lineR=\lastbox%
1233   \@writepageofparR%
1234   \getline@numR%
1235   \ifnum\@lockR>\@ne%
1236     \inserthangingsymbolRtrue%
1237   \else%
1238     \inserthangingsymbolRfalse%
1239   \fi%
1240   \setbox\l@drightbox%
1241   \hb@xt@ \Rcolwidth{%
1242     \ifl@dhiddenumber%
1243       \global\l@dhiddenumberfalse%
1244       \f@x@l@cksR%
1245     \else%
1246       \affixline@numR%
1247     \fi%
1248     \xifinlist{\the\l@dpscr}{\eled@sectionsR@@}%

```

```

1249 {%
1250   \if@firstlineofpageR%
1251     \set@Xtxtbeforenotes%
1252     \global\@firstlineofpageRfalse%
1253   \fi%
1254   \insert@msdata%
1255   \add@insertsR%
1256   \affixside@noteR%
1257 }%
1258 {\print@lineR}%
1259 }%
1260 \add@penaltiesR%
1261 \global\advance\@donereallinesR\@ne%
1262 \global\advance\@donetotallinesR\@ne%
1263 \else%
1264   \iflinenumberRevenifblank%
1265     \new@lineR
1266     \l@emptyd@ta%
1267     \getline@numR%
1268     \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1269       \affixline@numR%
1270       \l@dld@ta%
1271       \hspace*\Rcolwidth}%
1272     \ledrlfill\l@drd@ta%
1273   }%
1274   \else%
1275     \setbox\l@drightbox \hb@xt@ \Rcolwidth{\hspace*\Rcolwidth}}%
1276   \fi%
1277   \global\advance\@donetotallinesR\@ne%
1278   \fi%
1279   \ledRcol@false%
1280   \let\linenumrep\linenumrepL%
1281   \let\sublinenumrep\sublinenumrepL%
1282 }
1283
1284
1285 %

```

`\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.

```

1286 \newcommand*\getline@numR{%
1287   \global\advance\absline@numR \@ne
1288   \do@actionsR
1289   \do@ballastR
1290   \ifledgroupnotesR\else

```

```

1291     \ifnumberline
1292     \ifsublines@
1293         \ifnum\sub@lockR<\tw@
1294             \global\advance\subline@numR \@ne
1295         \fi
1296     \else
1297         \ifnum\@lockR<\tw@
1298             \global\advance\line@numR \@ne
1299             \global\subline@numR \z@
1300         \fi
1301     \fi
1302 \fi
1303 }
1304 }
1305 \newcommand*{\getline@numL}{%
1306     \global\advance\absline@num \@ne
1307     \do@actions
1308     \do@ballast
1309     \ifledgroupnotesL@else
1310         \ifnumberline
1311             \ifsublines@
1312                 \ifnum\sub@lock<\tw@
1313                     \global\advance\subline@num \@ne
1314                 \fi
1315             \else
1316                 \ifnum\@lock<\tw@
1317                     \global\advance\line@num \@ne
1318                     \global\subline@num \z@
1319                 \fi
1320             \fi
1321         \fi
1322     \fi
1323 }
1324
1325
1326 %

```

`\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.

```

1327 \newcommand*{\do@ballastR}{\global\ballast@count=\z@
1328     \begingroup
1329     \advance\absline@numR \@ne
1330     \ifnum\next@actionlineR=\absline@numR
1331         \ifnum\next@actionR>-1001
1332             \global\advance\ballast@count by -\c@ballast
1333         \fi
1334     \fi
1335     \endgroup}
1336 %

```

`\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.

`\do@actions@fixedcodeR`

`\do@actions@nextR`

```

1337
1338 \newif\ifl@dskipversenumberR
1339 \newcommand*{\do@actions@fixedcodeR}{%
1340   \ifcase\@l@dttempcnta%
1341   \or%           % 1001 = starting sublineation
1342     \global\sublines@true
1343   \or%           % 1002 = ending sublineation
1344     \global\sublines@false
1345   \or%           % 1003 = starting locking number
1346     \global\@lockR=\@ne
1347   \or%           % 1004 = ending locking number
1348     \ifnum\@lockR=\tw@
1349       \global\@lockR=\thr@@
1350     \else
1351       \global\@lockR=\z@
1352     \fi
1353   \or%           % 1005 = starting locking subnumber
1354     \global\sub@lockR=\@ne
1355   \or%           % 1006 = ending locking subnumber
1356     \ifnum\sub@lockR=\tw@
1357       \global\sub@lockR=\thr@@
1358     \else
1359       \global\sub@lockR=\z@
1360     \fi
1361   \or%           % 1007 = skipping numbering
1362     \l@dskipnumbertrue
1363   \or%           % 1008 = skipping numbering in stanza
1364     \l@dskipversenumberRtrue%
1365   \or%           % 1009 = hiding number
1366     \l@dhiddenumbertrue%
1367   \or%           % 1010 = inserting msdata
1368     \add@msdata%
1369   \else%
1370     \led@warn@BadAction
1371   \fi%
1372 }
1373
1374
1375 \newcommand*{\do@actionsR}{%
1376   \global\let\do@actions@nextR=\relax
1377   \@l@dttempcntb=\absline@numR
1378   \ifnum\@l@dttempcntb<\next@actionlineR\else
1379     \ifnum\next@actionR>-1001\relax
1380       \@firstlineofpageRtrue%
1381       \global\page@numR=\next@actionR

```

```

1382     \ifbypage@R
1383         \global\line@numR \z@ \global\subline@numR \z@
1384     \fi
1385     \add@msdata@firstlineofpage%
1386 \else
1387     \ifnum\next@actionR<-4999\relax % 9/05 added relax here
1388         \@l@tempcnta=-\next@actionR
1389         \advance\@l@tempcnta by -5001\relax
1390         \ifsublines@
1391             \global\subline@numR=\@l@tempcnta
1392         \else
1393             \global\line@numR=\@l@tempcnta
1394         \fi
1395     \else
1396         \@l@tempcnta=-\next@actionR
1397         \advance\@l@tempcnta by -1000\relax
1398         \do@actions@fixedcodeR
1399     \fi
1400 \fi
1401 \ifx\actionlines@listR\empty
1402     \gdef\next@actionlineR{1000000}%
1403 \else
1404     \gl@p\actionlines@listR\to\next@actionlineR
1405     \gl@p\actions@listR\to\next@actionR
1406     \global\let\do@actions@nextR=\do@actionsR
1407 \fi
1408 \fi
1409 \do@actions@nextR}
1410
1411 %

```

#### 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@cksR
\affixline@numR
1412
1413 \newcommand*{\l@dcalcnm}[3]{%
1414     \ifnum #1 > #2\relax
1415         \@l@tempcnta = #1\relax
1416         \advance\@l@tempcnta by -#2\relax
1417         \divide\@l@tempcnta by #3\relax
1418         \multiply\@l@tempcnta by #3\relax
1419         \advance\@l@tempcnta by #2\relax
1420     \else
1421         \@l@tempcnta=#2\relax
1422     \fi}
1423
1424 \newcommand*{\ch@cksub@l@ckR}{%
1425     \ifcase\sub@lockR

```

```

1426 \or
1427 \ifnum\sublock@disp=\@ne
1428 \l@dttempcntb \z@ \l@dttempcnta \@ne
1429 \fi
1430 \or
1431 \ifnum\sublock@disp=\tw@
1432 \else
1433 \l@dttempcntb \z@ \l@dttempcnta \@ne
1434 \fi
1435 \or
1436 \ifnum\sublock@disp=\z@
1437 \l@dttempcntb \z@ \l@dttempcnta \@ne
1438 \fi
1439 \fi}
1440
1441 \newcommand*\ch@ck@l@ckR}{%
1442 \ifcase\@lockR
1443 \or
1444 \ifnum\lock@disp=\@ne
1445 \l@dttempcntb \z@ \l@dttempcnta \@ne
1446 \fi
1447 \or
1448 \ifnum\lock@disp=\tw@
1449 \else
1450 \l@dttempcntb \z@ \l@dttempcnta \@ne
1451 \fi
1452 \or
1453 \ifnum\lock@disp=\z@
1454 \l@dttempcntb \z@ \l@dttempcnta \@ne
1455 \fi
1456 \fi}
1457
1458 \newcommand*\f@x@l@cksR}{%
1459 \ifcase\@lockR
1460 \or
1461 \global\@lockR \tw@
1462 \or \or
1463 \global\@lockR \z@
1464 \fi
1465 \ifcase\sub@lockR
1466 \or
1467 \global\sub@lockR \tw@
1468 \or \or
1469 \global\sub@lockR \z@
1470 \fi}
1471
1472
1473 \newcommand*\affixline@numR}{%
1474 \ifledgroupnotesR@else\ifnumberline
1475 \ifl@dskipnumber

```

```

1476 \global\l@dskipnumberfalse
1477 \else
1478 \ifsublines@
1479 \l@l@tempcntb=\subline@numR
1480 \l@dcalcnnum{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR
}%
1481 \ch@cksub@lockR
1482 \else
1483 \l@l@tempcntb=\line@numR
1484 \ifx\linenumberlist\empty
1485 \l@dcalcnnum{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%
1486 \else
1487 \l@l@tempcnta=\line@numR
1488 \edef\rem@inder{\linenumberlist,\number\line@numR,}%
1489 \edef\sc@n@list{\def\noexpand\sc@n@list
1490 ###1,\number\l@l@tempcnta,###2|\def\noexpand\rem@inder{###2}}%
1491 \sc@n@list\expandafter\sc@n@list\rem@inder|
1492 \ifx\rem@inder\empty\advance\l@l@tempcnta\@ne\fi
1493 \fi
1494 \ch@ck@l@ckR
1495 \fi
1496 \ifnum\l@l@tempcnta=\l@l@tempcntb
1497 \ifl@dskipversenumberR\else
1498 \iftwocolumn
1499 \if@firstcolumn
1500 \gdef\l@dld@ta{\llap{\leftlinenumR}}%
1501 \else
1502 \gdef\l@drd@ta{\rlap{\rightlinenumR}}%
1503 \fi
1504 \else
1505 \l@l@tempcntb=\line@marginR
1506 \ifnum\l@l@tempcntb>\@ne
1507 \advance\l@l@tempcntb by\page@numR
1508 \fi
1509 \ifodd\l@l@tempcntb
1510 \gdef\l@drd@ta{\rlap{\rightlinenumR}}%
1511 \else
1512 \gdef\l@dld@ta{\llap{\leftlinenumR}}%
1513 \fi
1514 \fi
1515 \fi
1516 \fi
1517 \f@x@l@cksR
1518 \fi
1519 \fi
1520 \fi}
1521 %

```

### 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@numL1522
\affixpstart@numR1523 \newcommand*{\affixpstart@numL}{%
  \leftpstartnum1524 \ifsidepstartnum
\rightpstartnum1525 \if@twocolumn
  \if@firstcolumn
  \leftpstartnumL1526
  \rightpstartnumL1527
  \ifpstartnumL1528
  \ifpstartnumR1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
\newcommand*{\affixpstart@numR}{%
  \ifsidepstartnum
  \if@twocolumn
  \if@firstcolumn
  \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
  \else
  \gdef\l@drd@ta{\rlap{\rightpstartnumR}}}%
  \fi
  \else
  \l@dttempcntb=\line@margin
  \ifnum\l@dttempcntb>\@ne
  \advance\l@dttempcntb \page@num
  \fi
  \ifodd\l@dttempcntb
  \gdef\l@drd@ta{\rlap{\rightpstartnumL}}}%
  \else
  \gdef\l@dld@ta{\llap{\leftpstartnumL}}}%
  \fi
  \fi
}
\newcommand*{\affixpstart@numR}{%
  \ifsidepstartnum
  \if@twocolumn
  \if@firstcolumn
  \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
  \else
  \gdef\l@drd@ta{\rlap{\rightpstartnumR}}}%
  \fi
  \else
  \l@dttempcntb=\line@marginR
  \ifnum\l@dttempcntb>\@ne
  \advance\l@dttempcntb \page@numR
  \fi
  \ifodd\l@dttempcntb
  \gdef\l@drd@ta{\rlap{\rightpstartnumR}}}%
  \else
  \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
  \fi
}

```

```

1562   \fi
1563 \fi
1564 }
1565
1566 \newcommand*\leftpstartnumL){
1567 \ifpstartnum
1568 \thepstartL
1569 \kern\linenumsep\global\pstartnumfalse\fi
1570 }
1571 \newcommand*\rightpstartnumL){
1572 \ifpstartnum\kern\linenumsep
1573 \thepstartL
1574 \global\pstartnumfalse\fi
1575 }
1576 \newif\ifpstartnumR
1577 \pstartnumRtrue
1578 \newcommand*\leftpstartnumR){
1579 \ifpstartnumR
1580 \thepstartR
1581 \kern\linenumsep\global\pstartnumRfalse\fi
1582 }
1583 \newcommand*\rightpstartnumR){
1584 \ifpstartnumR\kern\linenumsep
1585 \thepstartR
1586 \global\pstartnumRfalse\fi
1587 }
1588 %

```

## 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.

```

1589 \list@create{\inserts@listR}
1590 %

```

`\add@insertsR` The right text version.

```

\add@inserts@nextR
1591 \newcommand*\add@insertsR){%
1592 \global\let\add@inserts@nextR=\relax
1593 \ifx\inserts@listR\empty \else
1594 \ifx\next@insertR\empty
1595 \ifx\insertlines@listR\empty
1596 \global\noteschanged@true
1597 \gdef\next@insertR{100000}%
1598 \else
1599 \gl@p\insertlines@listR\to\next@insertR
1600 \fi
1601 \fi

```

```

1602 \ifnum\next@insertR=\absline@numR
1603 \gl@p\inserts@listR\to\@insertR
1604 \@insertR
1605 \global\let\@insertR=\undefined
1606 \global\let\next@insertR=\empty
1607 \global\let\add@inserts@nextR=\add@insertsR
1608 \fi
1609 \fi
1610 \add@inserts@nextR}
1611
1612 %

```

### 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@tempcnta` is used to calculate and accumulate the penalty; it is initially set to the value of `\ballast@count`, which has been worked out in `\do@ballast`. Finally, the penalty is checked to see that it 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.

```
1613 \newcommand*\add@penaltiesL}{ }
1614 \newcommand*\add@penaltiesR}{ }
1615
1616 %
```

## 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.

```
1617 \newcommand*\flush@notesR}{%
1618   \@xloop
1619   \ifx\inserts@listR\empty \else
1620     \gl@p\inserts@listR\to\@insertR
1621     \@insertR
1622     \global\let\@insertR=\undefined
1623   \repeat}
1624
1625 %
```

## IX Footnotes

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

`\print@Xnotes@forpages` `\correct@Xfootins@box` `\print@notesX@forpages` `\correct@footinsX@box` 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 `\print@Xnotes@forpages`, which replaces `\print@Xnotes` inside `\Pages`. Here is how we proceed<sup>6</sup>:

- 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.

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

- 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  $\TeX$  to a temporary box.
- After going to the next page, before typesetting anything 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.

```
1626 \newcommand\print@Xnotes@forpages[1]{%
1627 %
```

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

```
1628   \ifcseempty{Xonlyside@#1}{%
1629     \csuse{#1footstart}{#1}%
1630     \csuse{#1footgroup}{#1}%
1631   }%
1632 %
```

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

```
1633   {%
1634     \ifboolexpr{%
1635       ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page
1636       }})%
1637       or%
1638       (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}})%
1639     }%
1640 %
```

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

```
1640   {%
1641     \csuse{#1footstart}{#1}%
1642     \csuse{#1footgroup}{#1}%
1643 %
```

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`.

```
1644   \global\count\csuse{#1footins}=\z@%
1645   \global\skip\csuse{#1footins}=\z@%
1646   \global\dimen\csuse{#1footins}=\maxdimen%
1647 }%
1648 %
```

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.

```

1649     {%
1650     \global\count\csuse{#1footins}=\csuse{default@#1footins}%
1651     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
1652     \bgroup%
1653     \csuse{Xnotefontsize@#1}%
1654     \global\dimen\csuse{#1footins}=\csuse{Xmaxhnotes@#1}%
1655     \egroup%
1656 %

```

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

```

1657     \global\setnamebox{#1footins@kept}=\box\namebox{#1footins}%
1658     }%
1659 %

```

End of \print@Xnotes@forpages.

```

1660     }%
1661 }%
1662 %

```

And now, the same for familiar footnotes.

```

1663 \newcommand\print@notesX@forpages[1]{%
1664   \ifcsemtyp{onlysideX@#1}{%
1665     \csuse{footstart#1}{#1}%
1666     \csuse{footgroup#1}{#1}%
1667   }%
1668   {%
1669     \ifboolexpr{%
1670       ((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page
1671       }%
1672       (test {\ifcsstring{onlysideX@#1}{R}} and test{\ifnumodd{\c@page}}))%
1673     }%
1674     {%
1675       \csuse{footstart#1}{#1}%
1676       \csuse{footgroup#1}{#1}%
1677       \global\count\csuse{footins#1}=\z@%
1678       \global\skip\csuse{footins#1}=\z@%
1679       \global\dimen\csuse{footins#1}=\maxdimen%
1680     }%
1681   }%
1682   {%
1683     \global\count\csuse{footins#1}=\csuse{default@footins#1}%
1684     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
1685     \bgroup%
1686     \csuse{Xnotefontsize@#1}%
1687     \global\dimen\csuse{footins#1}=\csuse{maxhnotesX@#1}%
1688     \egroup%

```

```

1689 \global\setnamebox{footins#1@kept}=\box\namebox{footins#1}%
1690 }%
1691 }%
1692 }%
1693 %

```

`\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.

```

1694 \newcommand{\insert@notes@for@onlyside}{%
1695 \def\do##1{%
1696 \unless\ifnocritical@%
1697 \ifvoid\csuse{##1footins@kept}\else%
1698 \expandafter\insert\csname ##1footins\endcsname%
1699 \bgroup%
1700 \unvnamebox{##1footins@kept}%
1701 \egroup%
1702 \fi%
1703 \fi%
1704 \unless\ifnofamiliar@%
1705 \ifvoid\csuse{footins##1@kept}\else%
1706 \expandafter\insert\csname footins##1\endcsname%
1707 \bgroup%
1708 \unvnamebox{footins##1@kept}%
1709 \egroup%
1710 \fi%
1711 \fi%
1712 }%
1713 \dolistloop{\@series}%
1714 }%
1715 %

```

## 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.

```

1716 \list@create{\labelref@listR}
1717
1718 %

```

`\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`.

```

1719 \def\l@dmake@labelsR#1|#2|#3|#4|#5{%

```

```

1720 \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
else
1721 \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
1722 \fi
1723 \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
{#1|#2|#3|#4|\@Rlineflag}%
1724 \ignorespaces}
1725 \AtBeginDocument{%
1726 \def\l@dmake@labelsR#1|#2|#3|#4|#5{%
1727 }
1728
1729 %

```

`\@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*`

```

1730 \WithSuffix\newcommand\sidenotemargin*[1]{%
1731 \l@dgetsidenote@margin{#1}
1732 \global\sidenote@marginR=\@l@tempcntb
1733 \global\sidenote@margin=\@l@tempcntb
1734 }
1735 \newcount\sidenote@marginR
1736 \global\sidenote@margin=\@ne
1737
1738 %

```

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

```

1739 \newcommand*{\affixside@noteR}{%
1740 \def\sidenotecontent@{%
1741 \numgdef{\itemcount@}{0}%
1742 \def\do##1{%
1743 \ifnumequal{\itemcount@}{0}%
1744 {%
1745 \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
1746 {\appto\sidenotecontent@{\sidenotesep ##1}%
1747 }%

```

```

1748     \numgdef{\itemcount@}{\itemcount@+1}%
1749   }%
1750   \dolistloop{\l@dcsnotetext}%
1751   \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
1752   \gdef\@templ@d{%
1753     \gdef\@templ@n{\l@dcsnotetext\l@dcsnotetext@1\l@dcsnotetext@r}%
1754     \ifx\@templ@d\@templ@n \else%
1755       \if@twocolumn%
1756         \if@firstcolumn%
1757           \setl@dlp@rbox{##1}{\sidenotecontent@}%
1758         \else%
1759           \setl@drp@rbox{\sidenotecontent@}%
1760         \fi%
1761       \else%
1762         \@l@dttempcntb=\sidenote@marginR%
1763         \ifnum\@l@dttempcntb>\@ne%
1764           \advance\@l@dttempcntb by\page@numR%
1765         \fi%
1766         \ifodd\@l@dttempcntb%
1767           \setl@drp@rbox{\sidenotecontent@}%
1768           \gdef\sidenotecontent@{%
1769             \numdef{\itemcount@}{0}%
1770             \dolistloop{\l@dcsnotetext@1}%
1771             \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
1772             \setl@dlp@rbox{\sidenotecontent@}%
1773           \else%
1774             \setl@dlp@rbox{\sidenotecontent@}%
1775             \gdef\sidenotecontent@{%
1776               \numdef{\itemcount@}{0}%
1777               \dolistloop{\l@dcsnotetext@r}%
1778               \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
1779               \setl@drp@rbox{\sidenotecontent@}%
1780             \fi%
1781           \fi%
1782         \fi%
1783     }
1784
1785 %

```

## XII Familiar footnotes

`\l@dbfnote` `\l@dbfnote` adds the footnote to the insert list, and `\vl@dbfnote` calls the original `\@footnotetext`. There are both defined in `reledmac`.

`\normalbfnoteX`

### XIII 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.

```

\inserthangingsymbolL86 \newif\ifinserthangingsymbolR
\inserthangingsymbolR87 \newcommand{\inserthangingsymbolL}{%
1788   \ifinserthangingsymbol%
1789     \ifinstanzaL%
1790       \hskip \ifundefined{sza@00}{0}{\expandafter%
1791         \noexpand\cname sza@00\endcsname}\stanzaindentbase%
1792       \@hangingsymbol%
1793     \fi%
1794   \fi%
1795 }%
1796 \newcommand{\inserthangingsymbolR}{%
1797   \ifinserthangingsymbolR%
1798     \ifinstanzaR%
1799       \hskip \ifundefined{sza@00}{0}{\expandafter%
1800         \noexpand\cname sza@00\endcsname}\stanzaindentbase%
1801       \@hangingsymbol%
1802     \fi%
1803   \fi%
1804 }%
1805 %

```

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.

```

1806 \chardef\next=\catcode`\&
1807 \catcode`\&=\active
1808
1809 %

```

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

```

1810 \newenvironment{astanza}[1] [] {%
1811   \catcode`\&\active
1812   \global\stanza@count\@ne\stanza@modulo\@ne
1813   \ifnum\usernamecount{sza@00}=\z@
1814     \let\stanza@hang\relax
1815     \let\endlock\relax
1816   \else
1817     \rightskip\z@ plus 1fil\relax
1818   \fi
1819   \ifnum\usernamecount{szp@00}=\z@
1820     \let\sza@penalty\relax

```

```

1821 \fi
1822 \def&{%
1823   \endlock\mbox{}%
1824   \sza@penalty
1825   \global\advance\stanza@count\@ne
1826   \@astanza@line}%
1827 \def\&{\@stopastanza}%
1828 \ifboolexpr{not test{\ifdefvoid{\@at@every@stanza}} and test{\ifstrempy
1829   {\pstart[\@at@every@stanza]}}%
1830   {\pstart[#1]}}%
1831   \@astanza@line
1832   \let\par\relax\ignorespaces%No paragraph in verses
1833 }{}
1834
1835 %

```

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

```

1836 \newcommandx{\@stopastanza}[1][1,usedefault]{%
1837   \endlock\mbox{}%
1838   \ifboolexpr{not test{\ifdefvoid{\@at@every@stop@stanza}} and test{\
1839     ifstrempy{#1}}}%
1840     {\pend[\@at@every@stop@stanza]}}%
1841     {\pend[#1]}}%
1842 }%

```

`\@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.

```

1843 \newcommand*{\@astanza@line}{%
1844   \ifnum\value{stanzaindentrepetition}=0
1845     \parindent=\csname sza@\number\stanza@count
1846       @\endcsname\stanzaindentbase
1847   \else
1848     \parindent=\csname sza@\number\stanza@modulo
1849       @\endcsname\stanzaindentbase
1850     \managestanza@modulo
1851   \fi
1852   \endgraf
1853   \stanza@hang%
1854   \ignorespaces}
1855
1856 %

```

Lastly reset the modified category codes.

```

1857 \catcode`\&=\next
1858
1859 %

```

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

```

\thestanzaR
1860 \newcounter{stanzaL}
1861 \newcounter{stanzaR}
1862 \renewcommand{\thestanzaL}{%
1863   \textbf{\arabic{stanzaL}}%
1864 }
1865 \renewcommand{\thestanzaR}{%
1866   \textbf{\arabic{stanzaR}}%
1867 }
1868 %
1869 %

```

## XIV 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
1870 \newif\ifl@dusedbabel
1871 %

```

`\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 `\`.

```

1872 \patchcmd{\bbl@set@language}%
1873   {\select@language{\languagename}}%
1874   {\edef\languagename{#1}\select@language{\languagename}}%
1875   {}%
1876   {}%
1877
1878 %

```

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 \newcommand*{\l@duselanguage}[1]{%
1879 \gdef\theledlanguageL{%
1880 \gdef\theledlanguageR{%
1881
1882 %
1883 %

```

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

```

1884 \AtBeginDocument{%
1885   \ifundefined{xpg@main@language}{%
1886     \ifundefined{bbl@main@language}{%
1887 %

```

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

```

1888   \l@dusedbabelfalse
1889   }{%
1890 %

```

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.

```

1891   \l@dusedbabeltrue
1892   \let\l@doldselectlanguage\selectlanguage
1893   \let\l@doldbbl@set@language\bbl@set@language
1894   \renewcommand{\selectlanguage}[1]{%
1895     \l@doldselectlanguage{#1}%
1896     \ifledRcol \gdef\theledlanguageR{#1}%
1897     \else      \gdef\theledlanguageL{#1}%
1898     \fi}
1899 %

```

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

```

1900   \renewcommand*{\l@duselanguage}[1]{%
1901     \expandafter\l@doldselectlanguage\expandafter{#1}}
1902 %

```

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

```

1903   \gdef\theledlanguageL{\bbl@main@language}%
1904   \gdef\theledlanguageR{\bbl@main@language}%
1905   }%
1906   }
1907 %

```

If use polyglossia

```

1908 { \let\old@otherlanguage\otherlanguage%
1909 \renewcommand{\otherlanguage}[2] [] {%
1910 \selectlanguage[#1]{#2}%
1911 \ifledRcol \gdef\theledlanguageR{#2}%
1912 \else \gdef\theledlanguageL{#2}%
1913 \fi}%
1914 \renewcommand{\l@duselanguage}[1] {%
1915 \csuse{no\language@numbers}\select@language{#1}%
1916 }%
1917 \gdef\theledlanguageL{\xpg@main@language}%
1918 \gdef\theledlanguageR{\xpg@main@language}%
1919 %

```

That is it.

```

1920 }}
1921 %

```

## XV 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.

```

1922 \newcount\l@dc@maxchunks
1923 \newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}
1924 \maxchunks{5120}
1925
1926 %

```

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

```

1927 \newcount\l@dnumstartsR
1928
1929 %

```

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

```

\l@pscl
\l@pscl
1930 \newcount\l@dpscl
1931 \newcount\l@dpscl
1932
1933 %

```

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

```

1934 \newcommand*\l@dsetuprawboxes}{%
1935   \@l@tempcntb=\l@dc@maxchunks
1936   \loop\ifnum\@l@tempcntb>\z@
1937     \newnamebox{\l@dLcolrawbox\the\@l@tempcntb}
1938     \newnamebox{\l@dRcolrawbox\the\@l@tempcntb}
1939     \advance\@l@tempcntb \m@ne
1940   \repeat}
1941
1942 %

```

`\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.

```

1943 \newcommand*\l@dsetupmaxlinecounts}{%
1944   \@l@tempcntb=\l@dc@maxchunks
1945   \loop\ifnum\@l@tempcntb>\z@
1946     \newnamecount{\l@dmaxlinesinpar\the\@l@tempcntb}
1947     \advance\@l@tempcntb \m@ne
1948   \repeat}
1949 \newcommand*\l@dzeromaxlinecounts}{%
1950   \begingroup
1951   \@l@tempcntb=\l@dc@maxchunks
1952   \loop\ifnum\@l@tempcntb>\z@
1953     \global\usenamecount{\l@dmaxlinesinpar\the\@l@tempcntb}=\z@
1954     \advance\@l@tempcntb \m@ne
1955   \repeat
1956   \endgroup}
1957
1958 %

```

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

```

1959 \AtBeginDocument{%
1960   \l@dsetuprawboxes
1961   \l@dsetupmaxlinecounts
1962   \l@dzeromaxlinecounts
1963   \l@dnumpstartsL=\z@
1964   \l@dnumpstartsR=\z@
1965   \l@dpscL=\z@
1966   \l@dpscR=\z@}
1967
1968 %

```

## XVI Checking text to be processed

```

\if@pstarts \check@pstarts returns \@pstartstrue if there are any unprocessed chunks.
\@pstartstrue
\@pstartsfalse
\check@pstarts
1969 \newif\if@pstarts
1970 \newcommand*\check@pstarts}{%
1971   \@pstartsfalse
1972   \ifnum\l@dnumpstartsL>\l@dpscL
1973     \@pstartstrue
1974   \else
1975     \ifnum\l@dnumpstartsR>\l@dpscR
1976       \@pstartstrue
1977     \fi
1978   \fi
1979 }
1980
1981 %

```

```

\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
1982 \newif\ifaraw@text
1983 \newcommand*\checkraw@text}{%
1984   \araw@textfalse
1985   \ifvbox\namebox{1@dLcolrawbox\the\l@dpscL}
1986     \araw@texttrue
1987   \else
1988     \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}
1989       \araw@texttrue
1990     \fi
1991   \fi
1992 }
1993
1994 %

```

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

```

1995 \newcommand*\@writelinesinparL}{%
1996   \edef\next{%
1997     \write\linenum@out{\string\@pend[\the\@donereallinesL]}}%
1998   \next
1999   \global\@donereallinesL \z@}
2000 \newcommand*\@writelinesinparR}{%
2001   \edef\next{%
2002     \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}%
2003   \next
2004   \global\@donereallinesR \z@}
2005
2006 %

```

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

`\@writepageofparR`

```

2007 \newcommand*\@writepageofparL}[0]{%
2008   \ifnum\@donereallinesL=\z@%
2009     \edef\next{%
2010       \write\linenum@out{\string\@pstart{\the\l@dpscl}{\the\c@page}{\the\
numpagelinesL}}%
2011     }%
2012     \next%
2013   \fi%
2014 }%
2015 \newcommand*\@writepageofparR}[0]{%
2016   \ifnum\@donereallinesR=\z@%
2017     \edef\next{%
2018       \write\linenum@outR{\string\@pstartR{\the\l@dpscl}{\the\c@page}{\the\
numpagelinesR}}%
2019     }%
2020     \next%
2021   \fi%
2022 }%
2023 %

```

## XVII Parallel columns

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

`\@eledsectionR`

```

2024 \newsavebox{\@eledsectionL}%
2025 \newsavebox{\@eledsectionR}%
2026 %

```

`\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.

```

2027 \newcommand*\Columns}{%
2028   \ifl@dpairing%
2029     \led@err@Columns@InsideEnv%
2030   \fi%
2031   \expandafter\ifvoid\c@name l@dRcolrawbox1\endcsname%
2032     \led@err@Columns@WithoutEnv%
2033   \else%
2034     \global\l@dprintingcolumnstrue%
2035     \eledsection@correcting@skip=-\baselineskip% Correction for sections'
titles
2036     \ifnum\l@dnumstartsL=\l@dnumstartsR\else
2037       \led@err@BadLeftRightPstarts{\the\l@dnumstartsL}{\the\l@dnumstartsR}%
2038     \fi
2039 %

```

Start a group and zero counters, etc.

```

2040 \begingroup
2041 \l@dzeropenalties
2042 \endgraf\global\num@lines=\prevgraf
2043 \global\num@linesR=\prevgraf
2044 \global\par@line=\z@
2045 \global\par@lineR=\z@
2046 \global\l@dpscL=\z@
2047 \global\l@dpscR=\z@
2048 \get@familiarfootnote@number%
2049 %

```

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

```

2050 \check@pstarts
2051 \loop\if@pstarts
2052 \global\pstartnumtrue
2053 \global\pstartnumRtrue
2054 %

```

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.

```

2055 \global\advance\l@dpscL \@ne
2056 \global\advance\l@dpscR \@ne
2057 \restore@pstartL@pc%
2058 \restore@pstartR@pc%
2059 %

```

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

```

2060 \Columns@print@before@pstart%
2061 %

```

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

```

2062 \checkraw@text
2063 { \loop\ifaraw@text
2064 %

```

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

```

2065 \l@duselanguage{\theledlanguageL}%
2066 \do@lineL
2067 \xifinlist{\the\l@dpscL}{\eled@sections@@}
2068 {%
2069 \ifdefstring{\@eledsectmark}{L}%
2070 {\csuse{eled@sectmark@\the\l@dpscL}%
2071 }}%
2072 \global\csundef{eled@sectmark@\the\l@dpscL}%
2073 \savebox{\@eledsectionL}{\parbox[t][t]{\Lcolwidth}{\vbox
{\print@eledsectionL}}\vbox{}-> prevent alignment troubles with RTL
language

```

```

2074         }%
2075         {}%
2076         \l@duselanguage{\theledlanguageR}%
2077         \do@lineR
2078         \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}
2079         {%
2080         \ifdefstring{\@eledsectmark}{R}%
2081         {\csuse{eled@sectmark@the\l@dpscR R}%
2082         }}%
2083         \global\csundef{eled@sectmark@the\l@dpscR R}%
2084         \savebox{\@eledsectionR}{\parbox[t][t]{\Rcolwidth}{\vbox
\print@eledsectionR}}}%\vbox{}-> prevent alignment troubles with RTL
language
2085         {}%
2086         \hb@xt@ \hsize{%
2087         \ifdefstring{\columns@position}{L}{\hfill }%
2088         \unhbox\l@dleftbox%
2089         \ifhbox\@eledsectionL%
2090         \usebox{\@eledsectionL}%
2091         \fi%
2092         \print@columnseparator%
2093         \unhbox\l@drightbox%
2094         \ifhbox\@eledsectionR%
2095         \usebox{\@eledsectionR}%
2096         \fi%
2097         \ifdefstring{\columns@position}{R}{\hfill}%
2098         }%
2099         \checkraw@text
2100         \checkverseL
2101         \checkverseR
2102         \checkpb@columns
2103         \repeat}
2104 %

```

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`.

```

2105     \@writelinesinparL
2106     \@writelinesinparR
2107     \check@pstarts
2108     \ifbypstart@%
2109     \write\linenum@out{\string\@set[1]}
2110     \resetprevline@
2111     \fi
2112     \ifbypstart@R
2113     \write\linenum@outR{\string\@set[1]}
2114     \resetprevline@
2115     \fi
2116     \Columns@print@after@pend%
2117     \repeat

```

```

2118 %
    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.
2119     \flush@notes
2120     \flush@notesR
2121 \endgroup
2122 %
2123 \global\l@dpscL=\z@
2124 \global\l@dpscR=\z@
2125 \global\l@dnumstartsL=\z@
2126 \global\l@dnumstartsR=\z@
2127 \global\l@dprintingcolumnsfalse%
2128 \ignorespaces
2129     \global\instanzaLfalse%
2130     \global\instanzaRfalse%
2131 \fi}
2132
2133 %

```

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

```

2134 \def\print@columnseparator{%
2135   \ifdim\beforecolumnseparator<0pt%
2136     \hfill%
2137   \else%
2138     \hspace{\beforecolumnseparator}%
2139   \fi%
2140   \columnseparator%
2141   \ifdim\aftercolumnseparator<0pt%
2142     \hfill%
2143   \else%
2144     \hspace{\beforecolumnseparator}%
2145   \fi%
2146 }%
2147 %

```

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

```

2148
2149 \newcommand{\checkpb@columns}{%
2150   \newif\if@pb
2151   \newif\if@nopb
2152   \IfStrEq{\led@pb@setting}{before}{
2153     \numdef{\next@absline}{\the\absline@num+1}%

```

```

2154 \numdef{\next@abslineR}{\the\absline@numR+1}%
2155 \xifinlistcs{\next@absline}{l@prev@pb}{\@pbtrue}{}%
2156 \xifinlistcs{\next@abslineR}{l@prev@pbR}{\@pbtrue}{%
2157 \xifinlistcs{\next@absline}{l@prev@nopb}{\@nopbtrue}{}%
2158 \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\@nopbtrue}{%
2159 }{}
2160 \IfStrEq{\led@pb@setting}{after}{
2161 \xifinlistcs{\the\absline@num}{l@prev@pb}{\@pbtrue}{}%
2162 \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\@pbtrue}{%
2163 \xifinlistcs{\the\absline@num}{l@prev@nopb}{\@nopbtrue}{}%
2164 \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\@nopbtrue}{%
2165 }{}
2166 \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
2167 \if@pb\pagebreak[4]\fi
2168 }
2169 %

```

**\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).

```

2170 \newcommand*{\columnseparator}{%
2171 \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}}
2172 \newdimen\columnrulewidth
2173 \columnrulewidth=\z@
2174
2175 %

```

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

**\columns@position**

```

2176 \newcommand*{\columnspan}[1]{%
2177 \xdef\columns@position{#1}%
2178 }%
2179 \xdef\columns@position{R}%
2180 %

```

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

**\aftercolumnseparator**

```

2181 \newlength{\beforecolumnseparator}%
2182 \setlength{\beforecolumnseparator}{-2pt}%
2183
2184 \newlength{\aftercolumnseparator}%
2185 \setlength{\aftercolumnseparator}{-2pt}%
2186
2187 %

```

`\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 `\setpositionliketwocolumns@L` macros are called in `\beginnumbering` in a **non-parallel** typesetting context to fix the position of the lines. The `\setnoteposition...` macros are called in `\xxxfootstart` in a **non-parallel** typesetting context to fix the position of notes block.

```

2188 \newcommand{\setwidthliketwocolumns@L}{%
2189 % Temporary dimension, initially equal to the standard hsize, i.e. text
2190 width
2191 % \begin{macrocode}
2192 \newdimen\temp%
2193 \temp=\hsize%
2194 %

```

Hsize : Left + Right width

```

2194 \hsize=\Lcolwidth%
2195 \advance\hsize\Rcolwidth%
2196 %

```

Now, calculating the remaining space

```

2197 \advance\temp-\hsize%
2198 %

```

And multiply the hsize by 2/3 of this space

```

2199 \multiply\temp by 2%
2200 \divide\temp by 3%
2201 \advance\hsize\temp%
2202 }%
2203
2204 \newcommand{\setpositionliketwocolumns@L}{%
2205 \renewcommand{\ledrfill}{\hfill}%
2206 }%
2207
2208 \newcommand{\setnotespositionliketwocolumns@L}{%
2209 }%
2210
2211
2212 %

```

```

2213 \newcommand{\setwidthliketwocolumns@C}{%
2214 % Temporary dimension, initially equal to the standard hsize, i.e. text
2215 width
2216 %

```

```

2216 \newdimen\temp%
2217 \temp=\hsize%
2218 % Hsize : Left + Right width
2219 %

```

```

2220 \hsize=\Lcolwidth%
2221 \advance\hsize\Rcolwidth%
2222 % Now, calculating the remaining space
2223 %

```

```

2224 \advance\temp-\hsize%
2225 %

```

And multiply the hsize by 1/2 of this space

```

2226 \divide\temp by 2%
2227 \advance\hsize\temp%
2228 }%
2229
2230 \newcommand{\setpositionliketwocolumns@C}{%
2231 \doinsidelinehook{\hfill}%
2232 \renewcommand{\ledrlfill}{\hfill}%
2233 }%
2234
2235 \newcommand{\setnotespositionliketwocolumns@C}{%
2236 \newdimen\temp%
2237 \newdimen\tempa%
2238 \temp=\hsize%
2239 \tempa=\Lcolwidth%
2240 \advance\tempa\Rcolwidth%
2241 \advance\temp-\tempa%
2242 \divide\temp by 2%
2243 \leftskip=\temp%
2244 \rightskip=-\temp%
2245 }%
2246
2247 \newcommand{\setwidthliketwocolumns@R}{%
2248 %

```

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

```

2249 \newdimen\temp%
2250 \temp=\hsize%
2251 %

```

Hsize : Left + Right width

```

2252 \hsize=\Lcolwidth%
2253 \advance\hsize\Rcolwidth%
2254 %

```

Now, calculating the remaining space

```

2255 \advance\temp-\hsize%
2256 %

```

And multiply the hsize by 2/3 of this space

```

2257 \multiply\temp by 2%
2258 \divide\temp by 3%
2259 \advance\hsize\temp%
2260 }%
2261
2262 \newcommand{\setpositionliketwocolumns@R}{%
2263 \doinsidelinehook{\hfill}%
2264 }%
2265
2266 \newcommand{\setnotespositionliketwocolumns@R}{%
2267 \newdimen\temp%
2268 \newdimen\tempa%
2269 \temp=\hsize%
2270 \tempa=Lcolwidth%
2271 \advance\tempa\Rcolwidth%
2272 \advance\temp-\tempa%
2273 \divide\temp by 2%
2274 \leftskip=\temp%
2275 \rightskip=-\temp%
2276 }%
2277
2278 %

```

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

```

2279 \newcommand{\Columns@print@before@pstart}{%
2280 \ifboolexpr{%
2281 test{\ifcsstring{before@pstartL@the\l@dpscl}{\at@every@pstart}}%
2282 and test {\ifcsstring{before@pstartR@the\l@dpscr}{\at@every@pstart}}%
2283 and test {\ifdefempty{\at@every@pstart}}}%
2284 {}%
2285 {%
2286 \hb@xt@ \hsize{%
2287 \ifdefstring{\columns@position}{L}{\hfill }%
2288 \par\parbox[t] [] [t]{\Lcolwidth}{%
2289 \cuse{before@pstartL@the\l@dpscl}%
2290 }%
2291 \print@columnseparator%
2292 \parbox[t] [] [t]{\Rcolwidth}{%
2293 \set@sectcountR%
2294 \cuse{before@pstartR@the\l@dpscr}%
2295 }%
2296 \ifdefstring{\columns@position}{R}{\hfill}%
2297 }%
2298 }%
2299 \global\csundef{before@pstartL@the\l@dpscl}%
2300 \global\csundef{before@pstartR@the\l@dpscr}%
2301 }%

```

```

2302 \newcommand{\Columns@print@after@pend}{%
2303   \ifboolexpr{%
2304     test{\ifcsstring{after@pendL@the\l@dpscl}{\at@every@pend}}%
2305     and test {\ifcsstring{after@pendR@the\l@dpscR}{\at@every@pend}}%
2306     and test {\ifdefempty{\at@every@pend}}}%
2307     {%
2308     {%
2309       \hb@xt@ \hsize{%
2310         \ifdefstring{\columns@position}{L}{\hfill }%
2311         \parbox[t] [] [t]{\Lcolwidth}{%
2312           \csuse{after@pendL@the\l@dpscl}%
2313         }%
2314         \print@columnseparator%
2315         \parbox[t] [] [t]{\Rcolwidth}{%
2316           \set@sectcountR%
2317           \csuse{after@pendR@the\l@dpscR}%
2318         }%
2319         \ifdefstring{\columns@position}{R}{\hfill}%
2320       }%
2321     }%
2322     \global\csundef{after@pendL@the\l@dpscl}%
2323     \global\csundef{after@pendR@the\l@dpscR}%
2324   }%
2325   %

```

## XVIII Parallel pages

This is considerably more complicated than parallel columns.

### XVIII.1 Specific counters

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

```

2326 \newcount\numpagelinesL
2327 \newcount\numpagelinesR
2328 \newcount\l@dminpagelines
2329
2330 %

```

### XVIII.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.

```

2331 \newcommandx*{\Pages}[1][1,usedefault]{%
2332   \ifl@dpairing%

```

```

2333 \led@err@Pages@InsideEnv%
2334 \fi%
2335 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
2336 \led@err@Pages@WithoutEnv%
2337 \else%
2338 \ifstrequal{#1}{mainmatter}{\Pages@mainmattertrue}{\Pages@mainmatterfalse
}%
2339 \eledsection@correcting@skip=-2\baselineskip% line correcting for section
titles.
2340 \parledgroup@notespacing@set@correction%
2341 \typeout{}%
2342 \typeout{***** PAGES *****}%
2343 \ifnum\l@dnumpststartsL=\l@dnumpststartsR\else%
2344 \led@err@BadLeftRightPstarts{\the\l@dnumpststartsL}{\the\l@dnumpststartsR}%
2345 \fi%
2346 %

```

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

```

2347 \cleartol@devenpage%
2348 \global\l@dprintingpagetrue%
2349 \begingroup%
2350 %

```

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

```

2351 \setlength{\Lcolwidth}{\textwidth}%
2352 \setlength{\Rcolwidth}{\textwidth}%
2353 %

```

```

2354 \l@dzeropenalties%
2355 \endgraf\global\num@lines=\prevgraf%
2356 \global\num@linesR=\prevgraf%
2357 \global\par@line=\z@%
2358 \global\par@lineR=\z@%
2359 \global\l@dpscL=\z@%
2360 \global\l@dpscR=\z@%
2361 \writtenlinesLfalse%
2362 \writtenlinesRfalse%
2363 \get@familiarfootnote@number%
2364 %

```

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

```

2365 \let\print@Xnotes\print@Xnotes@forpages%
2366 \let\print@notesX\print@notesX@forpages%
2367 %

```

Check if there are chunks to be processed.

```

2368 \check@pstarts%

```

```
2369 \loop@if@pstarts%
2370 %
```

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

```
2371 \global\advance\l@dpscL \@ne%
2372 \global\advance\l@dpscR \@ne%
2373 %
```

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

```
2374 \getlinesfromparlistL%
2375 \getlinesfromparlistR%
2376 \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}%
2377 {\usernamecount{\l@dmaxlinesinpar\the\l@dpscL}}%
2378 \check@pstarts%
2379 \repeat%
2380 %
```

Zero the counts again, ready for the next bit.

```
2381 \global\l@dpscL=\z@%
2382 \global\l@dpscR=\z@%
2383 %
```

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

```
2384 \getlinesfrompagelistL%
2385 \getlinesfrompagelistR%
2386 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
2387 {\l@dminpagelines}%
2388 %
```

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.

```
2389 \check@pstarts%
2390 \if@pstarts%
2391 %
```

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

```
2392 \global\advance\l@dpscL \@ne%
2393 \global\advance\l@dpscR \@ne%
2394 \restore@pstartL@pc%
2395 \restore@pstartR@pc%
2396 %
```

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

```

2397 \global\@donereallinesL=\z@%
2398 \global\@donetotallinesL=\z@%
2399 \global\@donereallinesR=\z@%
2400 \global\@donetotallinesR=\z@%
2401 %

```

Start a loop over the boxes (chunks).

```

2402 \checkraw@text%
2403 %
2404 % \begingroup
2405 { \loop\ifaraw@text%
2406 %

```

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

```

2407 \checkpageL%
2408 \l@duselanguage{\theledlanguageL}%
2409 { \loop\ifl@dsamepage%
2410 %

```

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

```

2411 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}%
2412 \csuse{before@pstartL@the\l@dpscL}%
2413 \global\csundef{before@pstartL@the\l@dpscL}%
2414 \do@lineL%
2415 \xifinlist{the\l@dpscL}{\eled@sections@@}
2416 {\print@eledsectionL}%
2417 {}%
2418 \advance\numpagelinesL \@one%
2419 %

```

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`

```

2420 \ifshiftedpstarts%
2421 \ifdim\ht\l@dleftbox>0pt%
2422 \parledgroup@correction@notespacing{L}%
2423 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2424 \else%
2425 \unless\ifadvancedshiftedpstarts%
2426 \dimen0=\pagetotal%
2427 \advance\dimen0 by \baselineskip%
2428 \global\pagetotal=\dimen0%
2429 \else%

```

```

2430         \ifnomaxlines%
2431         \numdef{\@tmp}{\the\l@dpscL+1}%
2432         \ifcsdef{minpage@pstart@\@tmp}{%
2433         \ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}%
2434         {\dimen0=\pagetotal%
2435         \advance\dimen0 by \baselineskip%
2436         \global\pagetotal=\dimen0%
2437         }%
2438         {}%
2439         }%
2440         \fi%
2441         \fi%
2442         \fi%
2443     \else%
2444         \parledgroup@correction@notespacing{L}%
2445         \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2446         \fi%
2447 %

```

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

```

2448         \get@nextboxL%
2449         \global\l@dskipversenumberfalse%
2450         \ifprint@last@after@pendL%
2451         \csuse{after@pendL@\the\l@dpscL}%
2452         \global\csundef{after@pendL@\the\l@dpscL}%
2453         \fi%
2454         \checkpageL%
2455         \checkverseL%
2456         \checkpbl%
2457         \repeat%
2458 %

```

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.

```

2459         \ifl@dpagfull%
2460         \@writelinesonpageL{\the\numpagelinesL}%
2461         \else%
2462         \@writelinesonpageL{1000}%
2463         \fi%
2464 %

```

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.

```

2465 \numpagelinesL \z@%
2466 \parledgroup@correction@notespacing@init%
2467 \clearl@dleftpage }%
2468 %

```

Now do the same for the right text.

```

2469 \checkpageR%
2470 \l@duselanguage{\theledlanguageR}%
2471 {
2472 \loop\ifl@dsamepage%
2473 \set@sectcountR%
2474 \ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{}%
2475 \csuse{before@pstartR@the\l@dpscR}%
2476 \global\csundef{before@pstartR@the\l@dpscR}%
2477 \do@lineR%
2478 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
2479 {\print@eledsectionR}%
2480 {}%
2481 \advance\numpagelinesR \@ne%
2482 \ifshiftedpstarts%
2483 \ifdim\ht\l@drightbox>0pt%
2484 \parledgroup@correction@notespacing{R}%
2485 \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2486 \else%
2487 \unless\ifadvancedshiftedpstarts%
2488 \dimen0=\pagetotal%
2489 \advance\dimen0 by \baselineskip%
2490 \global\pagetotal=\dimen0%
2491 \else%
2492 \ifnomaxlines%
2493 \numdef{\@tmp}{\the\l@dpscR+1}%
2494 \ifcsdef{minpage@pstart@\@tmp}{%
minpage@pstart@\@tmp}%
2495 {\dimen0=\pagetotal%
2496 \advance\dimen0 by \baselineskip%
2497 \global\pagetotal=\dimen0%
2498 }%
2499 {}%
2500 }-%
2501 \fi%
2502 \fi%
2503 \fi%
2504 \else%
2505 \parledgroup@correction@notespacing{R}%
2506 \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2507 \fi%
2508 \get@nextboxR%
2509 \global\l@dskipversenumberRfalse%
2510 \ifprint@last@after@pendR%
2511 \csuse{after@pendR@the\l@dpscR}%

```

```

2512         \global\csundef{after@pendR@the\l@dpscR}%
2513         \fi%
2514         \checkpageR%
2515         \checkverseR%
2516         \checkpbR%
2517         \repeat%
2518         \ifl@pagefull%
2519         \@writelinesonpageR{\the\numpagelinesR}%
2520         \else%
2521         \@writelinesonpageR{1000}%
2522         \fi%
2523         \numpagelinesR=\z@%
2524         \parledgroup@correction@notespacing@init%
2525 %

```

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

```

2526         \clearl@drighthpage}%
2527 %

```

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.

```

2528         \checkdraw@text%
2529         \ifaraw@text%
2530         \getlinesfrompagelistL%
2531         \getlinesfrompagelistR%
2532         \l@dcalc@minoftwo{\cs@linesonpageL}{\cs@linesonpageR}%
2533         {\l@dminpagelines}%
2534         \fi%
2535         \repeat}%
2536 %

```

We have now output the text from all the chunks.

```

2537         \fi%
2538 %

```

Make sure that there are no inserts hanging around.

```

2539         \flush@notes%
2540         \flush@notesR%
2541         \endgroup%
2542 %

```

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

```

2543         \global\l@dpscL=\z@%
2544         \global\l@dpscR=\z@%
2545         \global\l@dnumstartsL=\z@%
2546         \global\l@dnumstartsR=\z@%
2547         \global\instanzaLfalse%
2548         \global\instanzaRfalse%

```

```

2549 \global\l@printingpagesfalse%
2550 \finish@Pages@notes%Needed to prevent final notes overlap line number
2551 \ignorespaces\fi}
2552
2553
2554 %

```

### XVIII.3 Ensure all notes be 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 is no more long notes left for the next pages.

```

2555 \newcommand{\finish@Pages@notes}{%
2556 \def\do##1{%
2557 %

```

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

```

2558 \ifnocritical{%
2559 \global\newnamebox{##1footins}%
2560 \fi
2561 \ifnofamiliar{%
2562 \global\newnamebox{footins##1}%
2563 \fi
2564 %

```

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

```

2565 \ifvoid\csuse{##1footins}%
2566 \ifvoid\csuse{footins##1}\else%
2567 \newpage\null%
2568 \listbreak%
2569 \fi%
2570 \else%
2571 \newpage\null%
2572 \listbreak%
2573 \fi%
2574 }%
2575 \dolistloop{@series}%
2576 }%
2577 %

```

### XVIII.4 Struts

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

```

\ledstrutR
2578 \newcommand*{\ledstrutL}{-}
2579 \newcommand*{\ledstrutR}{-}
2580
2581 %

```

### XVIII.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@devenpage` is similar except that it first checks to see if it is already on an empty page.

```

2582 \providecommand{\cleartoevenpage}[1][\@empty]{%
2583   \clearpage
2584   \ifodd\c@page\hbox{#1}\clearpage\fi}
2585
2586 \newcommand*\cleartol@devenpage{%
2587   \ifdim\pagetotal<\topskip% on an empty page
2588   \else
2589     \clearpage
2590     \Pages@mainmatter%
2591   \fi
2592   \ifodd\c@page%
2593     \ifprevpgnotnumbered%
2594       \addtocounter{par@page}{-1}%
2595       \ifdef\prevpgstyle{\thispagestyle{\prevpgstyle}}{}%
2596     \fi%
2597     \hbox{ }\clearpage%
2598   \fi%
2599 }%
2600 %

```

`\clearl@dleftpage` `\clearl@dleftpage` and `\clearl@drightpage` 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`

```

2601 \newcommand*\clearl@dleftpage{%
2602   \ifdim\pagetotal=0pt\hbox{ }\fi%
2603   \newpage%
2604   \insert@notes@for@onlyside%
2605   \ifodd\c@page\else
2606     \led@err@LeftOnRightPage
2607     \hbox{ }%
2608     \cleardoublepage
2609   \fi}
2610
2611 \newcommand*\clearl@drightpage{%
2612   \ifdim\pagetotal=0pt\hbox{ }\fi%
2613   \newpage%
2614   \insert@notes@for@onlyside%
2615   \ifodd\c@page
2616     \led@err@RightOnLeftPage
2617     \hbox{ }%

```

```

2618 \cleartoevenpage
2619 \fi}
2620
2621 %

```

## XVIII.6 Lines managing

`\getlinesfromparlistL` `\@cs@linesinparL` `\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` for `\getlinesfromparlistR`.

```

2622 \newcommand*{\getlinesfromparlistL}{%
2623 \ifx\linesinpar@listL\empty
2624 \gdef\@cs@linesinparL{0}%
2625 \else
2626 \gl@p\linesinpar@listL\to\@cs@linesinparL
2627 \fi}
2628 \newcommand*{\getlinesfromparlistR}{%
2629 \ifx\linesinpar@listR\empty
2630 \gdef\@cs@linesinparR{0}%
2631 \else
2632 \gl@p\linesinpar@listR\to\@cs@linesinparR
2633 \fi}
2634
2635 %

```

`\getlinesfrompagelistL` `\@cs@linesonpageL` `\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`.

```

2636 \newcommand*{\getlinesfrompagelistL}{%
2637 \ifx\linesonpage@listL\empty
2638 \gdef\@cs@linesonpageL{1000}%
2639 \else
2640 \gl@p\linesonpage@listL\to\@cs@linesonpageL
2641 \fi}
2642 \newcommand*{\getlinesfrompagelistR}{%
2643 \ifx\linesonpage@listR\empty
2644 \gdef\@cs@linesonpageR{1000}%
2645 \else
2646 \gl@p\linesonpage@listR\to\@cs@linesonpageR
2647 \fi}
2648
2649 %

```

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

```

2650 \newcommand*{\@writelinesonpageL}[1]{%

```

```

2651 \edef\next{\write\linenum@out{\string\@lopL{#1}}}%
2652 \next}
2653 \newcommand*{\@writelinesonpageR}[1]{%
2654 \edef\next{\write\linenum@outR{\string\@lopR{#1}}}%
2655 \next}
2656
2657 %

```

`\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⟩`.

```

2658 \newcommand*{\l@dcalc@maxoftwo}[3]{%
2659 \ifnum #2>#1\relax
2660 #3=#2\relax
2661 \else
2662 #3=#1\relax
2663 \fi}
2664 \newcommand*{\l@dcalc@minoftwo}[3]{%
2665 \ifnum #2<#1\relax
2666 #3=#2\relax
2667 \else
2668 #3=#1\relax
2669 \fi}
2670
2671 %

```

### XVIII.7 Page break managing

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

```

\checkpageL
\checkpageR
2672 \newif\ifl@dsamepage
2673 \l@dsamepagetrue
2674 \newif\ifl@dpagfull
2675
2676 \newcommand*{\checkpageL}{%
2677 \l@dpagfulltrue
2678 \l@dsamepagetrue
2679 \check@goal
2680 \ifdim\pagetotal<\ledthegoal
2681 \ifnum\numpagelinesL<\l@dminpagelines
2682 \else
2683 \ifnomaxlines%

```

```

2684     \else%
2685         \l@dsamepagefalse%
2686         \l@dpagefullfalse%
2687     \fi%
2688 \fi
2689 \else
2690     \l@dsamepagefalse
2691     \l@dpagefulltrue
2692 \fi%
2693 \ifprint@last@after@pendL%
2694     \l@dpagefullfalse%
2695     \l@dsamepagefalse%
2696     \print@last@after@pendLfalse%
2697 \fi%
2698 }%
2699
2700 \newcommand*{\checkpageR}{%
2701     \l@dpagefulltrue
2702     \l@dsamepagetrue
2703     \check@goal
2704     \ifdim\pagetotal<\ledthegoal
2705         \ifnum\numpagelinesR<\l@dmminpagelines
2706             \else
2707                 \ifnomaxlines%
2708                     \else%
2709                         \l@dsamepagefalse%
2710                         \l@dpagefullfalse%
2711                     \fi%
2712                 \fi
2713             \else
2714                 \l@dsamepagefalse
2715                 \l@dpagefulltrue
2716             \fi%
2717             \ifprint@last@after@pendR%
2718                 \l@dpagefullfalse%
2719                 \l@dsamepagefalse%
2720                 \print@last@after@pendRfalse%
2721             \fi%
2722         }%
2723
2724 %

```

**\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.

```

2725 \newcommand{\checkpbL}{
2726     \IfStrEq{\led@pb@setting}{after}{
2727         \xifinlistcs{\the\absline@num}{l@prev@pb}{\l@dpagefulltrue\
l@dsamepagefalse}{}

```

```

2728 \xifinlistcs{\the\absline@num}{l@prev@nopb}{\l@dpagfullfalse\
l@dsamepagetrue}{-}
2729 }{-}
2730 \IfStrEq{\led@pb@setting}{before}{
2731 \numdef{\next@absline}{\the\absline@num+1}
2732 \xifinlistcs{\next@absline}{l@prev@pb}{\l@dpagfulltrue\
l@dsamepagefalse}{-}
2733 \xifinlistcs{\next@absline}{l@prev@nopb}{\l@dpagfullfalse\
l@dsamepagetrue}{-}
2734 }{-}
2735 }
2736
2737 \newcommand{\checkpbR}{
2738 \IfStrEq{\led@pb@setting}{after}{
2739 \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\l@dpagfulltrue\
l@dsamepagefalse}{-}
2740 \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\l@dpagfullfalse\
l@dsamepagetrue}{-}
2741 }{-}
2742 \IfStrEq{\led@pb@setting}{before}{
2743 \numdef{\next@abslineR}{\the\absline@numR+1}
2744 \xifinlistcs{\next@abslineR}{l@prev@pbR}{\l@dpagfulltrue\
l@dsamepagefalse}{-}
2745 \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\l@dpagfullfalse\
l@dsamepagetrue}{-}
2746 }{-}
2747 }
2748 %

```

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

```

2749 \newcommand{\checkverseL}{
2750 \ifinstanzaL
2751 \iflednopbinverse
2752 \ifinserthangingsymbol
2753 \numgdef{\prev@abslineverse}{\the\absline@num-1}
2754 \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{-}
2755 \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesL<3\ledpbnum{\
prev@abslineverse}\fi}{-}
2756 \fi
2757 \fi
2758 \fi
2759 }
2760 \newcommand{\checkverseR}{
2761 \ifinstanzaR
2762 \iflednopbinverse
2763 \ifinserthangingsymbolR
2764 \numgdef{\prev@abslineverse}{\the\absline@numR-1}
2765 \IfStrEq{\led@pb@setting}{after}{\lednopbnumR{\prev@abslineverse}}{-}

```

```

2766 \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesR<3\ledpbnumR{\
prev@abslineverse}\fi}{\fi}
2767 \fi
2768 \fi
2769 \fi
2770 }
2771 %

```

`\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`.

```

\check@goal
2772 \newdimen\ledthegoal
2773 \ifshiftedpstarts
2774 \newcommand*\@goalfraction{0.95}
2775 \else
2776 \newcommand*\@goalfraction{0.9}
2777 \fi
2778
2779 \newcommand*\check@goal{%-
2780 \ledthegoal=\@goalfraction\pagegoal}
2781 \newcommand\setgoalfraction[1]{%-
2782 \xdef\@goalfraction{#1}%
2783 }
2784 %

```

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

```

\ifwrittenlinesL
2785 \newif\ifwrittenlinesL
2786 \newif\ifwrittenlinesR
2787
2788 %

```

## XVIII.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.

```

2789 \newif\if@getnextbox%
2790 %

```

`\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.

```

2791 \newcommand*\get@nextboxL}{%
2792   \ifvbox\namebox{1@dLcolrawbox\the\l@dpscl}% box is not empty
2793   %

```

The current box is not empty; do nothing.

```

2794   \else%                               box is empty
2795   %

```

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

```

2796   \@getnextboxtrue%Should be local, but be cautious
2797   %

```

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

```

2798   \ifnum\usenamecount{1@dmaxlinesinpar\the\l@dpscl}>\@donetotallinesL
2799   \parledgroup@notes@endL%
2800   \unless\ifnosyncpstarts%
2801   \getnextboxfalse%
2802   %

```

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.

```

2803   \ifnomaxlines%
2804   \ifdim\pagetotal<\ledthegoal%
2805   \numdef{\@tmp}{\l@dpscl+1}%
2806   \ifcsdef{afterlines@pstart@\@tmp R}{%
2807   \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp R}}
2808   {}%
2809   {\ifcsdef{minpage@pstart@\@tmp}%
2810   {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
2811   {\ifnum\numpagelinesL=\l@dminpagelines%
2812   \getnextboxtrue%
2813   \fi%
2814   }%
2815   {\@getnextboxtrue}}}%
2816   {\@getnextboxtrue}%
2817   }%
2818   }%
2819   {}%
2820   \fi%
2821   \fi%
2822   \else%
2823   \ifnomaxlines%
2824   \numdef{\@tmp}{\the\l@dpscl+1}%
2825   \ifcsdef{minpage@pstart@\@tmp}{%
2826   \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
2827   %

```

```

2828         {\ifdimgreater{\pagetotal}{\ledthegoal}%
2829             {\@getnextboxtrue}%
2830             {\@getnextboxfalse}%
2831         }%
2832         {\@getnextboxtrue}%
2833     }%
2834     \fi%
2835 \fi%
2836 %

```

Sufficient lines have been output.

```

2837     \if@getnextbox%
2838         \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscL}=\@donetotallinesL
2839             \parledgroup@notes@endL
2840         \fi
2841         \ifwrittenlinesL\else
2842 %

```

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

```

2843         \@writelinesinparL
2844         \writtenlinesLtrue
2845     \fi
2846     \ifnum\l@dnumstartsL>\l@dpscL
2847 %

```

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.

```

2848         \writtenlinesLfalse
2849         \ifbypstart@
2850             \global\line@num=0%
2851             \resetprevline@%
2852         \fi
2853 % Add the content of the optional argument of the previous \protect\cs{pend
2854 }%.
2854 % \begin{macrocode}
2855         \csuse{after@pendL@\the\l@dpscL}%
2856         \global\csundef{after@pendL@\the\l@dpscL}%
2857 %

```

Check the number of lines

```

2858         \l@dcalc@maxoftwo{\the\usernamecount{1@dmaxlinesinpar\the\l@dpscL}}%
2859             {\the\@donetotallinesL}%
2860             {\usernamecount{1@dmaxlinesinpar\the\l@dpscL}}%
2861         \global\@donetotallinesL \z@
2862 %

```

Go to the next pstart

```

2863 \global\advance\l@dpscL \@ne
2864 \global\pstartnumtrue%
2865 \restore@pstartL@pc%
2866 %

```

Add notes of parallel ledgroup.

```

2867 \parledgroup@notes@endL
2868 \parledgroup@correction@notes@spacing@final{L}
2869 \else
2870 %

```

```

2871 \print@last@after@pendLtrue%
2872 \fi
2873 \fi
2874 \fi}
2875 %

```

```

2876 \newcommand*{\get@nextboxR}{%
2877 \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}% box is not empty
2878 \else% box is empty
2879 \@getnextboxtrue%
2880 \ifnum\usenamecount{1@dmaxlinesinpar\the\l@dpscR}>\@donetotallinesR
2881 \parledgroup@notes@endR
2882 \unless\ifnosyncpstarts%
2883 \@getnextboxfalse%
2884 \ifnomaxlines%
2885 \ifdim\pagetotal<\ledthegoal%
2886 \numdef{\@tmp}{\l@dpscR+1}%
2887 \ifcsdef{afterlines@pstart@\@tmp L}{%
2888 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp L}}
2889 }%
2890 {\ifcsdef{minpage@pstart@\@tmp}%
2891 {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2892 {\ifnum\numpagelinesR=\l@dminpagelines%
2893 \@getnextboxtrue%
2894 \fi%
2895 }%
2896 {\@getnextboxtrue}}%
2897 {\@getnextboxtrue}%
2898 }%
2899 }%
2900 }%
2901 \fi%
2902 \fi%
2903 \fi%
2904 \else%
2905 \ifnomaxlines%
2906 \numdef{\@tmp}{\the\l@dpscR+1}%
2907 \ifcsdef{minpage@pstart@\@tmp}{%

```

```

2908     \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2909     {\ifdimgreater{\pagetotal}{\ledthegoal}%
2910      {\@getnextboxtrue}%
2911      {\@getnextboxfalse}%
2912     }%
2913     {\@getnextboxtrue}%
2914   }{}
2915   \fi%
2916 \fi%
2917 \if@getnextbox%
2918   \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscR}=\@donetotallinesR
2919     \parledgroup@notes@endR
2920     \fi
2921   \ifwrittenlinesR\else
2922     \@writelinesinparR
2923     \writtenlinesRtrue
2924     \fi
2925   \ifnum\l@dnumpstartsR>\l@dpscR
2926     \writtenlinesRfalse
2927     \ifbypstart@R
2928       \global\line@numR=0%
2929       \resetprevline@%
2930     \fi
2931     \csuse{after@pendR@\the\l@dpscR}%
2932     \global\csundef{after@pendR@\the\l@dpscR}%
2933     \l@dcalcm@maxoftwo{\the\usernamecount{1@dmaxlinesinpar\the\l@dpscR}}%
2934       {\the\@donetotallinesR}%
2935       {\usernamecount{1@dmaxlinesinpar\the\l@dpscR}}%
2936     \global\@donetotallinesR \z@
2937     \global\advance\l@dpscR \@ne
2938     \global\pstartnumRtrue%
2939     \restore@pstartR@pc%
2940     \parledgroup@notes@endR
2941     \parledgroup@correction@notespacing@final{R}
2942   \else
2943     \print@last@after@pendRtrue%
2944   \fi
2945 \fi
2946 \fi}
2947
2948 %

```

## XIX Page numbering

### XIX.1 Global options

The `sameparallelpagenumber` 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  $\LaTeX$  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  $\LaTeX$ '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  $\LaTeX$  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.

```

2949
2950 \newcommand{\par@patch@thepage}{%
2951   \ifboolexpr{%
2952     bool{sameparallelpagenumber}%
2953     or bool{prevpgnotnumbered}%
2954   }%
2955   {%
2956     \patchcmd{\thepage}%
2957       {page}{par@page}%
2958       {}%
2959       {\led@error@fail@patch@thepage}%
2960   }{}%
2961 }%
2962
2963 \newcommand{\par@patch@pagenumbering}{%
2964   \ifboolexpr{%
2965     bool{sameparallelpagenumber}%
2966     or bool{prevpgnotnumbered}%
2967   }%
2968   {%
2969     \setcounter{par@page}{1}%
2970   }%
2971   {}%
2972   \par@patch@thepage%
2973 }%
2974
2975 \ifl@dmemoir%
2976   \apptocmd{\@mempnum}%
2977     {\par@patch@pagenumbering}%
2978     {}%

```

```

2979     {\led@error@fail@patch@mempnum}%
2980
2981 \else%
2982   \apptocmd{\pagenumbering}%
2983     {\par@patch@pagenumbering}%
2984     {}%
2985     {\led@error@fail@patch@pagenumbering}%
2986 \fi%
2987
2988 \AtBeginDocument{\par@patch@thepage}%
2989 %

```

`\@outputpage` As its name says, `\@outputpage` is a  $\LaTeX$ '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.

```

2990 \AtBeginDocument{%
2991   \apptocmd{\@outputpage}{%
2992     \ifsameparallelpagelength%
2993       \ifl@dprintingpages%
2994         \ifodd\c@page\else%
2995           \stepcounter{par@page}%
2996         \fi%
2997       \else%
2998         \stepcounter{par@page}%
2999       \fi%
3000     \else%
3001       \stepcounter{par@page}%
3002     \fi%
3003   }%
3004   {}%
3005   {\led@error@fail@patch@outputpage}%
3006 }
3007 %

```

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

```

3008 \newcounter{par@page}%
3009 \setcounter{par@page}{1}%
3010 %

```

## XIX.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 \newif\ifPages@mainmatter
\Pages@mainmatter \newcommand{\Pages@mainmatter}{%
3013   \ifPages@mainmatter%
3014   \pagenumbering{arabic}%
3015   \addtocounter{page}{1}%
3016   \addtocounter{par@page}{-1}%
3017   \patchcmd{\thepage}{page}{par@page}{-}{-}%
3018   \fi%
3019 }
3020 %

```

## XX 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 \pretocmd{\chapter}{%
3022   \ifl@printingpages%
3023   \vbox{}%
3024   \fi%
3025 }%
3026 {}%
3027 {}%
3028 %

```

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

```

3029 \newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}}
3030 \eledsectnotoc{R}
3031 %

```

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

```

3032 \newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}}
3033 \eledsectmark{L}
3034 %

```

`\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`.

```

3035 \newskip\eledsection@correcting@skip
3036 %

```

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

```
3037 \newwrite\eled@sectioningR@out
3038 %
```

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

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

`\l@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).

```
3039 \def\l@prev@pbR{}
3040 \def\l@prev@nopbR{}
3041 %
```

`\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.

```
3042 \newcommand{\ledpbR}{\write\linenum@outR{\string\led@pbR}}
3043 \newcommand{\ledpbnumR}[1]{\write\linenum@outR{\string\led@pbnumR{#1}}}
3044 \newcommand{\lednopbR}{\write\linenum@outR{\string\led@nopbR}}
3045 \newcommand{\lednopbnumR}[1]{\write\linenum@outR{\string\led@nopbnumR{#1}}}
3046 %
```

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

```
3047 \newcommand{\led@pbR}{\listxadd{\l@prev@pbR}{\the\absline@numR}}
3048 \newcommand{\led@pbnumR}[1]{\listxadd{\l@prev@pbR}{#1}}
3049 \newcommand{\led@nopbR}{\listxadd{\l@prev@nopbR}{\the\absline@numR}}
3050 \newcommand{\led@nopbnumR}[1]{\listxadd{\l@prev@nopbR}{#1}}
3051 %
```

## XXII 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` `\parledgroup@type` contains the type of the footnote: critical (Xfootnote) or familiar (footnoteX).

```

3052 \newmarks\parledgroup@
3053 \newmarks\parledgroup@series
3054 \newmarks\parledgroup@type
3055 %

```

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

```

3056 \newcommand{\parledgroup@notes@startL}{%
3057   \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscL}>0%
3058   \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\cuse{
3059     \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\cuse{
3060       \hooknoteX@\splitfirstmarks\parledgroup@series}}{}}%
3061   \fi%
3062   \global\ledgroupnotesL@true%
3063   \insert@noterule@ledgroup{L}%
3064 }
3065 \newcommand{\parledgroup@notes@startR}{%
3066   \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscR}>0%
3067   \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\cuse{
3068     \hooknoteX@\splitfirstmarks\parledgroup@series}}{}}%
3069   \fi%
3070   \global\ledgroupnotesR@true%
3071   \insert@noterule@ledgroup{R}%
3072 }
3073 %

```

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

```

3073 \newcommand{\parledgroup@notes@endL}{%
3074   \global\ledgroupnotesL@false%
3075 }
3076 \newcommand{\parledgroup@notes@endR}{%
3077   \global\ledgroupnotesR@false%
3078 }
3079 %

```

`\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`

```

3080 \newcommand{\insert@noterule@ledgroup}[1]{
3081   \IfStrEq{\splitbotmarks\parledgroup@}{begin}{%
3082     \IfStrEq{\splitbotmarks\parledgroup@type}{Xfootnote}{
3083       \cuse{ifledgroupnotes#1@}
3084       \vskip\skip\cuse{mp\splitbotmarks\parledgroup@series footins}
3085       \cuse{\splitbotmarks\parledgroup@series footnoterule}

```

```

3086     \fi
3087   }
3088   {}
3089   \IfStrEq{\splitbotmarks\parledgroup@type}{footnoteX}{
3090     \csuse{ifledgroupnotes#1@}
3091     \vskip\skip\csuse{mpfootins\splitbotmarks\parledgroup@series}
3092     \csuse{footnoterule\splitbotmarks\parledgroup@series}
3093     \fi
3094   }{}
3095   }
3096 {}
3097 }
3098 %

```

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

```

3099 \newcommand{\setparledgroupnotespacing}[1]{\gdef\@parledgroupnotespacing
3100   {#1}}
3101 \newcommand{\@parledgroupnotespacing}{}
3102 %

```

`\parledgroup@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`.

```

3102 \dimdef{\parledgroup@notespacing@correction}{0pt}
3103 \newcommand{\parledgroup@notespacing@set@correction}{%
3104   {\@getfirstseries\csuse{Xnotefontsize@\@firstseries}%We suppose all the
3105   series has the same footnote size setup
3106   \parledgroup@notespacing\dimgdef{\temp@spacing}{\baselineskip}}%
3107   \dimgdef{\parledgroup@notespacing@correction}{\baselineskip-\temp@spacing}
3108   }%
3109 }
3110 %

```

`\parledgroup@correction@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.

```

3109 \newcommand{\parledgroup@correction@notespacing@init}{
3110   \dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}
3111   \dimdef{\parledgroup@notespacing@correction@modulo}{0pt}
3112 }
3113 \parledgroup@correction@notespacing@init
3114 %

```

`\parledgroup@correction@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.

```

3115 \newcommand{\parledgroup@correction@notespacing@final}[1]{
3116   \ifparledgroup
3117   \vspace{\parledgroup@notespacing@correction@accumulated}
3118   \parledgroup@correction@notespacing@init%
3119   \ifstrequal{#1}{L}{
3120     \numdef{\@checking}{\the\l@dpscL-1}
3121   }{
3122     \numdef{\@checking}{\the\l@dpscR-1}
3123   }
3124   \dimdef{\@beforenotes@current@diff}{\csuse{\parledgroup@beforenotes@
3125 @checking L}-\csuse{\parledgroup@beforenotes@\@checking R}}%
3126   \ifstrequal{#1}{L}%
3127     {% Left
3128       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{-\
3129 @beforenotes@current@diff}}%
3130     }{% Right
3131       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{\
3132 @beforenotes@current@diff}}{}
3133     }%
3134   \fi
3135 }
3136 %

```

`\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.

```

3135 \newcommand{\parledgroup@correction@notespacing}[1]{%
3136   \csuse{ifledgroupnotes#1@}%
3137   \vspace{-\parledgroup@notespacing@correction}%
3138   \dimdef{\parledgroup@notespacing@correction@accumulated}{\
3139 parledgroup@notespacing@correction@accumulated+
3140 parledgroup@notespacing@correction}%
3141   \dimdef{\parledgroup@notespacing@correction@modulo}{\
3142 parledgroup@notespacing@correction@modulo+
3143 parledgroup@notespacing@correction}%

```

```

3140 \ifdimless{\parledgroup@notespacing@correction@modulo}{\baselineskip
}{}{\advance\numpagelinesL -\@ne%
3141 \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo-\baselineskip}%
3142 }% mean greater than equal
3143 \fi%
3144 }
3145 %

```

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

```

3146 \dimdef\parledgroup@beforenotesL{0pt}
3147 \dimdef\parledgroup@beforenotesR{0pt}
3148 %

```

`\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.

```

3149 \newcommand{\parledgroup@beforenotes@save}[1]{
3150 \ifparledgroup
3151 \csdimgdef{@parledgroup@beforenotes@the\csuse{lednumstarts#1}#1}{\
csuse{\parledgroup@beforenotes#1}}
3152 \csdimgdef{\parledgroup@beforenotes#1}{0pt}
3153 \fi
3154 }
3155 %

```

## XXIII Compatibility with eledmac

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

```

3156 \ifeledmaccompat%
3157
3158
3159 \unless\ifnocritical@
3160 \let\onlyXside\Xonlyside
3161 \fi
3162 \fi
3163 %

```

## XXIV 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. 341.

**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>\@pstartfalse</code> .....	1
<code>\@pstartstrue</code> .....	1
<code>\@ref</code> .....	1
<code>\@ref@regR</code> .....	1
<code>\@set</code> .....	1
<code>\@stopastanza</code> .....	1
<code>\@writelinesinparL</code> .....	1

\@writelinesinparR	1
\@writelinesonpageL	1
\@writelinesonpageR	1
\@writepageofparL	1
\@writepageofparR	1
CLASSmemoir	110
COMMAND\+	48
COMMAND\@Rlineflag	74, 145
COMMAND\@adv	38, 144
COMMAND\@cs@linesinparL	101
COMMAND\@cs@linesonpageL	101
COMMAND\@eledsectionL	84
COMMAND\@eledsectionR	84
COMMAND\@eledsectmark	112
COMMAND\@eledsectnotoc	112
COMMAND\@footnotetext	76
COMMAND\@goalfraction	12, 105
COMMAND\@l@dtempcnta	70
COMMAND\@lab	37, 75, 144
COMMAND\@lopL	43, 95, 101
COMMAND\@lopR	43, 101
COMMAND\@mempnum	110
COMMAND\@namedef	26
COMMAND\@namuse	26
COMMAND\@nl	37, 44, 75, 144
COMMAND\@nl@regR	37
COMMAND\@outputpage	110, 111
COMMAND\@page	75
COMMAND\@par@sync@option	37
COMMAND\@parledgroupnotespacing	115
COMMAND\@pend	42
COMMAND\@pendR	42
COMMAND\@pstart	42
COMMAND\@pstartstrue	83
COMMAND\@ref	40, 41, 44, 144
COMMAND\@ref@regR	40
COMMAND\@set	39, 144
COMMAND\@sw	41
COMMAND\AtBeginPairs	8, 50, 142
COMMAND\AtEveryPend	142, 144
COMMAND\AtEveryPstart	2, 17, 57, 85, 142, 144
COMMAND\AtEveryPstartCall	2, 17, 57, 143
COMMAND\AtEveryStanza	146
COMMAND\AtEveryStopStanza	146
COMMAND\Clear the right lines for \read@linelist	36
COMMAND\Columns	7, 8, 17, 49, 84, 88, 139–141, 144, 145
COMMAND\Columns@print@after@pend	91
COMMAND\Columns@print@before@pstart	91
COMMAND\Colwidth	8, 10, 93
COMMAND\Leftsidehook	139

COMMAND\Leftsidehookend	139
COMMAND\MakePerPage	13
COMMAND\Pages	4, 7, 10, 12, 17, 48, 49, 68, 71, 72, 74, 92, 93, 99, 110, 111, 115, 139, 141, 143–145
COMMAND\Pages@mainmatter	111
COMMAND\Pairs	48
COMMAND\Rcolwidth	8, 10, 93
COMMAND\Rightsidehook	139
COMMAND\Rightsidehookend	139
COMMAND\Rlineflag	118
COMMAND\Xendlineflag	15, 145
COMMAND\Xlineflag	15, 119, 145
COMMAND\Xmaxhnotes	13
COMMAND\Xnoteswidthliketwocolumns	9, 142
COMMAND\Xonlyside	14, 71, 119, 146
COMMAND\Xtxtbeforenotes	146
COMMAND\&	18
COMMAND\absline@numR	36
COMMAND\add@penalties	70
COMMAND\add@penaltiesL	70
COMMAND\advanceline	38, 44, 144
COMMAND\affixline@num	65
COMMAND\affixline@numR	65, 139, 140
COMMAND\affixpstart@num	68
COMMAND\affixpstart@numR	68
COMMAND\affixside@note	75
COMMAND\aftercolumnseparator	9, 88, 142
COMMAND\araw@textfalse	83
COMMAND\araw@texttrue	83
COMMAND\at@begin@pairs	50
COMMAND\autopar	17
COMMAND\ballast@count	70
COMMAND\baselineskip	88, 116
COMMAND\bbf@set@language	79, 80, 144
COMMAND\beforecolumnseparator	9, 88, 142
COMMAND\begin	18
COMMAND\beginnumbering	15–17, 27, 37, 89, 140, 142, 144
COMMAND\beginnumberingR	31, 43
COMMAND\bf	140
COMMAND\bfseries	140
COMMAND\brokenpenalty	70
COMMAND\chapter	112, 139
COMMAND\check@goal	105
COMMAND\check@pstarts	83
COMMAND\checkpageL	95, 102
COMMAND\checkpb@columns	87
COMMAND\checkpbL	103
COMMAND\checkpbR	103
COMMAND\checkraw@text	83
COMMAND\checkverseL	104
COMMAND\checkverseR	104

COMMAND\clear(double)page	100
COMMAND\clearl@dleftpage	100
COMMAND\clearl@drightpage	100
COMMAND\clearpage	100, 143
COMMAND\cleartoevenpage	100
COMMAND\cleartol@devenpage	100, 111
COMMAND\columnrulewidth	8, 88
COMMAND\columns@position	88
COMMAND\columnseparator	8, 9
COMMAND\columnspan	9, 142
COMMAND\correct@Xfootins@box	143
COMMAND\correct@footinsX@box	143
COMMAND\critext	143
COMMAND\csname	49
COMMAND\displaywidowpenalty	70
COMMAND\do@actions	63
COMMAND\do@actions@fixedcode	139
COMMAND\do@actions@nextR	64
COMMAND\do@actionsR	64, 139
COMMAND\do@ballast	70
COMMAND\do@ballastR	63
COMMAND\do@insidelineLhook	141
COMMAND\do@insidelineRhook	141
COMMAND\do@line	57
COMMAND\do@line(L/R)	61
COMMAND\do@lineL	58, 70, 139, 140
COMMAND\do@lineLhook	139
COMMAND\do@lineR	61, 139, 140, 142
COMMAND\do@lineRhook	139
COMMAND\do@lockoff	144
COMMAND\do@lockoffR	39
COMMAND\do@lockon	144
COMMAND\do@lockonR	39
COMMAND\doinsidelineLhook	142
COMMAND\doinsidelineRhook	142
COMMAND\dolineLhook	142
COMMAND\dolineRhook	142
COMMAND\edindex	143
COMMAND\edlabel	140, 143
COMMAND\edtext	40, 44, 45, 142, 143
COMMAND\eled@sectioningR@out	113
COMMAND\eledchapter	143
COMMAND\eledsection	142, 143, 145
COMMAND\eledsection@correcting@skip	112
COMMAND\eledsectmark	20, 112
COMMAND\eledsectnotoc	20, 112
COMMAND\eledxxx	142
COMMAND\end	18
COMMAND\endgraf	56
COMMAND\endlock	44, 144

COMMAND\endnumbering	9, 15, 17, 28, 144
COMMAND\endsub	44, 144
COMMAND\endumbering	15
COMMAND\expandafter	47
COMMAND\extensionchars	27
COMMAND\firstlinenum	16, 141, 144, 145
COMMAND\firstsublinenum	141, 144, 145
COMMAND\fix@page	38, 144
COMMAND\flag@end	44, 139, 142
COMMAND\flag@start	44, 142
COMMAND\flush@notesR	71
COMMAND\footnote	49
COMMAND\footnoteX	46
COMMAND\footnoteX@reading	146
COMMAND\footnoteXmk	14
COMMAND\footnoteXnomk	14, 46
COMMAND\frontmatter	12, 20
COMMAND\get@nextboxL	140
COMMAND\get@nextboxR	140
COMMAND\getline@numL	62
COMMAND\getline@numR	62
COMMAND\getlinesfrompagelistL	101
COMMAND\getlinesfrompagelistR	101
COMMAND\getlinesfromparlistL	101
COMMAND\getlinesfromparlistR	101
COMMAND\gl@p	47
COMMAND\goalfraction	118
COMMAND\hangingsymbol	118, 140
COMMAND\hfill	87, 88
COMMAND\hidenumbering	16, 144
COMMAND\if@getnextbox	105
COMMAND\ifPages@mainmatter	111
COMMAND\ifbypage@	145
COMMAND\ifbypstart@R	145
COMMAND\ifdim	87
COMMAND\ifinserthangingsymbol	77
COMMAND\ifinserthangingsymbolR	77
COMMAND\ifl@dpagefull	102
COMMAND\ifl@dpaging	24, 142
COMMAND\ifl@dpairing	24, 139
COMMAND\ifl@dsamelang	142
COMMAND\ifl@dsamepage	102
COMMAND\ifl@pagefull	95
COMMAND\ifledRcol	24
COMMAND\iflinenumberLevenifblank	35
COMMAND\iflinenumberRevenifblank	35
COMMAND\iflledRcol	140
COMMAND\ifnumberedpar@	53
COMMAND\ifnumberingR	140
COMMAND\ifnumberpstart	49

COMMAND\ifpst@rtedL	27, 28, 54, 139
COMMAND\ifpst@rtedR	27
COMMAND\ifsublines@	39
COMMAND\insert@countR	40
COMMAND\insert@noterule@ledgroup	114
COMMAND\insert@notes@for@onlyside	74
COMMAND\insertlines@list	40
COMMAND\insertlines@listR	40
COMMAND\inserts@list	53
COMMAND\inserts@listR	69
COMMAND\l@d@nums	45
COMMAND\l@d@set	39, 44, 144
COMMAND\l@dLcolrawbox	53
COMMAND\l@dLcolrawbox1	82
COMMAND\l@dLcolrawbox2	82
COMMAND\l@dRcolrawbox	53
COMMAND\l@dbfnote	76, 144
COMMAND\l@dcalc@maxoftwo	102
COMMAND\l@dcalc@minoftwo	102
COMMAND\l@dchecklang	139, 141
COMMAND\l@dcsnote	142
COMMAND\l@dleftbox	57, 95, 143
COMMAND\l@dlinenumR	35, 139
COMMAND\l@dlsnote	142
COMMAND\l@dmake@labels	74
COMMAND\l@dmaxlinesinpar	94
COMMAND\l@dmaxlinesinpar1	82
COMMAND\l@dminpagelines	94, 139
COMMAND\l@dnumpstartsL	81, 139
COMMAND\l@dprintingcolumnstrue	143
COMMAND\l@dprintingpagestrue	143
COMMAND\l@dpscL	85, 94, 107
COMMAND\l@dpscR	85, 94
COMMAND\l@drsnote	142
COMMAND\l@dsetupmaxlinecounts	82
COMMAND\l@duselanguage	80, 139
COMMAND\l@dzeromaxlinecounts	82
COMMAND\l@prev@nopbR	113
COMMAND\l@prev@pbR	113
COMMAND\labelpstarttrue	140
COMMAND\labelref@list	75
COMMAND\labelref@listR	74
COMMAND\lang	79
COMMAND\last@page@numR	38
COMMAND\led	140
COMMAND\led@nopbR	113
COMMAND\led@nopbnumR	113
COMMAND\led@pbR	113
COMMAND\led@pbnumR	113
COMMAND\ledinnerrote	19

COMMAND\ledleftnote	19
COMMAND\lednopb	19, 87, 103
COMMAND\lednopbR	113
COMMAND\lednopbnumR	113
COMMAND\ledouterote	19
COMMAND\ledpb	87, 103
COMMAND\ledpbR	113
COMMAND\ledpbnumR	113
COMMAND\ledrightnote	19
COMMAND\ledsidenote	19
COMMAND\ledstrutL	139
COMMAND\ledstrutR	139, 145
COMMAND\ledthegoal	105
COMMAND\ledtrutL	139, 145
COMMAND\leftlinenumR	35, 139
COMMAND\let	47
COMMAND\line@list@R	41
COMMAND\line@list@stuff	37, 43
COMMAND\line@margin	32
COMMAND\line@marginR	32, 139
COMMAND\line@numR	36
COMMAND\lineation	16, 143
COMMAND\lineation*	16, 32, 142
COMMAND\lineationR	16, 31, 143
COMMAND\linenum@out	75
COMMAND\linenum@outR	43
COMMAND\linenumberLevenifblanktrue	17, 146
COMMAND\linenumberRevenifblank	146
COMMAND\linenumberRevenifblanktrue	17
COMMAND\linenumberstyle	16, 119
COMMAND\linenumberstyle*	119
COMMAND\linenumberstyleR	16, 119
COMMAND\linenumincrement	16, 141, 144, 145
COMMAND\linenummargin	16, 32, 139, 144–146
COMMAND\linenummargin*	16, 32, 145
COMMAND\linenummarginR	16, 32, 145
COMMAND\linenumrepR	34, 139
COMMAND\linesinpar@listL	42, 101
COMMAND\linesonpage@listL	43, 101
COMMAND\lock@off	39
COMMAND\lock@on	39
COMMAND\mainmatter	2, 12, 20, 145
COMMAND\makeatletter	60
COMMAND\maxchunks	7, 18, 82
COMMAND\maxdimen	72
COMMAND\maxhnotesX	13
COMMAND\memorydump	15, 30
COMMAND\n@num	143
COMMAND\new@lineL	44
COMMAND\new@lineR	44

COMMAND\newhookcommand@series	48
COMMAND\newif	143
COMMAND\newpage	99, 100, 143
COMMAND\newseries	48
COMMAND\newseries@	45
COMMAND\newseries@par	45, 47, 48
COMMAND\noeledxxx	142
COMMAND\nomark@	46
COMMAND\nomaxlines	42
COMMAND\normalbfnoteX	139, 144
COMMAND\notesXwidthliketwocolumns	9, 142
COMMAND\num@lines	70
COMMAND\num@lines(R)	53
COMMAND\numberingR	29
COMMAND\numberlinefalse	7
COMMAND\numberonlyfirstinline	140
COMMAND\numberpstartfalse	16
COMMAND\numberpstarttrue	16, 140, 145
COMMAND\one@line	53, 77
COMMAND\one@lineR	53
COMMAND\onlyXside	119
COMMAND\onlysideX	14, 71, 74, 144, 146
COMMAND\otherlanguage	144
COMMAND\page@action	39, 144
COMMAND\pagenumbering	110, 145
COMMAND\pages	12
COMMAND\pagetotal	95, 143
COMMAND\par@line	70
COMMAND\par@line(R)	53
COMMAND\par@patch@pagenumbering	110
COMMAND\par@patch@thepage	110
COMMAND\par@sync@option	22
COMMAND\parledgroup@	113
COMMAND\parledgroup@beforenotes@save	117
COMMAND\parledgroup@beforenotesL	117
COMMAND\parledgroup@beforenotesR	117
COMMAND\parledgroup@correction@notespacing	116
COMMAND\parledgroup@correction@notespacing@final	115
COMMAND\parledgroup@correction@notespacing@init	115
COMMAND\parledgroup@notes@endL	114
COMMAND\parledgroup@notes@endR	114
COMMAND\parledgroup@notes@startL	114
COMMAND\parledgroup@notes@startR	114
COMMAND\parledgroup@notespacing@correction	115, 116
COMMAND\parledgroup@notespacing@correction@accumulated	116
COMMAND\parledgroup@notespacing@correction@modulo	116
COMMAND\parledgroup@notespacing@set@correction	115
COMMAND\parledgroup@series	113
COMMAND\parledgroup@type	113
COMMAND\parledgroupnotespacing	118

COMMAND\parledgrouptrue	19
COMMAND\patchcmd	145
COMMAND\pausenumbering	9, 30, 146
COMMAND\pend	4, 7, 10, 17–19, 49, 53, 56, 57, 81, 91, 142, 144, 145
COMMAND\pendL	141
COMMAND\pendR	141
COMMAND\pends	17
COMMAND\perpage	13
COMMAND\prev@nopbR	113
COMMAND\prev@pbR	113
COMMAND\prevpgstyle	23
COMMAND\print@Xnotes	71
COMMAND\print@Xnotes@forpages	71, 73, 143
COMMAND\print@columnseparator	87, 141
COMMAND\print@eledsectionL	60
COMMAND\print@line	59
COMMAND\print@lineL	59
COMMAND\print@notesX@forpages	143
COMMAND\printlinenumR	119
COMMAND\printlinesR	145
COMMAND\pstart	4, 7, 10, 16–19, 31, 44, 49, 53, 54, 56, 57, 81, 85, 91, 140, 142, 144, 145
COMMAND\pstartL	57, 141
COMMAND\pstartR	57, 140, 141
COMMAND\pstartinfootnote	143
COMMAND\raw@text	81
COMMAND\read@linelist	36, 37, 121, 144
COMMAND\ref@reg	40
COMMAND\ref@regR	40, 144
COMMAND\relax	119
COMMAND\reledmac	144
COMMAND\renewcommand	118
COMMAND\resumenummering	9, 30, 142, 146
COMMAND\resumenummeringR	31, 142
COMMAND\rightlinenumR	35, 139
COMMAND\section	139
COMMAND\section@num	27
COMMAND\selectlanguage	18, 79, 80
COMMAND\set@continuousnumberingforR	31
COMMAND\set@line	45, 144
COMMAND\set@line@action	39, 144
COMMAND\setRlineflag	17, 118
COMMAND\setgoalfraction	12, 118
COMMAND\sethangingsymbol	18
COMMAND\setline	39, 44, 144
COMMAND\setlinenum	39, 44, 144
COMMAND\setnoteposition...	89
COMMAND\setparledgroupnotespacing	118, 145
COMMAND\setposition...	89
COMMAND\setstanzaindents	9, 18, 118
COMMAND\setwidth...	89

COMMAND\sidenotemargin	19, 141
COMMAND\sidenotemargin*	19, 142
COMMAND\skipnumbering	16, 143
COMMAND\sloppy	8
COMMAND\stanza	7, 9, 16, 18, 51, 77, 119, 140
COMMAND\stanzanumtrue	19
COMMAND\startlock	44, 144
COMMAND\startsub	44, 144
COMMAND\sub@action	39, 144
COMMAND\sub@off	75
COMMAND\sub@on	75
COMMAND\subline@numR	36
COMMAND\sublinenumberstyle	16, 119
COMMAND\sublinenumberstyle*	119
COMMAND\sublinenumberstyleR	16, 119
COMMAND\sublinenumincrement	141, 144, 145
COMMAND\sublinenumrepR	34, 139
COMMAND\syntaxonly	146
COMMAND\sza@0@	18
COMMAND\textheight	13
COMMAND\textwidth	50
COMMAND\the@labelX	145
COMMAND\thefootnoteX	141
COMMAND\theledlanguageL	80
COMMAND\theledlanguageR	80
COMMAND\thepage	20, 110
COMMAND\thepstartL	16, 140
COMMAND\thepstartR	16, 140
COMMAND\thestanzaL	19
COMMAND\thestanzaR	19
COMMAND\vbox	54
COMMAND\vl@dbfnote	76
COMMAND\vskip	114
COMMAND\vsplit	70
COMMAND\widthliketwocolumns	9
COMMAND\widthliketwocolumnsfalse	9
COMMAND\widthliketwocolumnstrue	9
COMMAND\xflagref	145
COMMAND\xright@appenditem	47
COMMAND\xspace	21
COMMAND\xxxfootstart	89
COMMAND\xxxmatter	110
ENVIRONMENTLeftside	51
ENVIRONMENTRightside	52
ENVIRONMENTastanza	18, 77, 78, 119, 145
ENVIRONMENTcolumns	29, 146
ENVIRONMENTledgroup	6, 146
ENVIRONMENTleft	16
ENVIRONMENTpages	29, 50, 145, 146
ENVIRONMENTpairs	50, 145

PACKAGEEDMAC	120
PACKAGEEDSTANZA	120
PACKAGEEledmac	143
PACKAGEEledpar	143
PACKAGETABMAC	120
PACKAGEbabel	18, 79, 80, 146
PACKAGEedmac	120
PACKAGEeledmac	5, 81, 117, 118, 120, 141, 142, 144, 145
PACKAGEeledpar	5, 6, 13, 33, 118, 119, 141–143
PACKAGEtoolbox	87, 113
PACKAGEledmac	6
PACKAGEledpar	1, 6
PACKAGEMemoir	120
PACKAGEMusixtex	142
PACKAGEperpage	2, 13, 119, 146
PACKAGEpolyglossia	18, 79–81, 146
PACKAGEreledmac	1, 3, 5–7, 9, 13, 16–21, 23, 24, 27, 28, 32, 33, 35–40, 42–46, 48, 49, 59, 68, 74–77, 93, 99, 118, 119, 144–146
PACKAGEreledpar	1, 3, 5–7, 9–12, 18–23, 31, 35, 36, 42, 43, 45, 48, 49, 74, 118, 119, 145
PACKAGEsetspace	2, 20
PACKAGESyntonly	146
PACKAGEXkeyval	21
PACKAGEXR	146

## A

<code>\absline@numR</code>	1
<code>\actionlines@listR</code>	1
<code>\actions@listR</code>	1
<code>\add@inserts@nextR</code>	1
<code>\add@insertsR</code>	1
<code>\add@penaltiesL</code>	1
<code>\add@penaltiesR</code>	1
<code>\advanceline</code>	1
<code>\affixline@numR</code>	1
<code>\affixpstart@numL</code>	1
<code>\affixpstart@numR</code>	1
<code>\affixside@noteR</code>	1
<code>\aftercolumnseparator</code>	1, 9
<code>\araw@textfalse</code>	1
<code>\araw@texttrue</code>	1
astanza (environment)	18
<code>\AtBeginPairs</code>	1, 8
<code>\AtEveryPstartCall</code>	1
<code>\autopar</code>	17

## B

<code>\bbl@set@language</code>	1
<code>\beforecolumnseparator</code>	1, 9
<code>\beginnumbering</code>	15
<code>\beginnumberingR</code>	1

## C

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

## D

<code>\do@actions@fixedcodeR</code> .....	1
<code>\do@actions@nextR</code> .....	1
<code>\do@actionsR</code> .....	1
<code>\do@ballastR</code> .....	1
<code>\do@insidelineLhook</code> .....	1
<code>\do@insidelineRhook</code> .....	1
<code>\do@lineL</code> .....	1
<code>\do@lineLhook</code> .....	1
<code>\do@lineR</code> .....	1
<code>\do@lineRhook</code> .....	1
<code>\do@lockoff</code> .....	1
<code>\do@lockoffR</code> .....	1
<code>\do@lockon</code> .....	1

<code>\do@lockonR</code> .....	1
<code>\doinsidelineLhook</code> .....	1
<code>\doinsidelineRhook</code> .....	1
<code>\dolineLhook</code> .....	1
<code>\dolineRhook</code> .....	1
<code>\dump@pstartL@pc</code> .....	1
<code>\dump@pstartR@pc</code> .....	1

**E**

<code>\edlabel</code> .....	1
<code>\edtext</code> .....	1
<code>\eled@sectioningR@out</code> .....	1
<code>\eledsection@correcting@skip</code> .....	1
<code>\eledsectmark</code> .....	1, 20
<code>\eledsectnotoc</code> .....	1, 20
<code>\endlock</code> .....	1
<code>\endnumbering</code> .....	1, 15
<code>\endnumberingR</code> .....	1
<code>\endsub</code> .....	1
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> .....	1
<code>\finish@Pages@notes</code> .....	1
<code>\first@linenum@out@Rfalse</code> .....	1
<code>\first@linenum@out@Rtrue</code> .....	1
<code>\firstlinenum</code> .....	1, 16
<code>\firstlinenum*</code> .....	1, 16
<code>\firstlinenumR</code> .....	1, 16
<code>\firstsublinenum</code> .....	1, 16
<code>\firstsublinenum*</code> .....	1, 16
<code>\firstsublinenumR</code> .....	1, 16
<code>\fix@page</code> .....	1
<code>\flag@end</code> .....	1
<code>\flag@start</code> .....	1
<code>\flush@notesR</code> .....	1
<code>\footnote@reading</code> .....	1
<code>\footnote@typeset</code> .....	1
<code>\footnoteXmk</code> .....	14
<code>\footnoteXnomk</code> .....	14

**G**

<code>\get@familiarfootnote@number</code> .....	1
<code>\get@nextboxL</code> .....	1
<code>\get@nextboxR</code> .....	1

<code>\getline@numR</code> .....	1
<code>\getlinesfrompagelistL</code> .....	1
<code>\getlinesfrompagelistR</code> .....	1
<code>\getlinesfromparlistL</code> .....	1
<code>\getlinesfromparlistR</code> .....	1
<code>\goalfraction</code> .....	1

**H**

<code>\hidenumbering</code> .....	16
-----------------------------------	----

**I**

<code>\if@getnextbox</code> .....	1
<code>\if@pstarts</code> .....	1
<code>\ifaraw@text</code> .....	1
<code>\iffirst@linenum@out@R</code> .....	1
<code>\ifinstanzaL</code> .....	1
<code>\ifinstanzaR</code> .....	1
<code>\ifl@dpagfull</code> .....	1
<code>\ifl@dpaging</code> .....	1
<code>\ifl@dpairing</code> .....	1
<code>\ifl@dsamepage</code> .....	1
<code>\ifl@dusedbabel</code> .....	1
<code>\ifledRcol</code> .....	1
<code>\iflinenumberLevenifblank</code> .....	1
<code>\iflinenumberRevenifblank</code> .....	1
<code>\ifnomaxlines</code> .....	1
<code>\ifnosyncpstarts</code> .....	1
<code>\ifPages@mainmatter</code> .....	1
<code>\ifprevpgnotnumbered</code> .....	1
<code>\ifprint@last@after@pendL</code> .....	1
<code>\ifprint@last@after@pendR</code> .....	1
<code>\ifpst@rtedL</code> .....	1
<code>\ifpst@rtedR</code> .....	1
<code>\ifpstartnumR</code> .....	1
<code>\ifsameparallelpagenunder</code> .....	1
<code>\ifshiftedpstarts</code> .....	1
<code>\ifwidthliketwocolumns</code> .....	1
<code>\ifwrittenlinesL</code> .....	1
<code>\init@series@par</code> .....	1
<code>\initnumbering@sectcountR</code> .....	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@dbfnote</code>	1
<code>\l@dc@maxchunks</code>	1
<code>\l@dcalc@maxoftwo</code>	1
<code>\l@dcalc@minoftwo</code>	1
<code>\l@dcalcnm</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@dnumstartsL</code>	1
<code>\l@dnumstartsR</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@fail@patch@mempnum</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, 16
<code>\lineationR</code> .....	1, 16
<code>\linenum@outR</code> .....	1
<code>\linenumberLevenifblanktrue</code> .....	17
<code>\linenumberRevenifblanktrue</code> .....	17
<code>\linenumberstyle*</code> .....	1, 16
<code>\linenumberstyleR</code> .....	1, 16
<code>\linenumincrement</code> .....	1, 16
<code>\linenumincrement*</code> .....	1, 16
<code>\linenumincrementR</code> .....	1, 16
<code>\linenummargin</code> .....	1
<code>\linenummargin*</code> .....	1, 16
<code>\linenummarginR</code> .....	1, 16
<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

<code>\maxchunks</code> .....	1, 7
<code>\maxlinesinpar@list</code> .....	1
<code>\memorydump</code> .....	15
<code>\memorydumpL</code> .....	1
<code>\memorydumpR</code> .....	1

## N

<code>\n@num</code> .....	1
<code>\namebox</code> .....	1

<code>\new@lineL</code> .....	1
<code>\new@lineR</code> .....	1
<code>\newnamebox</code> .....	1
<code>\newnamecount</code> .....	1
<code>\newseries@par</code> .....	1
<code>\normalbfnoteX</code> .....	1
<code>\notesXwidthliketwocolumns</code> .....	9
<code>\num@linesR</code> .....	1
<code>\numberpstartfalse</code> .....	16
<code>\numberpstarttrue</code> .....	16
<code>\numpagelinesL</code> .....	1
<code>\numpagelinesR</code> .....	1

**O**

<code>\one@lineR</code> .....	1
<code>\onlysideX</code> .....	14
<code>optionadvancedshiftedpstarts</code> .....	10, 11
<code>optionauxdir</code> .....	146
<code>optioncontinuousnumberingwithcolumns</code> .....	9, 146
<code>optionnomaxlines</code> .....	10, 11, 22
<code>optionnosyncpstarts</code> .....	12, 22, 105
<code>optionshiftedpstarts</code> .....	6, 11, 22
<code>optionwidthliketwocolumns</code> .....	9

**P**

<code>\page@action</code> .....	1
<code>\page@numR</code> .....	1
<code>\Pages</code> .....	1, 10
<code>pages (environment)</code> .....	9
<code>\Pages@mainmatter</code> .....	1
<code>pairs (environment)</code> .....	8
<code>\par@lineR</code> .....	1
<code>\par@patch@pagenumbering</code> .....	1
<code>\par@patch@thepage</code> .....	1
<code>\parledgroup@</code> .....	1
<code>\parledgroup@beforenotes@save</code> .....	1
<code>\parledgroup@beforenotesL</code> .....	1
<code>\parledgroup@beforenotesR</code> .....	1
<code>\parledgroup@correction@notespacing</code> .....	1
<code>\parledgroup@correction@notespacing@final</code> .....	1
<code>\parledgroup@correction@notespacing@init</code> .....	1
<code>\parledgroup@notes@startL</code> .....	1
<code>\parledgroup@notes@startR</code> .....	1
<code>\parledgroup@notespacing@correction</code> .....	1
<code>\parledgroup@notespacing@set@correction</code> .....	1
<code>\parledgroupseries@</code> .....	1
<code>\parledgroupstype@</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
Rightside (environment) .....	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> .....	18
<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> .....	<u>1</u>
<code>\setRlineflag</code> .....	16
<code>\setwidthliketwocolumns@C</code> .....	<u>1</u>
<code>\setwidthliketwocolumns@L</code> .....	<u>1</u>
<code>\setwidthliketwocolumns@R</code> .....	<u>1</u>
<code>\sidenote@marginR</code> .....	<u>1</u>
<code>\sidenotemargin*</code> .....	<u>1</u>
<code>\skip@lockoff</code> .....	<u>1</u>
<code>\skipnumbering</code> .....	<u>1</u> , 16
<code>\startlock</code> .....	<u>1</u>
<code>\startsub</code> .....	<u>1</u>
<code>\sub@action</code> .....	<u>1</u>
<code>\subline@numR</code> .....	<u>1</u>
<code>\sublinenumberstyle*</code> .....	<u>1</u> , 16
<code>\sublinenumberstyleR</code> .....	<u>1</u> , 16
<code>\sublinenumincrement</code> .....	<u>1</u> , 16
<code>\sublinenumincrement*</code> .....	<u>1</u> , 16
<code>\sublinenumincrementR</code> .....	<u>1</u> , 16
<code>\sublinenumrepr</code> .....	<u>1</u>
<b>T</b>	
<code>\theledlanguageL</code> .....	<u>1</u>
<code>\theledlanguageR</code> .....	<u>1</u>
<code>\thepar@page</code> .....	<u>1</u>
<code>\thepstartL</code> .....	16
<code>\thepstartR</code> .....	16
<code>\thestanzaL</code> .....	<u>1</u> , 19
<code>\thestanzaR</code> .....	<u>1</u> , 19
<b>U</b>	
<code>\unhnamebox</code> .....	<u>1</u>
<code>\unvnamebox</code> .....	<u>1</u>
<code>\usernamecount</code> .....	<u>1</u>
<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	85
\Pages: Added \l@duselanguage to \Pages	95
General: Added section of babel related code	79
Fix babel problems	1
v0.3.0.	
\Pages: Added \ledstrutL to \Pages	95
Added \ledstrutR to \Pages	97
\Rightsidehookend: Added \Leftsidehook, \Leftsidehookend, \Rightsidehook and \Rightsidehookend	51
\affixline@numR: Changed \affixline@numR to match new eledmac	65
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR	64
\do@lineL: Added \do@lineLhook to \do@lineL	58
Simplified \do@lineL by using macros for some common code	58
\do@lineR: Changed \do@lineR similarly to \do@lineL	61
\flag@end: Removed extraneous spaces from \flag@end	44
\ifledRcol: Moved \ifl@dpairing to eledmac	24
\ifpst@rtedR: Moved \ifpst@rtedL to eledmac	27
\l@dlinenumR: Simplified \leftlinenumR and \rightlinenumR by introducing \l@dlinenumR	35
\l@dnumpstartsR: Moved \l@dnumpstartsL to eledmac	81
\ledstrutR: Added \ledtrutL and \ledstrutR	99
\normalbfnoteX: Removed extraneous spaces from \normalbfnoteX	76
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR	34
General: Added \do@lineLhook and \do@lineRhook	61
Added hooks into Leftside environment	51
Reorganize for ledarab	1
v0.3.a.	
\line@marginR: Do not just set \line@marginR in \linenummargin	32
General: Minor \linenummargin fix	1
v0.3.b.	
\Pages: Added \l@dminpagelines calculation for succeeding page pairs	98
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 unequal length.	1

v0.8.0.	General: Possibility to have a symbol on each hanging of verses, like in the french typography. Redefine the commande <code>\hangingsymbol</code> to define the character. . . . .	1
v0.9.0.	<code>\ifledRcol</code> : Moved <code>\iflledRcol</code> and <code>\ifnumberingR</code> to <code>eledmac</code> . . . . .	24
	General: Possibility to number <code>\pstart</code> . . . . .	16
	Possibility to number the <code>pstart</code> with the commands <code>\numberpstarttrue</code> . . . . .	1
v0.9.1.	General: The numbering of the <code>pstarts</code> restarts on each <code>\beginnumbering</code> . . . . .	1
v0.9.2.	General: Debug : with <code>\Columns</code> , the hanging indentation now runs on the left columns and the hanging symbol is shown only when <code>\stanza</code> is used. . . . .	1
v0.9.3.	General: <code>\thepstartL</code> and <code>\thepstartR</code> use now <code>\bfseries</code> and not <code>\bf</code> , which is deprecated and makes conflicts with <code>memoir</code> class. . . . .	1
v0.10.0.	General: <code>\edlabel</code> commands on the right side are now correctly indicated. . . . .	1
	<code>\edlabel</code> commands which start a paragraph are now put in the right place. . . . .	1
v0.11.0.	<code>\Columns</code> : Line numbering by <code>pstart</code> . . . . .	86
	<code>\affixline@numR</code> : Changed <code>\affixline@numR</code> to allow to disable line numbering (like in <code>eledmac 0.15</code> ). . . . .	65
	<code>\get@nextboxR</code> : Change <code>\get@nextboxL</code> and <code>\get@nextboxR</code> to allow to disable line numbering (like in <code>eledmac 0.15</code> ). . . . .	105
	<code>Pstart</code> number can be printed in side . . . . .	107
	<code>\inserthangingsymbolR</code> : Prevent the column separator for hanging verse from shifting	77
	General: Change <code>\do@lineL</code> and <code>\do@lineR</code> to allow line numbering by <code>pstart</code> (like in <code>eledmac 0.15</code> ). . . . .	58
	Lineation can be by <code>pstart</code> (like in <code>eledmac 0.15</code> ). . . . .	31
	New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . .	77
v0.12.0.	General: New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . . . .	77
v1.0.0.	General: Compatibility with <code>eledmac</code> . Change name to <code>eledpar</code> . . . . .	1
	Debug in lineation by <code>pstart</code> . . . . .	31
v1.0.1.	General: Correction on <code>\numberonlyfirstinline</code> with lineation by <code>pstart</code> or by page. .	1
v1.1.0.	<code>\pstartR</code> : Add <code>\labelpstarttrue</code> (from <code>eledmac</code> ). . . . .	53
	General: <code>Shiftedverses</code> becomes <code>shiftedpstarts</code> . . . . .	1
v1.1.1.	<code>\pstartR</code> : Correct <code>\pstartR</code> bug introduced by 1.1. . . . .	53
v1.1.2.	<code>\affixside@noteR</code> : Remove spurious space between line number and line content . .	75
v1.2.0.	General: Support for <code>\led&lt;section&gt;</code> commands in parallel texts. . . . .	1
v1.2.1.	<code>\set@sectcountR</code> : For the right section, the counter is defined only once. . . . .	29

v1.3.0.		
\edtext:	Manage RTL language. . . . .	45
v1.3.1.		
\l@dbfnote:	Compatibility of standard footnotes with eledmac when these footnotes contain any commands. . . . .	76
v1.3.2.		
General:	Debug with some classes. . . . .	1
v1.3.3.		
\l@dbfnote:	Spurious space with footnote in right column. . . . .	76
General:	Debugging the left notes of the right column. . . . .	75
v1.3.4.		
General:	Allow use of commands in sidenotes, as introduced by eledmac 1.0. . . . .	75
v1.3.5.		
\normalbfnoteX:	Allows one to redefine \thefootnoteX with alph when some packages are loaded. . . . .	76
v1.4.0.		
General:	Added \do@insidelineLhook and \do@insidelineRhook . . . . .	61
v1.4.1.		
\normalbfnoteX:	Fix bug with normal familiar footnotes when mixing RTL and LTR text. . . . .	76
General:	Enable the use of stanzaindentsrepetition within astanza environment. . . . .	77
v1.4.3.		
\inserthangingsymbolR:	Hanging verse is no longer automatically flush right. . . . .	77
\pendL:	Spurious spaces in \pendL. . . . .	56
\pendR:	Spurious spaces in \pendR. . . . .	56
\pstartR:	Spurious spaces in \pstartL and \pstartR. . . . .	53
General:	Corrects a false hanging verse when a verse is exactly the length of a line. . . . .	1
v1.5.0.		
\sublinenumincrement*:	Add starred version of \firstlinenum, \linenumincrement, \firstsublinenum, \sublinenumincrement to change both Left and Rightside. . . . .	33
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 pstart numbers. . . . .	1
v1.8.0.		
\Columns:	Modify \Columns to enable to add section's title. . . . .	84
Suppress \l@dchecklang	from \Columns. . . . .	85
\Pages:	Modify \Pages to enable to add section's title. . . . .	92
\l@dchecklang:	Suppress \l@dchecklang which did not work and was not logical, because both columns could have the same language but not the main language of the document. . . . .	79
\pendL:	As in eledmac, \pendL can have an optional argument. . . . .	56
\pendR:	As in eledmac, \pendR can have an optional argument. . . . .	56
\print@columnseparator:	Move some code of \Columns to \print@columnseparator. . . . .	87
\pstartR:	As in eledmac, \pendL and \pendR can have an optional argument. . . . .	53
\sidenotemargin*:	\sidenotemargin is now directly defined in eledmac to be able to manage eledpar. . . . .	75

Add <code>\sidenotemargin*</code> .....	75
<code>\theledlanguageR</code> : Correct left/right language setting with polyglossia. ....	81
General: <code>\beginnumbering</code> is defined only on <code>eledmac</code> , not on <code>eledpar</code> . ....	27
<code>\l@dlsnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in <code>eledmac</code> . ....	75
Add <code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code> . ....	9
Add <code>\columnspan</code> . ....	9
Add, as in <code>eledmac</code> , new system of sectioning commands. ....	1
Add, as in <code>eledmac</code> , option to insert something after <code>\pends</code> / verses. ....	1
Add, as in <code>eledmac</code> , 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. ....	58
Compatibility with <code>musixtex</code> . ....	1
Debug <code>eledmac</code> sectioning command after using <code>\resumenumbering</code> . ....	1
New sectioning commands, as in <code>eledmac</code> . ....	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. ....	79
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) (?). ....	58
<code>\do@lineR</code> : Fix a bug with critical notes at the beginning of a page, added by v1.8.0 (?). ....	61
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> .....	44
<code>\lineation*</code> : Add <code>\lineation*</code> .....	32
<code>\reledpar@error</code> : Errors specific to <code>eledpar</code> send to <code>eledpar</code> handbook .....	24
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> . ....	92
<code>\doinsidelineRhook</code> : Added <code>\dolineLhook</code> , <code>\dolineRhook</code> , <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code> .....	60
<code>\resumenumberingR</code> : Debug <code>\resumenumberingR</code> .....	30
General: Add <code>\noeledxxx</code> , as in <code>eledmac</code> .....	1
v1.9.0.	
<code>\ifwidthliketwocolumns</code> : Added <code>widthliketwocolumns</code> option .....	23
<code>\theledlanguageR</code> : Debug left/right language switching with polyglossia. Do not write in .aux file when setting left/right lines. ....	81
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> .....	24
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). ....	92
Debug wrong parallel pages synchronization when an <code>\edtext</code> falls across two pages. ....	92
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.	
\line@list@stuffR: Revert modification of 1.4.2, which makes bugs with numbering. Leave vertical mode to solve spurious space before minipage. . . . .	43
v1.11.0.	
\edtext: \critext and \edtext are now defined only in eledmac. . . . .	45
General: Compatibility of standard footnotes with some biblatex styles. . . . .	1
v1.12.0.	
\Columns: Add \l@dprintingcolumnstrue . . . . .	84
\Pages: Add \l@dprintingpagestrue . . . . .	92
\edlabel: \edlabel and \edindex works now with hyperref when using eledpar. . . . .	74
\edlabel is now defined only one time for both eledmac and eledpar . . . . .	74
\print@eledsectionL: Compatibility with Lua $\TeX$ RTL languages. . . . .	60
\print@eledsectionR: Compatibility with Lua $\TeX$ RTL languages. . . . .	62
\print@lineL: Compatibility with Lua $\TeX$ RTL languages. . . . .	59
General: Compatibility with Lua $\TeX$ RTL languages. . . . .	1
v1.12.1.	
\print@eledsectionL: Fixes bug with Lua $\TeX$ RTL \eledsection. . . . .	60
v1.13.0.	
\Pages: Prevent false overfull hboxes when using \Pages outside of pages environment. When using shiftedpstarts option, a \l@dleftbox with a null height will advance the \pagetotal in any case. . . . .	93 92
\clearl@drightpage: Use \newpage instead of \clearpage. . . . .	100
\ifledRcol: Remove false boolean settings which are not needed. . . . .	24
General: Enable the use of optional argument of & in a stanza environment. . . . .	77
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) . . . . .	92
\correct@footinsX@box: Call \correct@footinsX@box and \correct@Xfootins@box directly in \print@notesX@forpages and \print@Xnotes@forpages. . . . .	71
Correct \correct@footinsX@box and \correct@Xfootins@box . . . . .	71
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 . . . . .	64
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting when a verse is hanging . . . . .	77
\lineationR: As \lineation, \lineationR automatically set the \pstartinfootnote. . . . .	31
\n@num: \n@num defined only one time for both Eledmac and Eledpar. . . . .	40
\skipnumbering: \skipnumbering defined only one time for both Eledmac and Eledpar . . . . .	45
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 <code>\AtEveryPstart</code> or <code>\AtEveryPend</code> with a command which prints nothing	1
v1.16.0.	
<code>\newseries@par</code> : Fix bug with <code>\onlysideX</code> .	45
General: Error message when calling <code>\Pages</code> inside ‘pages’ environment and <code>\Columns</code> 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 <code>\beginnumbering.. \endnumbering</code> without <code>\pstart</code> .	1
Fix bug with <code>nofamiliar / nocritical</code> option of <code>eledmac</code> .	1
New package option <code>sameparallelpagenumber</code> 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> .	38
<code>\@lab</code> : <code>\@lab</code> defined only in <code>eledmac</code> .	75
<code>\@ref@regR</code> : <code>\@ref</code> defined only in <code>reledmac</code> , code specific to right side moved in <code>\ref@regR</code> .	40
<code>\@set</code> : <code>\@set</code> defined only in <code>reledmac</code> .	39
<code>\advanceline</code> : <code>\advanceline</code> defined only in <code>reledmac</code> .	44
<code>\bbl@set@language</code> : Patch <code>\bbl@set@language</code> instead of redefining it	79
<code>\do@lockonR</code> : <code>\do@lockon</code> defined only in <code>reledmac</code> .	39
<code>\endlock</code> : <code>\startlock</code> and <code>\endlock</code> defined only in <code>reledmac</code> .	44
<code>\endsub</code> : <code>\startsub</code> and <code>\endsub</code> defined only in <code>reledmac</code> .	44
<code>\fix@page</code> : <code>\fix@page</code> is defined only once in <code>reledmac</code>	38
<code>\l@d@set</code> : <code>\l@d@set</code> defined only in <code>reledmac</code> .	39
<code>\l@dbfnote</code> : <code>\l@dbfnote</code> defined only in <code>reledmac</code> .	76
<code>\line@marginR</code> : <code>\linenummargin</code> now defined only once time in <code>reledmac</code> .	32
<code>\normalbfnoteX</code> : <code>\normalbfnoteX</code> defined only in <code>reledmac</code> .	76
<code>\page@action</code> : <code>\page@action</code> defined only in <code>reledmac</code> .	39
<code>\read@linelist</code> : <code>\read@linelist</code> is defined only once time in <code>\reledmac</code> .	37
<code>\set@line</code> : <code>\set@line</code> defined only in <code>reledmac</code> .	45
<code>\set@line@action</code> : <code>\set@line@action</code> defined only in <code>reledmac</code> .	39
<code>\setline</code> : <code>\setline</code> defined only in <code>reledmac</code> .	44
<code>\setlinenum</code> : <code>\setlinenum</code> defined only in <code>reledmac</code> .	44
<code>\skip@lockoff</code> : <code>\do@lockoff</code> defined only in <code>reledmac</code> .	39
<code>\sub@action</code> : <code>\sub@action</code> defined only in <code>reledmac</code> .	39
<code>\sublinenumincrement*</code> : <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> are now defined only in <code>reledmac</code> .	33
<code>\theledlanguageR</code> : Patch <code>\otherlanguage</code> instead of redefining it.	81
General: <code>\@nl</code> is now defined only in <code>reledmac</code> .	37

<code>\ifbypage@</code> and <code>\ifbypstart@R</code> defined in <code>eledmac</code> . . . . .	31
Fix some bugs with ‘ <code>sameparallelpagenu</code> ’ option. . . . .	1
Many code refactored and moved to <code>reledmac</code> . . . . .	1
Package’s name becomes <code>reledpar</code> . . . . .	1
Totally new implementation of ‘ <code>sameparallelpagenu</code> ’ option. . . . .	1
<code>chapterinpages</code> : Deleting the old system of managing parallel chapter, keep only the new one with <code>\patchcmd</code> . . . . .	50
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 memoir (bug added in v2.0.0). . . . .	1
Fix bug with <code>\setparledgroupnotespacing</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). . . . .	93
General: Change some internal codes in order to provide compatibility with L <sup>A</sup> T <sub>E</sub> X 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> . . . . .	99
Fix bug with dotted letter . . . . .	99
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> . . . . .	74
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 babel (broken in v.2.0.0). . . . .	1
No error messages about ends of left / right page when using the \syntaxonly command of the syntonly package. . . . .	1
v2.6.1.	
General: Fix bug, introduced in v2.6.0, with footnote numbering when using perpage package. . . . .	1
v2.6.2.	
\newseries@par: The T <sub>E</sub> X counter \footnoteX@reading is defined in reledmac. . . . .	46
General: Fix (again) bugs with footnote numbering in parallel typesetting while using ledgroup 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 . . . . .	64
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. . . . .	71
New implementation of \Xonlyside and \onlysideX hooks, prevent trouble with vertical spacing. . . . .	71
General: Compatibility with reledmac's auxdir option. . . . .	1