

# 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](https://github.com/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 v2.10.0a, last revised 2016/06/03.

<sup>†</sup>`maieul` at `maieul` dot `net`

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

<b>4 Parallel columns</b>	<b>8</b>
4.1 Basic use . . . . .	8
4.2 Setting . . . . .	8
4.2.1 Column's width . . . . .	8
4.2.2 Column's separator . . . . .	8
4.2.3 Column's positions . . . . .	9
4.2.4 Mixing two columns and one column texts . . . . .	9
<b>5 Facing pages</b>	<b>9</b>
5.1 Basic usage . . . . .	9
5.2 Setting . . . . .	10
5.2.1 Text width . . . . .	10
5.2.2 Way of synchronizing . . . . .	10
5.2.3 Page number . . . . .	12
5.2.4 Page breaking . . . . .	12
5.2.5 Right page before \Pages . . . . .	12
5.2.6 Notes about \mainmatter . . . . .	12
5.3 Critical and familiar footnotes . . . . .	13
5.3.1 Notes height setting . . . . .	13
5.3.2 About the numbering of familiar footnotes . . . . .	13
5.3.3 Using perpage package . . . . .	13
5.3.4 Notes for one side only . . . . .	14
5.3.5 Familiar notes called in the right side, but to be printed in the left side . . . . .	14
5.4 Using line flag . . . . .	14
<b>6 Left and right texts</b>	<b>15</b>
6.1 Environments . . . . .	15
6.2 Numbering text lines and paragraphs . . . . .	15
6.3 Lineation system . . . . .	16
6.4 Chunks . . . . .	17
6.5 \AtEveryPstart and \AtEveryPstartCall . . . . .	17
6.6 Language setting . . . . .	17
<b>7 Verse</b>	<b>18</b>
<b>8 Side notes</b>	<b>19</b>
<b>9 Parallel ledgroups</b>	<b>19</b>
9.1 General . . . . .	19
9.2 Parallel ledgroups and setspace package . . . . .	20
<b>10 Sectioning commands</b>	<b>20</b>
<b>11 Notes about page number</b>	<b>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
<b>III Sectioning commands</b>	<b>26</b>
<b>IV Line counting</b>	<b>30</b>
IV.1 Setting lineation reset . . . . .	30
IV.2 Setting line number margin . . . . .	31
IV.3 Setting lineation start and step . . . . .	32
IV.4 Setting line flag . . . . .	33
IV.5 Setting line number style . . . . .	33
IV.6 Print marginal line number . . . . .	34
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> . . . . .	35
IV.8 Reading the line-list file . . . . .	36
IV.9 Commands within the line-list file . . . . .	36
IV.10 Writing to the line-list file . . . . .	42
<b>V Marking text for notes</b>	<b>44</b>
V.1 Specific hooks and commands for notes . . . . .	44
V.1.1 Notes to be printed on one side only . . . . .	44
V.2 Tools specific to familiar footnotes . . . . .	45
V.2.1 Managing correct number . . . . .	45
V.2.2 Familiar footnotes without marks . . . . .	45
V.2.3 Get correct footnote number . . . . .	47
V.3 Create hooks . . . . .	47
V.4 Init standards series (A,B,C,D,E,Z) . . . . .	48
V.5 Tools specific to $\LaTeX$ 's classical footnotes . . . . .	48
<b>VI Pstart numbers dumping and restoration</b>	<b>48</b>
<b>VII Parallel environments</b>	<b>49</b>
<b>VIII Paragraph decomposition and reassembly</b>	<b>52</b>
VIII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code> . . . . .	52
VIII.2 Processing one line . . . . .	57
VIII.3 Line and page number computation . . . . .	62
VIII.4 Line number printing . . . . .	64

VIII.5 Pstart number printing in side . . . . .	67
VIII.6 Add insertions to the vertical list . . . . .	68
VIII.7 Penalties . . . . .	69
VIII.8 Printing leftover notes . . . . .	70
<b>IX Footnotes</b>	<b>70</b>
IX.1 Footnotes output specific to \Pages . . . . .	70
<b>X Cross referencing</b>	<b>75</b>
<b>XI Side notes</b>	<b>75</b>
<b>XII Familiar footnotes</b>	<b>77</b>
<b>XIII Verse</b>	<b>77</b>
<b>XIV Naming macros</b>	<b>79</b>
<b>XV Fixing babel and polyglossia</b>	<b>80</b>
<b>XVI Counts and boxes for parallel texts</b>	<b>82</b>
<b>XVII Checking text to be processed</b>	<b>83</b>
<b>XVIII Parallel columns</b>	<b>85</b>
<b>XIX Parallel pages</b>	<b>93</b>
XIX.1 Specific counters . . . . .	93
XIX.2 Main macro . . . . .	93
XIX.3 Ensure all notes be printed at the end of parallel pages . . . . .	100
XIX.4 Struts . . . . .	100
XIX.5 Page clearing . . . . .	101
XIX.6 Lines managing . . . . .	102
XIX.7 Page break managing . . . . .	103
XIX.8 Getting boxes content . . . . .	106
<b>XX Page numbering</b>	<b>110</b>
XX.1 Global options . . . . .	110
XX.2 mainmatter option of \Pages . . . . .	112
<b>XXI Sections' titles' commands</b>	<b>113</b>
<b>XXII Page break/no page break, depending on the specific line</b>	<b>113</b>
<b>XXIII Parallel ledgroup</b>	<b>114</b>
<b>XXIV Compatibility with eledmac</b>	<b>118</b>

<b>XXV The End</b>	<b>118</b>
<b>Appendix A Some things to do when changing version</b>	<b>119</b>
Appendix A.1 Migration to eledpar 1.4.3 . . . . .	119
Appendix A.2 Migration from eledpar to reledpar . . . . .	119
Appendix A.2.1 Deprecated options . . . . .	119
Appendix A.2.2 \renewcommandreplaced with command . . . . .	119
Appendix A.2.3 Commands the names of which have changed . . . . .	120
Appendix A.3 Migration to reledpar 2.2.0 . . . . .	120
Appendix A.4 Migration to reledpar 2.3.0 . . . . .	120
Appendix A.5 Migration to reledpar 2.4.0 . . . . .	120
Appendix A.6 Migration to reledpar 2.5.0 . . . . .	120
Appendix A.7 Migration to reledpar 2.6.0 . . . . .	120
Appendix A.8 Migration to reledpar 2.6.1 . . . . .	120
<b>References</b>	<b>121</b>
<b>Index</b>	<b>121</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 XXV); and an Index to the source code. As `reledpar` is an adjunct to `reledmac` we assume that you have read the `reledmac` manual. Also `reledpar` requires `reledmac` to be used, in the version distributed with version.

You do not need to read the source code for this package in order to use it but doing so may help to answer any questions you might have. The documentation's sections are numbered in roman numeral.

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

## 1.2 Historical overview

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

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

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

## 2 Options

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

### 2.1 Synchronization's options

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

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

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

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

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

### 2.2 Other options

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

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

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

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

---

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

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

### 3 General

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

The `reledpar` package lets you typeset two *numbered* texts in parallel<sup>3</sup>. This can be done either as setting the 'Leftside' and 'Rightside' texts in two columns or on facing pages. In the paired pages case footnotes are placed at the bottom of the page on which they are called out — that is, footnotes belonging to the left are set at the foot of a left (even numbered) page, and those for right texts are at the bottom of the relevant right (odd numbered) page. However, in the columnar case, all footnotes are set at the bottom left of the page on which they are called out — they are not set below the relevant column.

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

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

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

```
\maxchunks{5120}
```

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

If you `\maxchunks` is too little you can get a `reledpar` error message along the lines: "Too many `\pstart` without printing. Some text will be lost." then you will have to either increase `\maxchunks` or use the parallel printing commands (`\Columns` or `\Pages`) more frequently.

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

In general, `reledmac` is a  $\TeX$  resource hog, and `reledpar` only makes things worse

---

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

in this respect.

## 4 Parallel columns

### 4.1 Basic use

`pairs` Numbered text that is to be set in columns must be within a `pairs` environment. Within the environment the text for the lefthand and righthand columns is placed within the `Leftside` and `Rightside` environments, respectively; these are described in more detail below in section 6.

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

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

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

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

There is no required pagebreak before or after the columns.

### 4.2 Setting

#### 4.2.1 Column's width

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

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

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

#### 4.2.2 Column's separator

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



Opt the rule is invisible. For a visible rule between the columns you could try:

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

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

### 4.2.3 Column's positions

`\columnspan`

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

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

`\beforecolumnseparator`  
`\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 `sameparallelpagelapnumber` 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. 42). 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.

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

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

`\footnoteXmk`

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

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

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

## 5.4 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 T<sub>E</sub>X’s memory of previous texts and any associated notes, allowing longer apparent streams of parallel texts. The command should be applied to both left and right texts, and after making sure that all previous notes have been output. For example, along the lines of:

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

```

\begin{Rightside}
  \memorydump
  ...
\end{pages}

```

```

\numberpstarttrue
\numberpstartfalse
  \thepstartL
  \thepstartR
\skipnumbering
\hidenumbering

```

It is possible to insert a number at every `\pstart` command. You must use the `\numberpstarttrue` command to have it. You can stop the numbering with `\numberpstartfalse`. You can redefine the 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
\linenumberLevenifblanktrue
\linenumberRevenifblanktrue

```

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`  
`\pend` 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 `\szi@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. 47) 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. 48). To do it you have:

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

See the following example:

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

```
\end{pages}
\Pages
```

## 9.2 Parallel ledgroups and setspace package

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

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

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

## 10 Sectioning commands

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

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

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

## 11 Notes about page number

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

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

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

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

## I Implementation overview

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

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

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

## II Preliminaries

### II.1 Package’s meta-data

Announce the name and version of the package, which is targeted for  $\text{\LaTeX 2e}$ . 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}[2016/06/03 v2.10.0a 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.

```
\ifshiftedpstarts 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   \nomaxlinestrue%
35   \nosyncpstartstrue%
36   \apptocmd{\@par@this@sync@option}{nosync}{-}{-}%
37 }%
38 %

```

### II.4.2 Other options

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

```

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

```

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

```

41 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
42 \DeclareOptionX{continuousnumberingwithcolumns}{\
43   continuousnumberingwithcolumnstrue}%
44 %

```

`\ifsameparallelpagenunder` Options related to page numbering

`\ifprevpgnotnumbered`

```

44 \newif\ifsameparallelpagenunder
45 \newif\ifprevpgnotnumbered
46 \DeclareOptionX{sameparallelpagenunder}{\sameparallelpagenundertrue}
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@PreviousNotPrinted 82 \newcommand*{\led@err@Leftside@PreviousNotPrinted}{%
htside@PreviousNotPrinted 83 \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@InsideEnv 87 \newcommand*{\led@err@Pages@InsideEnv}{%
led@err@Columns@InsideEnv 88 \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@WithoutEnv 92 \newcommand*{\led@err@Pages@WithoutEnv}{%
led@err@Columns@WithoutEnv 93 \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@thepage 97 \newcommand{\led@error@fail@patch@thepage}{%
98 \reledpar@error{Fail to patch \string\@thepage\space command.}{\@ehc}%
99 }%
100 %

```

```

@fail@patch@pagenumbering 01 \newcommand{\led@error@fail@patch@pagenumbering}{%
102 \reledpar@error{Fail to patch \string\pagenumbering\space command.}{\@ehc
}%
103 }%
104 %

```

```

error@fail@patch@@mempnum 05 \newcommand{\led@error@fail@patch@@mempnum}{%
106 \reledpar@error{Fail to patch \string@mempnum\space command.}{\@ehc}%
107 }%
108 %

```

```

or@fail@patch@@outputpage 09 \newcommand{\led@error@fail@patch@@outputpage}{%
110 \reledpar@error{Fail to patch \string\@outputpage\space command.}{\@ehc}%
111 }%
112 %

```

```

\led@warn@ChangeSyncOption13 \newcommand*{\led@warn@ChangeSyncOption}[1]{%
114 \reledpar@warning{You have changed synchronization's options since last
run. We have not read line-list file #1. Please run LaTeX again.}%
115 }%
116 %

```

```

\led@warn@setting@in@rightside17 \newcommand{\led@warn@setting@in@rightside}[1]{%
118 \reledpar@warning{You use #1 inside rightside environment.\MessageBreak%
119 Such behavior is deprecated.\MessageBreak%
120 Use instead #1R or #1* in your preamble.}%
121 }
122 %

```

```

\led@error@missing@numbering23 \newcommand{\led@error@missing@numbering}[1]{%
124 \reledpar@error{Missing \string\...pstart\string\pend\space inside `#1`
environment}{\@ehc}%
125 }%
126 %

```

### 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\section@num.nn`, where `nn` is the section number. However, for right side texts the file is called `\jobname\section@num.nnR`. The `\extensionchars` applies to the right side files just as it does to the normal files.

```

127 \newcount\section@numR
128 \section@numR=\z@
129 %

```

`\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`.

```

130 \pst@rtedLfalse
131 \newif\ifpst@rtedR
132
133 %

```

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

```

134 \newcommand*{\beginnumberingR}{%
135 \ifnumberingR
136 \led@err@NumberingStarted

```

```

137 \endnumberingR
138 \fi
139 \global\l@dnumstartsR \z@
140 \global\pst@rtedRfalse
141 \global\numberingRtrue
142 \global\advance\section@numR \@ne
143 \global\absline@numR \z@
144 \gdef\normal@page@breakR{}
145 \gdef\l@prev@pbR{}
146 \gdef\l@prev@nopbR{}
147 \global\line@numR \z@
148 \global\@lockR \z@
149 \global\sub@lockR \z@
150 \global\sublines@false
151 \global\let\next@page@numR\relax
152 \global\let\sub@change\relax
153 \global\stopmsdata@inserted@true%
154 \set@continuousnumberingforR%
155 \message{Section \the\section@numR R }%
156 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
157 \l@dend@stuff
158 \setcounter{pstartR}{1}
159 \begingroup
160 \initnumbering@sectcountR
161 \gdef\eled@sectionsR@{ }%
162 \if@noeled@sec\else%
163 \makeatletter\inputIfFileExists{\jobname.eledsec\the\section@numR R
164 }{ }{\makeatother}%
165 \immediate\openout\eled@sectioningR@out=\jobname.eledsec\the\
166 section@numR R\relax%
167 \fi%
168 }
169 %

```

**\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.

```

168 \def\endnumberingR{%
169 \ifnumberingR
170 \global\numberingRfalse
171 \normal@pars
172 \ifnum\l@dnumstartsR=0%
173 \led@err@NumberingWithoutPstart%
174 \fi%
175 \ifl@dpairing
176 \global\pst@rtedRfalse

```

```

177 \else
178   \ifx\insertlines@listR\empty\else
179     \global\noteschanged@true
180   \fi
181   \ifx\line@listR\empty\else
182     \global\noteschanged@true
183   \fi
184 \fi
185 \ifnoteschanged@
186   \led@mess@NotesChanged
187 \fi
188 \else
189   \led@err@NumberingNotStarted
190 \fi
191 \endgroup
192 \if@noeled@sec\else%
193   \immediate\closeout\eled@sectioningR@out%
194 \fi%
195 }
196
197 %

```

`\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`

```

198 \newcounter{chapterR}
199 \newcounter{sectionR}
200 \newcounter{subsectionR}
201 \newcounter{subsubsectionR}
202
203 \newcount\old@chapter%
204 \newcount\old@section%
205 \newcount\old@subsection%
206 \newcount\old@subsubsection%
207 \newcommand{\save@section@number}{%
208   \ifdefined\c@chapter%
209     \global\old@chapter\value{chapter}%
210   \fi%
211   \global\old@section\value{section}%
212   \global\old@subsection\value{subsection}%
213   \global\old@subsubsection\value{subsubsection}%
214 }%
215 \newcommand{\initnumbering@sectcountR}{
216   \ifdefined\c@chapter%
217     \setcounter{chapterR}{\old@chapter}%
218   \fi%
219   \setcounter{sectionR}{\old@section}%
220   \setcounter{subsectionR}{\old@subsection}%
221   \setcounter{subsubsectionR}{\old@subsubsection}%

```

```

222 \set@sectcountR%
223 }
224 \newcommand{\set@sectcountR}{%
225     \let\c@chapter\c@chapterR%
226     \let\c@section\c@sectionR%
227     \let\c@subsection\c@subsectionR%
228     \let\c@subsubsection\c@subsubsectionR%
229 }%
230 %

```

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

```

231 \newcommand*{\pausenumberingR}{%
232     \endnumberingR\global\numberingRtrue}
233 \newcommand*{\resumenumberingR}{%
234     \ifnumberingR
235         \global\pst@rtedRtrue
236         \global\advance\section@numR \@ne
237         \led@mess@SectionContinued{\the\section@numR R}%
238         \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
239         \l@dend@stuff
240         \begingroup%
241         \initnumbering@sectcountR%
242         \set@continuousnumberingforR%
243     \else
244         \led@err@numberingShouldHaveStarted
245         \endnumberingR
246         \beginnumberingR
247     \fi}
248
249 %

```

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

```

250 \newcommand*{\memorydumpL}{%
251     \endnumbering
252     \numberingtrue
253     \global\pst@rtedLtrue
254     \global\advance\section@num \@ne
255     \led@mess@SectionContinued{\the\section@num}%
256     \line@list@stuff{\jobname.\extensionchars\the\section@num}%
257     \l@dend@stuff}
258
259 \newcommand*{\memorydumpR}{%
260     \endnumberingR
261     \numberingRtrue
262     \global\pst@rtedRtrue
263     \global\advance\section@numR \@ne
264     \led@mess@SectionContinued{\the\section@numR R}%

```

```

265 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
266 \l@dend@stuff}
267
268 %

```

## 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`.

```

269 \newcommand*{\lineationR}[1]{%
270   \ifnumbering
271     \led@err@LineationInNumbered
272   \else
273     \def\@tempa{#1}\def\@tempb{page}%
274     \ifx\@tempa\@tempb
275       \global\bypage@Rtrue
276       \global\bypstart@Rfalse
277       \unless\ifnocritical@%
278         \Xpstart[] [false]%
279       \fi%
280     \else
281       \def\@tempb{pstart}%
282       \ifx\@tempa\@tempb
283         \global\bypage@Rfalse
284         \global\bypstart@Rtrue
285         \unless\ifnocritical@%
286           \Xpstart%
287         \fi%
288       \else
289         \def\@tempb{section}
290         \ifx\@tempa\@tempb
291           \global\bypage@Rfalse%
292           \global\bypstart@Rfalse%
293           \unless\ifnocritical@%
294             \Xpstart[] [false]%
295           \fi%
296         \else
297           \led@warn@BadLineation
298         \fi%
299       \fi
300     \fi

```

```

301 \fi}}
302 %

```

**\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.

```

303 \newcommand{\set@continuousnumberingforR}{%
304   \ifcontinuousnumberingwithcolumns%
305     \ifnum\line@numR<\line@num%
306       \expandafter\setlinenum\expandafter{\the\line@num}%
307     \fi%
308     \ifnum\last@page@num>\last@page@numR%
309       \global\last@page@numR=\last@page@num%
310     \fi%
311   \fi%
312 }
313 %

```

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

```

314 \WithSuffix\newcommand\lineation*[1]{%
315   \lineation{#1}%
316   \lineationR{#1}%
317 }%
318 %

```

## IV.2 Setting line number margin

**\linenummargin** You call \linenummargin{<word>} to specify which margin you want your right text's line numbers in; it takes one argument, a string. You can put the line numbers in the same margin on every page using left or right; or you can use inner or outer to get them in the inner or outer margins. You can change this within a numbered section, but the change may not take effect just when you would like; if it is done between paragraphs nothing surprising should happen.

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

It is defined only once time, in reledmac.

```

319 \newcount\line@marginR
320 %

```

By default put right text numbers at the right.

```

321 \line@marginR=\@ne
322
323 %

```

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

```

324 \newcommand{\linenummarginR}[1]{%
325   \l@getline@margin{#1}%
326   \ifnum\@l@tempcntb>\m@ne%
327     \global\line@marginR=\@l@tempcntb%
328   \fi%
329 }
330 \WithSuffix\newcommand\linenummargin*[1]{%
331   \l@getline@margin{#1}%
332   \ifnum\@l@tempcntb>\m@ne%
333     \global\line@marginR=\@l@tempcntb%
334     \global\line@margin=\@l@tempcntb%
335   \fi%
336 }
337 %

```

### IV.3 Setting lineation start and step

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

```

338 \newcounter{firstlinenumR}
339   \setcounter{firstlinenumR}{5}
340 \newcounter{linenumincrementR}
341   \setcounter{linenumincrementR}{5}
342 %

```

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

```

343 \newcounter{firstsublinenumR}
344   \setcounter{firstsublinenumR}{5}
345 \newcounter{sublinenumincrementR}
346   \setcounter{sublinenumincrementR}{5}
347
348 %

```

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

```

349 \firstsublinenum
350 \sublinenumincrement
351 \firstlinenum*
352 \linenumincrement*
353 \firstsublinenum*
354 \sublinenumincrement*
355
\WithSuffix\newcommand\firstlinenum*[1]{%
  \setcounter{firstlinenumR}{#1}%
  \setcounter{firstlinenum}{#1}%
}
\WithSuffix\newcommand\linenumincrement*[1]{%
  \setcounter{linenumincrementR}{#1}%
  \setcounter{linenumincrement}{#1}%
}

```



```

356 }
357 \WithSuffix\newcommand\firstsublinenum*[1]{%
358   \setcounter{subfirstlinenumR}{#1}%
359   \setcounter{subfirstlinenum}{#1}%
360 }
361 \WithSuffix\newcommand\sublinenumincrement*[1]{%
362   \setcounter{sublinenumincrementR}{#1}%
363   \setcounter{sublinenumincrement}{#1}%
364 }
365 %

```

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

```

\linenumincrementR
\firstsublinenumR
\sublinenumincrementR
366 \newcommand\firstlinenumR[1]{%
367   \setcounter{firstlinenumR}{#1}%
368 }
369 \newcommand\linenumincrementR[1]{%
370   \setcounter{linenumincrementR}{#1}%
371 }
372 \newcommand\firstsublinenumR[1]{%
373   \setcounter{subfirstlinenumR}{#1}%
374 }
375 \newcommand\sublinenumincrementR[1]{%
376   \setcounter{sublinenumincrementR}{#1}%
377 }
378 %

```

## IV.4 Setting line flag

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

```

379 \newcommand{\setRlineflag}[1]{%
380   \gdef\Rlineflag{#1}%
381 }
382 \setRlineflag{R}
383 %

```

## IV.5 Setting line number style

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

```

384 \newcommand*\linenumrepR[1]{\@arabic{#1}}
385 \newcommand*\sublinenumrepR[1]{\@arabic{#1}}
386
387 %

```

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

`\sublinenumbersstyleR`

```

388 \newcommand*\linenumberstyleR}[1]{%
389   \def\linenumrepR##1{\@nameuse{@#1}{##1}}
390 \newcommand*\sublinenumberstyleR}[1]{%
391   \def\sublinenumrepR##1{\@nameuse{@#1}{##1}}
392 %

```

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

```

393 \WithSuffix\newcommand\linenumberstyle*[1]{%
394   \linenumberstyle{#1}%
395   \linenumberstyleR{#1}%
396 }%
397
398 \WithSuffix\newcommand\sublinenumberstyle*[1]{%
399   \sublinenumberstyle{#1}%
400   \sublinenumberstyleR{#1}%
401 }%
402 %
403 %

```

## IV.6 Print marginal line number

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

```

404 \newif\iflinenumberLevenifblank
405 \newif\iflinenumberRevenifblank
406 %

```

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

```

407 \newcommand*\leftlinenumR{%
408   \l@dlinenumR
409   \kern\linenumsep}
410 \newcommand*\rightlinenumR{%
411   \kern\linenumsep
412   \l@dlinenumR}
413 \newcommand*\l@dlinenumR{%
414   \numlabfont\linenumrepR{\line@numR}\@Rlineflag%
415   \ifsublines@
416     \ifnum\subline@num>\z@
417       \unskip\fullstop\sublinenumrepR{\subline@numR}%
418     \fi
419   \fi}
420
421 %

```

## 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.

```
422 \newcount\line@numR
423 \newcount\subline@numR
424 \newcount\absline@numR
425
426 %
```

`\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:

```
427 \list@create{\line@listR}
428 \list@create{\insertlines@listR}
429 \list@create{\actionlines@listR}
430 \list@create{\actions@listR}
431
432 %
```

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

```
433 \newcount\page@numR
434
435 %
```

### IV.7.2 Specific to `reledpar`

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

```
436 \list@create{\linesinpar@listL}
437 \list@create{\linesinpar@listR}
438 \list@create{\maxlinesinpar@list}
439
440 %
```

## IV.8 Reading the line-list file

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

```

441 \newcommand{\list@clearing@regR}{%
442   \list@clear{\line@listR}%
443   \list@clear{\insertlines@listR}%
444   \list@clear{\actionlines@listR}%
445   \list@clear{\actions@listR}%
446   \list@clear{\linesinpar@listR}%
447   \list@clear{\linesonpage@listR}
448 }
449 %

```

`\@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.

```

450 \newcommand{\@par@sync@option}[1]{%
451   \IfStrEq{#1}{\@par@this@sync@option}%
452   {%
453     {\ifledRcol%
454       \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}%
455     }%
456     {\else%
457       \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}%
458     }%
459     {\fi%
460       \endinput%
461     }%
462   }%
463 }%
464 %

```

`\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.

```

462 \newcommand{\@nl@regR}{%
463   \ifx\l@dchset@num\relax \else
464     \advance\absline@numR \@ne

```

```

465 \set@line@action
466 \let\l@dchset@num\relax
467 \advance\absline@numR \m@ne
468 \advance\line@numR \m@ne% % do we need this?
469 \fi
470 \advance\absline@numR \@ne
471 \ifx\next@page@numR\relax \else
472 \page@action
473 \let\next@page@numR\relax
474 \fi
475 \ifx\sub@change\relax \else
476 \ifnum\sub@change>\z@
477 \sublines@true
478 \else
479 \sublines@false
480 \fi
481 \sub@action
482 \let\sub@change\relax
483 \fi
484 \ifcase\@lockR
485 \or
486 \@lockR \tw@
487 \or\or
488 \@lockR \z@
489 \fi
490 \ifcase\sub@lockR
491 \or
492 \sub@lockR \tw@
493 \or\or
494 \sub@lockR \z@
495 \fi
496 \ifsublines@
497 \ifnum\sub@lockR<\tw@
498 \advance\subline@numR \@ne
499 \fi
500 \else
501 \ifnum\@lockR<\tw@
502 \advance\line@numR \@ne \subline@numR \z@
503 \fi
504 \fi}
505
506
507 %

```

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

```

508 \newcount\last@page@numR
509 \last@page@numR=-10000
510

```

511 %

- \@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`.

```

512 \newcount\@lockR
513 \newcount\sub@lockR
514
515 \newcommand*{\do@lockonR}{%
516   \xright@appenditem{\the\absline@numR}{\to\actionlines@listR
517     \ifsublines@
518       \xright@appenditem{-1005}{\to\actions@listR
519         \ifnum\sub@lockR=\z@
520           \sub@lockR \@ne
521         \else
522           \ifnum\sub@lockR=\thr@@
523             \sub@lockR \@ne
524           \fi
525         \fi
526       \else
527         \xright@appenditem{-1003}{\to\actions@listR
528           \ifnum\@lockR=\z@
529             \@lockR \@ne
530           \else
531             \ifnum\@lockR=\thr@@
532               \@lockR \@ne
533             \fi
534           \fi

```

```

535 \fi}
536
537 %

```

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

```

\skip@lockoff
538
539
540 \newcommand{\do@lockoffR}{%
541   \xright@appenditem{\the\absline@numR}{\to\actionlines@listR
542   \ifsublines@
543     \xright@appenditem{-1006}{\to\actions@listR
544     \ifnum\sub@lockR=\tw@
545       \sub@lockR \thr@@
546     \else
547       \sub@lockR \z@
548     \fi
549   \else
550     \xright@appenditem{-1004}{\to\actions@listR
551     \ifnum\@lockR=\tw@
552       \@lockR \thr@@
553     \else
554       \@lockR \z@
555     \fi
556   \fi}
557
558 %
559

```

`\n@num`

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

`\@ref@regR` `\@ref@regR`

`\insert@countR`

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

```

560 \newcount\insert@countR
561 %

```

- #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.

```

562 \newcommand*{\@ref@regR}[2]{%
563   \global\advance\@edtext@level by 1%
564   \global\insert@countR=#1\relax
565   \loop\ifnum\insert@countR>\z@
566     \xright@appenditem{\the\absline@numR}\to\insertlines@listR
567     \global\advance\insert@countR \m@ne
568   \repeat
569 %

```

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.

```

570 \begingroup
571   \let\@ref=\dummy@ref
572   \let\@lopR\@gobble
573   \let\page@action=\relax
574   \let\sub@action=\relax
575   \let\set@line@action=\relax
576   \let\@lab=\relax
577   \let\@lemma=\relax
578   \let\@sw\@gobblethree%
579   #2
580   \global\endpage@num=\page@numR
581   \global\endline@num=\line@numR
582   \global\endsubline@num=\subline@numR
583 \endgroup
584 %

```

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

```

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

```

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

```

591 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
@edtext@level\endcsname}%
592 %

```

Declare and init boolean for lemma in this level.

```

593 \providebool{lemmacommand@\the\@edtext@level}%

```



```

594 \boolfalse{lemmacommand@\the\@edtext@level}%
595 %

```

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

```

596 #2
597 % Now, we store the list of \protect\cs{@sw} of this current \protect\cs{
edtext} as an element of
598 % the global list of list of \protect\cs{@sw} for a \protect\cs{edtext}
depth.
599 % \begin{macrocode}
600 \ifnum\@edtext@level>0%
601 \def\create@this@edtext@level{\expandafter\list@create\expandafter{
csname sw@list@edtextR@\the\@edtext@level\endcsname}}%
602 \ifcsundef{sw@list@edtextR@\the\@edtext@level}{\
create@this@edtext@level}{}%
603 \letcs{\@tmp}{sw@list@edtextR@\the\@edtext@level}%
604 \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}%
605 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
606 \global\cslet{sw@list@edtextR@\the\@edtext@level}{\@tmp}%
607 \fi%
608 %

```

Decrease edtext level counter.

```

609 \global\advance\@edtext@level by -1%
610 }
611 %

```

**\@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`.

```

612 \renewcommand*{\@pend}[1]{%
613 \ifbypstart@\global\line@num=0\fi%
614 \xright@appenditem{#1}\to\linesinpar@listL}
615 \renewcommand*{\@pendR}[1]{%
616 \ifbypstartR@\global\line@numR=0\fi
617 \xright@appenditem{#1}\to\linesinpar@listR}
618
619 %

```

**\@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

```

620 \newcommand{\@pstart}[3]{%
621 \ifcsdef{minpage@pstart@#1}%
622 {\ifnumgreater{#2}{\csuse{minpage@pstart@#1}}}%
623 {\csnumgdef{minpage@pstart@#1}{#2}}%

```

```

624     }%
625   }%
626   {\csnumgdef{minpage@pstart@#1}{#2}}
627   \csnumgdef{afterlines@pstart@#1L}{#3}%
628 }%
629
630 \newcommand{\@pstartR}[3]{%
631   \numdef{\@tmp}{#2-1}%Because we have not to know in which page the pstart
632   starts, but in which pair of facing page
633   \ifcsdef{minpage@pstart@#1}%
634     {\ifnumgreater{\@tmp}{\csuse{minpage@pstart@#1}}%
635       {\csnumgdef{minpage@pstart@#1}{\@tmp}}%
636     }%
637     {\csnumgdef{minpage@pstart@#1}{\@tmp}}
638   \csnumgdef{afterlines@pstart@#1R}{#3}%
639 }%
640 %

```

`\@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`.

```

641 \renewcommand*{\@lopL}[1]{%
642   \xright@appenditem{#1}\to\linesonpage@listL}
643 \renewcommand*{\@lopR}[1]{%
644   \xright@appenditem{#1}\to\linesonpage@listR}
645
646 %

```

## 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`.

```

647 \newwrite\linenum@outR
648 %

```

`\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
649 \newif\iffirst@linenum@out@R
650 \first@linenum@out@Rtrue
651 %

```

**\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.

```

652 \newcommand*{\line@list@stuffR}[1]{%
653   \read@linelist{#1}%
654   \iffirst@linenum@outR
655     \immediate\closeout\linenum@outR
656     \global\first@linenum@outRfalse
657     \immediate\openout\linenum@outR=#1
658     \immediate\write\linenum@outR{\string\line@list@version{\
this@line@list@version}}%
659     \ifl@dpaging%
660       \immediate\write\linenum@outR{\string\@par@sync@option{\
@par@this@sync@option}}%
661     \fi%
662   \else
663     \if@minipage%
664       \leavevmode%
665     \fi%
666     \closeout\linenum@outR%
667     \openout\linenum@outR=#1%
668   \fi}
669
670 %

```

**\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.

```

671 \newcommand*{\new@lineL}{%
672   \write\linenum@out{\string\@nl[\the\c@page][\thepage]}}
673 %

```

**\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.

```

674 \newcommand*{\new@lineR}{%
675   \write\linenum@outR{\string\@nl[\the\c@page][\thepage]}}
676 %

```

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

**\startsub** `\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` 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`.
- `\endlock`
- `\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@par  \newcommand{\newseries@par}[1]{%
678             %
```

#### 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 boolean flags, and to set them to true when a note series is not printed on one side. We check the `nofamiliar` and `nocritical` `Eledmac` options.

```
679     \unless\ifnofamiliar@%
680     \csgdef{onlysideX@#1}{}%
681     \global\newbool{keepforsideX@#1}%
682     \fi%
```

```

683 \unless\ifnocritical@%
684 \global\newbool{keepforXside@#1}%
685 \csgdef{Xonlyside@#1}{}%
686 \fi%
687 %

```

## V.2 Tools specific to familiar footnotes

```

688 \unless\ifnofamiliar@%
689 %

```

### 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.

```

690 \newcounter{footnote#1@typeset}%
691 %

```

### 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.

```

692 \expandafter\list@create\csname footnote#1@mk\endcsname%
693 %

```

Then, declare the `\footnoteXnomk` command.

```

694 \expandafter\newcommand\csname footnote#1nomk\endcsname[1]{%
695 %

```

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

```

696 \toggletrue{nomk}%
697 \csuse{footnote#1}{##1}%
698 \togglefalse{nomk}%
699 %

```

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`.

```

700 \letcs{\@tmp}{footnote#1@mk}%
701 \numdef\@tmpa{\csuse{c@footnote#1}}%
702 \global\xright@appenditem{\@tmpa}\to\@tmp%
703 \global\cslet{footnote#1@mk}{\@tmp}%
704 }%
705 %

```

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

```

706 \expandafter\newcommand\csname footnote#1mk\endcsname{%
707 %

```

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

```

708 \letcs{\@tmp}{footnote#1@mk}%
709 \gl@p\@tmp\to\@tmpa%
710 \global\cslet{footnote#1@mk}{\@tmp}%
711 %

```

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

```

712 \letcs{\old@footnote}{c@footnote#1}%
713 \setcounter{footnote#1}{\@tmpa}%
714 %

```

Define the footnote mark and print it

```

715 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
716 \csuse{@footnotemark#1}%
717 %

```

Restore previous footnote counter and finally add space.

```

718 \setcounter{footnote#1}{\old@footnote}%
719 \xspace%
720 }%
721 %

```

End of tools specific to familiar notes.

```

722 \fi
723 %
    End of \newseries@par.
724 }%
725 %

```

### V.2.3 Get correct footnote number

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

```

726 \newcommand{\save@familiarfootnote@number}{%
727   \unless\ifnofamiliar%
728     \def\do##1{\csxdef{saved@footnote##1}{\the\csname c@footnote##1\
endcsname}}%
729     \dolistloop{\@series}%
730     \fi%
731     \xdef\saved@footnote{\the\c@footnote}%
732   }
733 \newcommand{\get@familiarfootnote@number}{%
734   \unless\ifnofamiliar%
735     \def\do##1{\setcounter{footnote##1@typeset}{\csuse{saved@footnote##1}}}
736     %
737     \dolistloop{\@series}%
738     \fi%
739     \setcounter{footnote@typeset}{\saved@footnote}%
740   }
741   %

```

## V.3 Create hooks

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

```

741 \unless\ifnocritical%
742   \newhookcommand@series{Xonlyside}%
743   \fi%
744   \unless\ifnofamiliar%
745     \newhookcommand@series{onlysideX}%
746   \fi
747
748
749 %

```

## 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`.

```

750 \newcommand{\init@series@par}{%
751   \def\do##1{\newseries@par{##1}}%
752   \dolistloop{\@series}%
753 }%
754 \init@series@par%
755 %

```

## V.5 Tools specific to $\text{\LaTeX}$ 's classical footnotes

As users can use classical footnotes of  $\text{\LaTeX}$  (`\footnote`) in parallel texts, we must integrate the same tools to get correct number as for `reledmac`' footnotes (V.2.1 p. 45).

```

\footnote@reading56 \newcount\footnote@reading%
\footnote@typeset57 \newcounter{footnote@typeset}%
758 %

```

# 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@pc59 \list@create{\list@pstartL@pc}%
\list@pstartR@pc60 \list@create{\list@pstartR@pc}%
761 %

```

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@pc62 \def\dump@pstartL@pc{%
\dump@pstartR@pc63   \xright@appenditem{\the\c@pstartL}{\to\list@pstartL@pc%
764   \global\cslet{numberpstart@L\the\l@dumpstartsL}{\ifnumberpstart}%
765 }%
766
767 \def\dump@pstartR@pc{%

```



```

768 \xright@appenditem{\the\c@pstartR}\to\list@pstartR@pc%
769 \global\cslet{numberpstartR\the\l@dnumpstartsR}{\ifnumberpstart}%
770 }%
771
772 %

```

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

`\restore@pstartR@pc`

```

773 \def\restore@pstartL@pc{%
774 \ifx\list@pstartL@pc\empty\else%
775 \gl@p\list@pstartL@pc\to\@temp%
776 \global\c@pstartL=\@temp%
777 \fi%
778 }%
779 \def\restore@pstartR@pc{%
780 \ifx\list@pstartR@pc\empty\else%
781 \gl@p\list@pstartR@pc\to\@temp%
782 \global\c@pstartR=\@temp%
783 \fi%
784 }%
785 %

```

## 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.

```

786 \newenvironment{pairs}{%}
787 \l@dpairingtrue
788 \l@dpagingfalse
789 \initnumbering@quote
790 \save@familiarfootnote@number%
791 \if@ledgroup%
792 \get@familiarfootnote@number%
793 \fi%
794 \save@section@number%
795 \at@begin@pairs%
796 }{%
797 \l@dpairingfalse
798 }
799
800 %

```

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

```

801 \newcommand{\AtBeginPairs}[1]{\xdef\at@begin@pairs{#1}}%
802 \def\at@begin@pairs{}%
803
804 %

```

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.

```

805 \newenvironment{pages}{%
806   \l@dpairingtrue
807   \l@dpagingtrue
808   \initnumbering@quote
809   \save@familiarfootnote@number%
810   \if@ledgroup%
811     \get@familiarfootnote@number%
812   \fi%
813   \save@section@number%
814   \setlength{\Lcolwidth}{\textwidth}%
815   \setlength{\Rcolwidth}{\textwidth}%
816 }{%
817   \l@dpairingfalse
818   \l@dpagingfalse
819 }
820
821 %

```

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

```

822 \newif\ifinstanzaL
823 \newif\ifinstanzaR
824 %

```

**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.

```

825 \newenvironment{Leftside}{%
826   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname\else%
827     \led@err@Leftside@PreviousNotPrinted%
828   \fi%
829   \ledRcolfalse
830   \setcounter{pstartL}{1}
831   \let\pstart\pstartL
832   \let\thepstart\thepstartL
833   \let\pend\pendL
834   \let\memorydump\memorydumpL
835   \Leftsidehook

```

```

836 \let\old@startstanza\@startstanza
837 \def\@startstanza[##1]{\global\instanzaLtrue\old@startstanza[##1]}
838 }{
839 \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname%
840 \led@error@missing@numbering{Leftside}%
841 \fi%
842 \Leftsidehookend}
843 %

```

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

```

\Leftsidehookend
\Rightsidehook
\Rightsidehookend
844 \newcommand*{\Leftsidehook}{}
845 \newcommand*{\Leftsidehookend}{}
846 \newcommand*{\Rightsidehook}{}
847 \newcommand*{\Rightsidehookend}{}
848
849 %

```

**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.

```

850 \newenvironment{Rightside}{%
851 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else%
852 \led@err@Rightside@PreviousNotPrinted%
853 \fi%
854 \ledRcoltrue
855 \let\beginnumbering\beginnumberingR
856 \let\endnumbering\endnumberingR
857 \let\pausenumbering\pausenumberingR
858 \let\resumenumbering\resumenumberingR
859 \let\memorydump\memorydumpR
860 \let\thepstart\thepstartR
861 \let\pstart\pstartR
862 \let\pend\pendR
863 \let\ledpb\ledpbR
864 \let\lednopb\lednopbR
865 \let\lineation\lineationR
866 \Rightsidehook
867 \let\old@startstanza\@startstanza
868 \def\@startstanza[##1]{\global\instanzaRtrue\old@startstanza[##1]}
869 }{%
870 \ledRcolfalse
871 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
872 \led@error@missing@numbering{Rightside}%
873 \fi%
874 \Rightsidehookend
875 }
876

```

877 %

## 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.

```
878 \newcount\num@linesR
879 \newbox\one@lineR
880 \newcount\par@lineR
881 %
```

`\pstartL` `\pstart` starts the paragraph by clearing the `\inserts@list` list and other relevant variables, and then arranges for the subsequent text to go into the appropriate box. `\pstart` needs to appear at the start of every paragraph that 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.

```
882
883 \newcounter{pstartL}
884 \renewcommand{\thepstartL}{\bfseries\@arabic\c@pstartL}. }
885 \newcounter{pstartR}
886 \renewcommand{\thepstartR}{\bfseries\@arabic\c@pstartR}. }
887
888 \newcommandx*\pstartL}[1][1]{%
889   \if@nbreak%
890     \let\@oldnbreak\@nbreaktrue%
891   \else%
892     \let\@oldnbreak\@nbreakfalse%
```

```

893 \fi%
894 \@nbreaktrue%
895 \ifluatex%
896 \xdef\l@luatextextdir@L{\the\textdir}%
897 \xdef\l@luatexpardir@L{\the\pardir}%
898 \xdef\l@luatexbodydir@L{\the\bodydir}%
899 \fi%
900 \ifnumbering \else%
901 \led@err@PstartNotNumbered%
902 \beginnumbering%
903 \fi%
904 \ifnumberedpar%
905 \led@err@PstartInPstart%
906 \pend%
907 \fi%
908 %

```

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

```

909 \ifpst@rtedL\else%
910 \list@clear{\inserts@list}%
911 \global\let\next@insert=\empty%
912 \global\pst@rtedLtrue%
913 \fi%
914 \begingroup\everypar{}%
915 %

```

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.

```

916 \global\advance\l@dnumpstartsL \@ne%
917 \ifnum\l@dnumpstartsL>\l@dc@maxchunks%
918 \led@err@TooManyPstarts%
919 \global\l@dnumpstartsL=\l@dc@maxchunks%
920 \fi%
921 \global\setnamebox{\l@dLcolrawbox\the\l@dnumpstartsL}=\vbox\bgroup%
922 %

```

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.

```

923 \l@dzeropenalties%
924 \ifautopar\else%
925 \ifnumberpstart%
926 \ifsidepstartnum%
927 \else%
928 \thepstartL%
929 \fi%
930 \fi%
931 \fi%

```

```

932 \hsize=\Lcolwidth%
933 \numberedpar@true%
934 \iflabelpstart\protected@edef\@currentlabel%
935     {\p@pstartL\thepstartL}\fi%
936 %

```

Dump the optional arguments

```

937 \ifstrempy{#1}%
938     {\csgdef{before@pstartL@the\l@dnumpstartsL}{\at@every@pstart}}%
939     {\csgdef{before@pstartL@the\l@dnumpstartsL}{\noindent#1}}%
940     \at@every@pstart@call%
941 %

```

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

```

942 \ignorespaces%
943 %

```

```

944 }
945 %

```

The same for right side.

```

946 \newcommandx*{\pstartR}[1][1]{%
947     \if@nbreak%
948         \let\@oldnbreak\@nbreaktrue%
949     \else%
950         \let\@oldnbreak\@nbreakfalse%
951     \fi%
952     \@nbreaktrue%
953     \ifluatex%
954         \xdef\l@luatextextdir@R{\the\textdir}%
955         \xdef\l@luatexpardir@R{\the\pardir}%
956         \xdef\l@luatexbodydir@R{\the\bodydir}%
957     \fi%
958     \ifnumberingR \else%
959         \led@err@PstartNotNumbered%
960         \beginnumberingR%
961     \fi%
962     \ifnumberedpar@%
963         \led@err@PstartInPstart%
964         \pendR%
965     \fi%
966     \ifpst@rtedR\else%
967         \list@clear{\inserts@listR}%
968         \global\let\next@insertR=\empty%
969         \global\pst@rtedRtrue%
970     \fi%
971     \begingroup\everypar{}%
972     \global\advance\l@dnumpstartsR \@one%
973     \ifnum\l@dnumpstartsR>\l@dc@maxchunks%

```

```

974 \led@err@TooManyPstarts%
975 \global\l@dnumpstartsR=\l@dc@maxchunks%
976 \fi%
977 \global\setnamebox{l@dRcolrawbox\the\l@dnumpstartsR}=\vbox\bgroup%
978 \l@dzeropenalties%
979 \ifautopar\else%
980 \ifnumberpstart%
981 \ifsidepstartnum\else%
982 \thepstartR%
983 \fi%
984 \fi%
985 \fi%
986 \hsize=\Rcolwidth%
987 \numberedpar@true%
988 \iflabelpstart\protected@edef\@currentlabel%
989 {\p@pstartR\thepstartR}\fi%
990 \ifstrempy{#1}%
991 {\csgdef{before@pstartR@the\l@dnumpstartsR}{\at@every@pstart}}%
992 {\csgdef{before@pstartR@the\l@dnumpstartsR}{\noindent#1}}%
993 \at@every@pstart@call%
994 \ignorespaces%
995 }
996 %

```

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

```

997 \newcommandx*\pendL}[1][1]{%
998 \ifnumbering \else%
999 \led@err@PendNotNumbered%
1000 \fi%
1001 \ifnumberedpar@ \else%
1002 \led@err@PendNoPstart%
1003 \fi%
1004 %

```

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.

```

1005 \endgraf\global\num@lines=\prevgraf\egroup%
1006 \global\par@line=0%
1007 %

```

End the group that was begun in the \pstart.

```

1008 \endgroup%
1009 \ignorespaces%
1010 \@oldnbreak%
1011 \dump@pstartL@pc%
1012 \ifnumberpstart%
1013 \addtocounter{pstartL}{1}%
1014 \fi

```

```

1015 \parledgroup@beforenotes@save{L}%
1016 %
Dump content of the optional argument.
1017 \ifstrempy{#1}%
1018   {\csgdef{after@pendL@the\l@dnumstartsL}{\at@every@pend}}%
1019   {\csgdef{after@pendL@the\l@dnumstartsL}{\noindent#1}}%
1020 }
1021 %

```

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

```

1022 \newcommandx*{\pendR}[1][1]{%
1023   \ifnumberingR \else%
1024     \led@err@PendNotNumbered%
1025   \fi%
1026   \ifnumberedpar@ \else%
1027     \led@err@PendNoPstart%
1028   \fi%
1029   \endgraf\global\num@linesR=\prevgraf\egroup%
1030   \global\par@lineR=0%
1031   \endgroup%
1032   \ignorespaces%
1033   \@oldnobreak%
1034   \dump@pstartR@pc%
1035   \ifnumberpstart%
1036     \addtocounter{pstartR}{1}%
1037   \fi%
1038   \parledgroup@beforenotes@save{R}%
1039   \ifstrempy{#1}%
1040     {\csgdef{after@pendR@the\l@dnumstartsR}{\at@every@pend}}%
1041     {\csgdef{after@pendR@the\l@dnumstartsR}{\noindent#1}}%
1042 }
1043 %
1044 %

```

**\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.

```

1045 \newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
1046 \gdef\at@every@pstart@call{}%
1047 %

```

**\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**

```

1048 \newif\ifprint@last@after@pendL%
1049 \newif\ifprint@last@after@pendR%
1050 %

```



## 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  
`\l@drightbox` text.

```
1051 \newbox\l@dleftbox
1052 \newbox\l@drightbox
1053
1054 %
```

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

```
\countRline
1055 \newcount\countLline
1056 \countLline \z@
1057 \newcount\countRline
1058 \countRline \z@
1059
1060 %
```

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

```
1061 \newcount\@donereallinesL
1062 \newcount\@donetotallinesL
1063 \newcount\@donereallinesR
1064 \newcount\@donetotallinesR
1065
1066 %
```

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

```
1067 \newcommand*\do@lineL{%
1068 \letcs{\ifnumberpstart}{numberpstart@L\the\l@dpscL}%
1069 \advance\countLline \@ne%
1070 \ifvbox\namebox{\l@dLcolrawbox\the\l@dpscL}%
1071 {\vbadness=10000%
1072 \splittopskip=\z@%
1073 \do@lineLhook%
1074 \l@demptyd@ta%
1075 \global\setbox\one@line=\vsplit\namebox{\l@dLcolrawbox\the\l@dpscL}%
1076 to\baselineskip}%
1077 \IfStrEq{\splitfirstmarks\parledgroup@}{\begin}{\
parledgroup@notes@startL}{}%
1078 \unvbox\one@line \global\setbox\one@line=\lastbox%
1079 \@writepageofparL%
```

```

1080 \getline@numL%
1081 \ifnum\@lock>\@ne%
1082   \inserthangingsymboltrue%
1083 \else%
1084   \inserthangingsymbolfalse%
1085 \fi%
1086 \setbox\l@dleftbox%
1087 \hb@xt@ \Lcolwidth{%
1088   \ifl@dhiddenumber%
1089     \global\l@dhiddenumberfalse%
1090     \f@x@l@cks%
1091   \else%
1092     \affixline@num%
1093   \fi%
1094   \xifinlist{\the\l@dpscL}{\eled@sections@@}%
1095   {%
1096     \if@firstlineofpage%
1097       \set@Xtxtbeforenotes%
1098       \global\@firstlineofpagefalse%
1099     \fi%
1100     \add@inserts%
1101     \insert@msdata%
1102     \affixside@note%
1103   }%
1104   {\print@lineL}%
1105 }%
1106 \add@penaltiesL%
1107 \global\advance\@donereallinesL\@ne%
1108 \global\advance\@donetotallinesL\@ne%
1109 \else%
1110   \iflinenumberLevenifblank
1111     \new@lineL%
1112     \l@emptyd@ta%
1113     \getline@numL%
1114     \affixline@num%
1115     \setbox\l@dleftbox \hb@xt@ \Lcolwidth{%
1116       \l@dld@ta%
1117       \hspace*{\Lcolwidth}%
1118       \ledrlfill\l@drd@ta%
1119     }%
1120   \else%
1121     \setbox\l@dleftbox \hb@xt@ \Lcolwidth{\hspace*{\Lcolwidth}}%
1122   \fi%
1123   \global\advance\@donetotallinesL\@ne%
1124 \fi%
1125 }%
1126
1127
1128 %

```

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

```

1129 \def\print@lineL{%
1130   \affixpstart@numL%
1131   \l@dld@ta%
1132   \if@firstlineofpage%
1133     \set@Xtxtbeforenotes%
1134     \global\@firstlineofpagefalse%
1135   \fi%
1136   \insert@msdata%
1137   \add@inserts\affixside@note%
1138   \l@dlsn@te%
1139   \hb@xt@ \Lcolwidth{\ledllfill\hb@xt@ \wd\one@line{%
1140     \do@insidelineLhook%
1141     \ifluatex%
1142       \texdir\l@luatextextdir@L%
1143     \fi%
1144     \new@lineL%
1145     \inserthangingsymbolL%
1146     \l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta%
1147   \l@drsn@te}}%
1148
1149 %

```

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

```

1150 \def\print@eledsectionL{%%
1151   \addtocounter{pstartL}{-1}%
1152   \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{%
1153     \ifdefstring{\@eledsectmark}{L}{\ledsectnomark}%
1154     \numdef{\temp@}{\l@dpscL-1}%
1155     \xifinlist{\temp@}{\eled@sections@}{\@nbreaktrue}{\@nbreakfalse}%
1156     \@eled@sectioningtrue%
1157     \bgroup%
1158       \ifluatex%
1159         \texdir\l@luatextextdir@L%
1160         \pardir\l@luatexpardir@L%
1161         \bodydir\l@luatexbodydir@L%
1162         \ifdefstring{\l@luatextextdir@L}{TRT}{\@RTLtrue}{}%
1163       \fi%
1164       \csuse{eled@sectioning@the\l@dpscL}%
1165     \egroup%
1166     \@eled@sectioningfalse%
1167     \global\csundef{eled@sectioning@the\l@dpscL}%
1168     \if@RTL%
1169       \hspace{-3\paperwidth}%
1170     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1171   \else%
1172     \hspace{3\paperwidth}%

```

```

1173 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1174 \fi%
1175 \vskip\eledsection@correcting@skip%
1176 }
1177
1178 %

```

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

```

\doinssideLhook \newcommand*\dolineLhook[1]{\gdef\dolineLhook{#1}}%
\doinssideRhook \newcommand*\dolineRhook[1]{\gdef\dolineRhook{#1}}%
1179 \newcommand*\doinssideLhook[1]{\gdef\doinssideLhook{#1}}%
1180 \newcommand*\doinssideRhook[1]{\gdef\doinssideRhook{#1}}%
1181
1182 %
1183
1184 %

```

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

```

\dolineLhook \newcommand*\do@lineLhook{}
\dolineRhook \newcommand*\do@lineRhook{}
1185 \do@insidelineLhook \newcommand*\do@insidelineLhook{}
1186 \do@insidelineRhook \newcommand*\do@insidelineRhook{}
1187
1188 %
1189
1190 %

```

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

```

1191 \newcommand*\do@lineR{%
1192   \let\linenumrepL\linenumrep%
1193   \let\sublinenumrepL\sublinenumrep%
1194   \let\linenumrepR\linenumrep%
1195   \let\sublinenumrepR\sublinenumrep%
1196   \letcs{ifnumberpstart}{numberpstart@R\the\l@dpscR}%
1197   \ledRcol@true%
1198   \advance\countRline \@ne%
1199   \ifvbox\namebox{\l@dRcolrawbox\the\l@dpscR}%
1200     {\vbadness=10000%
1201      \splittopskip=\z@%
1202      \do@lineRhook%
1203      \l@demtyd@ta%
1204      \global\setbox\one@lineR=\vsplit\namebox{\l@dRcolrawbox\the\l@dpscR}%
1205        to\baselineskip}%
1206     \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
1207       \parledgroup@notes@startR}{\
1208       \unvbox\one@lineR \global\setbox\one@lineR=\lastbox%
1209       \writepageofparR%
1210       \getline@numR%
1211       \ifnum\@lockR>\@ne%

```

```

1211 \inserthangingsymbolRtrue%
1212 \else%
1213 \inserthangingsymbolRfalse%
1214 \fi%
1215 \setbox\l@drightbox%
1216 \hb@xt@ \Rcolwidth{%
1217 \ifl@dhidenumber%
1218 \global\l@dhidenumberfalse%
1219 \f@x@l@cksR%
1220 \else%
1221 \affixline@numR%
1222 \fi%
1223 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
1224 {%
1225 \if@firstlineofpageR%
1226 \set@Xtxtbeforenotes%
1227 \global\@firstlineofpageRfalse%
1228 \fi%
1229 \insert@msdata%
1230 \add@insertsR%
1231 \affixside@noter%
1232 }%
1233 {\print@lineR}%
1234 }%
1235 \add@penaltiesR%
1236 \global\advance\@donereallinesR\@ne%
1237 \global\advance\@donetotallinesR\@ne%
1238 \else%
1239 \iflinenumberRevenifblank%
1240 \new@lineR
1241 \l@emptyd@ta%
1242 \getline@numR%
1243 \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1244 \affixline@numR%
1245 \l@dld@ta%
1246 \hspace*{\Rcolwidth}%
1247 \ledrlfill\l@drd@ta%
1248 }%
1249 \else%
1250 \setbox\l@drightbox \hb@xt@ \Rcolwidth{\hspace*{\Rcolwidth}}%
1251 \fi%
1252 \global\advance\@donetotallinesR\@ne%
1253 \fi%
1254 \ledRcol@false%
1255 \let\linenumrep\linenumrepL%
1256 \let\sublinenumrep\sublinenumrepL%
1257 }
1258
1259
1260 %

```

`\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.

```

1261 \newcommand*{\getline@numR}{%
1262   \global\advance\absline@numR \@ne
1263   \do@actionsR
1264   \do@ballastR
1265   \ifledgroupnotesR\else
1266     \ifnumberline
1267       \ifsublines@
1268         \ifnum\sub@lockR<\tw@
1269           \global\advance\subline@numR \@ne
1270         \fi
1271       \else
1272         \ifnum\@lockR<\tw@
1273           \global\advance\line@numR \@ne
1274           \global\subline@numR \z@
1275         \fi
1276       \fi
1277     \fi
1278   \fi
1279 }
1280 \newcommand*{\getline@numL}{%
1281   \global\advance\absline@num \@ne
1282   \do@actions
1283   \do@ballast
1284   \ifledgroupnotesL\else
1285     \ifnumberline
1286       \ifsublines@
1287         \ifnum\sub@lock<\tw@
1288           \global\advance\subline@num \@ne
1289         \fi
1290       \else
1291         \ifnum\@lock<\tw@
1292           \global\advance\line@num \@ne
1293           \global\subline@num \z@
1294         \fi
1295       \fi
1296     \fi
1297   \fi
1298 }
1299
1300 %
1301 %
```

`\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.

```

1302 \newcommand*{\do@ballastR}{\global\ballast@count=\z@
1303 \beginingroup
1304 \advance\absline@numR \@ne
1305 \ifnum\next@actionlineR=\absline@numR
1306 \ifnum\next@actionR>-1001
1307 \global\advance\ballast@count by -\c@ballast
1308 \fi
1309 \fi
1310 \endgroup}
1311 %

```

`\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.

```

1312 \newif\ifl@dskipversenumberR
1313 \newcommand*{\do@actions@fixedcodeR}{%
1314 \ifcase\@l@dttempcnta%
1315 \or% % 1001 = starting sublineation
1316 \global\sublines@true
1317 \or% % 1002 = ending sublineation
1318 \global\sublines@false
1319 \or% % 1003 = starting locking number
1320 \global\@lockR=\@ne
1321 \or% % 1004 = ending locking number
1322 \ifnum\@lockR=\tw@
1323 \global\@lockR=\thr@@
1324 \else
1325 \global\@lockR=\z@
1326 \fi
1327 \or% % 1005 = starting locking subnumber
1328 \global\sub@lockR=\@ne
1329 \or% % 1006 = ending locking subnumber
1330 \ifnum\sub@lockR=\tw@
1331 \global\sub@lockR=\thr@@
1332 \else
1333 \global\sub@lockR=\z@
1334 \fi
1335 \or% % 1007 = skipping numbering
1336 \l@dskipnumbertrue
1337 \or% % 1008 = skipping numbering in stanza
1338 \l@dskipversenumberRtrue%
1339 \or% % 1009 = hiding number
1340 \l@dhiddenumbertrue%
1341 \or% % 1010 = inserting msdata
1342

```

```

1343 \add@msdata%
1344 \else%
1345 \led@warn@BadAction
1346 \fi%
1347 }
1348
1349
1350 \newcommand*{\do@actionsR}{%
1351 \global\let\do@actions@nextR=\relax
1352 \@l@dttempcntb=\absline@numR
1353 \ifnum\@l@dttempcntb<\next@actionlineR\else
1354 \ifnum\next@actionR>-1001\relax
1355 \@firstlineofpageRtrue%
1356 \global\page@numR=\next@actionR
1357 \ifbypage@R
1358 \global\line@numR \z@ \global\subline@numR \z@
1359 \fi
1360 \add@msdata@firstlineofpage%
1361 \else
1362 \ifnum\next@actionR<-4999\relax % 9/05 added relax here
1363 \@l@dttempcnta=-\next@actionR
1364 \advance\@l@dttempcnta by -5001\relax
1365 \ifsublines@
1366 \global\subline@numR=\@l@dttempcnta
1367 \else
1368 \global\line@numR=\@l@dttempcnta
1369 \fi
1370 \else
1371 \@l@dttempcnta=-\next@actionR
1372 \advance\@l@dttempcnta by -1000\relax
1373 \do@actions@fixedcodeR
1374 \fi
1375 \fi
1376 \ifx\actionlines@listR\empty
1377 \gdef\next@actionlineR{1000000}%
1378 \else
1379 \gl@p\actionlines@listR\to\next@actionlineR
1380 \gl@p\actions@listR\to\next@actionR
1381 \global\let\do@actions@nextR=\do@actionsR
1382 \fi
1383 \fi
1384 \do@actions@nextR}
1385
1386 %

```

#### VIII.4 Line number printing

`\l@dcalcnum` `\affixline@numR` is the right text version of the `\affixline@num` macro.  
`\ch@cksub@l@ckR`  
`\ch@ck@l@ckR`  
`\f@x@l@cksR`  
`\affixline@numR`



```

1388 \newcommand*{\l@dcalcnm}[3]{%
1389   \ifnum #1 > #2\relax
1390     \l@dtempcnta = #1\relax
1391     \advance\l@dtempcnta by -#2\relax
1392     \divide\l@dtempcnta by #3\relax
1393     \multiply\l@dtempcnta by #3\relax
1394     \advance\l@dtempcnta by #2\relax
1395   \else
1396     \l@dtempcnta=#2\relax
1397   \fi}
1398
1399 \newcommand*{\ch@cksub@l@ckR}{%
1400   \ifcase\sub@lockR
1401   \or
1402     \ifnum\sublock@disp=\@ne
1403       \l@dtempcntb \z@ \l@dtempcnta \@ne
1404     \fi
1405   \or
1406     \ifnum\sublock@disp=\tw@
1407     \else
1408       \l@dtempcntb \z@ \l@dtempcnta \@ne
1409     \fi
1410   \or
1411     \ifnum\sublock@disp=\z@
1412       \l@dtempcntb \z@ \l@dtempcnta \@ne
1413     \fi
1414   \fi}
1415
1416 \newcommand*{\ch@ck@l@ckR}{%
1417   \ifcase\@lockR
1418   \or
1419     \ifnum\lock@disp=\@ne
1420       \l@dtempcntb \z@ \l@dtempcnta \@ne
1421     \fi
1422   \or
1423     \ifnum\lock@disp=\tw@
1424     \else
1425       \l@dtempcntb \z@ \l@dtempcnta \@ne
1426     \fi
1427   \or
1428     \ifnum\lock@disp=\z@
1429       \l@dtempcntb \z@ \l@dtempcnta \@ne
1430     \fi
1431   \fi}
1432
1433 \newcommand*{\f@x@l@cksR}{%
1434   \ifcase\@lockR
1435   \or
1436     \global\@lockR \tw@
1437   \or \or

```

```

1438 \global\@lockR \z@
1439 \fi
1440 \ifcase\sub@lockR
1441 \or
1442 \global\sub@lockR \tw@
1443 \or \or
1444 \global\sub@lockR \z@
1445 \fi}
1446
1447
1448 \newcommand*{\affixline@numR}{%
1449 \ifledgroupnotesR@ \else \ifnumberline
1450 \ifl@dskipnumber
1451 \global\l@dskipnumberfalse
1452 \else
1453 \ifsublines@
1454 \l@dttempcntb=\subline@numR
1455 \l@dcalcnun{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR
}%
1456 \ch@cksub@lockR
1457 \else
1458 \l@dttempcntb=\line@numR
1459 \ifx\linenumberlist\empty
1460 \l@dcalcnun{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%
1461 \else
1462 \l@dttempcnta=\line@numR
1463 \edef\rem@inder{\linenumberlist,\number\line@numR,%
1464 \edef\sc@n@list{\def\noexpand\sc@n@list
1465 ###1,\number\l@dttempcnta,###2|\def\noexpand\rem@inder{###2}}}%
1466 \sc@n@list\expandafter\sc@n@list\rem@inder|}%
1467 \ifx\rem@inder\empty\advance\l@dttempcnta\@ne\fi
1468 \fi
1469 \ch@ck@l@ckR
1470 \fi
1471 \ifnum\l@dttempcnta=\l@dttempcntb
1472 \ifl@dskipversenumberR\else
1473 \if@twocolumn
1474 \if@firstcolumn
1475 \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1476 \else
1477 \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1478 \fi
1479 \else
1480 \l@dttempcntb=\line@marginR
1481 \ifnum\l@dttempcntb>\@ne
1482 \advance\l@dttempcntb by\page@numR
1483 \fi
1484 \ifodd\l@dttempcntb
1485 \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1486 \else

```

```

1487 \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1488 \fi
1489 \fi
1490 \fi
1491 \fi
1492 \f@x@l@cksR
1493 \fi
1494 \fi
1495 \fi}
1496 %

```

## 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@numL
\affixpstart@numR \newcommand*{\affixpstart@numL}{%
\leftpstartnumR \ifsidepstartnum
\rightpstartnumR \if@twocolumn
\leftpstartnumL \if@firstcolumn
\rightpstartnumL \gdef\l@dld@ta{\llap{\leftpstartnumL}}}%
\ifpstartnumR \else
\gdef\l@drd@ta{\rlap{\rightpstartnumL}}}%
\fi
\else
\@l@tempcntb=\line@margin
\ifnum\@l@tempcntb>\@ne
\advance\@l@tempcntb \page@num
\fi
\ifodd\@l@tempcntb
\gdef\l@drd@ta{\rlap{\rightpstartnumL}}}%
\else
\gdef\l@dld@ta{\llap{\leftpstartnumL}}}%
\fi
\fi
\fi
}
\newcommand*{\affixpstart@numR}{%
\ifsidepstartnum
\if@twocolumn
\if@firstcolumn
\gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
\else

```

```

1525 \gdef\l@drd@ta{\rlap{\rightpstartnumR}}}%
1526 \fi
1527 \else
1528 \l@dttempcntb=\line@marginR
1529 \ifnum\l@dttempcntb>\@ne
1530 \advance\l@dttempcntb \page@numR
1531 \fi
1532 \ifodd\l@dttempcntb
1533 \gdef\l@drd@ta{\rlap{\rightpstartnumR}}}%
1534 \else
1535 \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
1536 \fi
1537 \fi
1538 \fi
1539 }
1540
1541 \newcommand*\leftpstartnumL{%
1542 \ifpstartnum
1543 \thepstartL
1544 \kern\linenumsep\global\pstartnumfalse\fi
1545 }
1546 \newcommand*\rightpstartnumL{%
1547 \ifpstartnum\kern\linenumsep
1548 \thepstartL
1549 \global\pstartnumfalse\fi
1550 }
1551 \newif\ifpstartnumR
1552 \pstartnumRtrue
1553 \newcommand*\leftpstartnumR{%
1554 \ifpstartnumR
1555 \thepstartR
1556 \kern\linenumsep\global\pstartnumRfalse\fi
1557 }
1558 \newcommand*\rightpstartnumR{%
1559 \ifpstartnumR\kern\linenumsep
1560 \thepstartR
1561 \global\pstartnumRfalse\fi
1562 }
1563 %

```

## 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.

```

1564 \list@create{\inserts@listR}
1565 %

```

`\add@insertsR` The right text version.  
`\add@inserts@nextR`

```

1566 \newcommand*{\add@insertsR}{%
1567   \global\let\add@inserts@nextR=\relax
1568   \ifx\inserts@listR\empty \else
1569     \ifx\next@insertR\empty
1570       \ifx\insertlines@listR\empty
1571         \global\noteschanged@true
1572         \gdef\next@insertR{100000}%
1573       \else
1574         \gl@p\insertlines@listR\to\next@insertR
1575       \fi
1576     \fi
1577     \ifnum\next@insertR=\absline@numR
1578       \gl@p\inserts@listR\to\@insertR
1579       \@insertR
1580       \global\let\@insertR=\undefined
1581       \global\let\next@insertR=\empty
1582       \global\let\add@inserts@nextR=\add@insertsR
1583     \fi
1584   \fi
1585   \add@inserts@nextR}
1586
1587 %

```

## VIII.7 Penalties

**\add@penaltiesL** **\add@penaltiesR** **\add@penaltiesL** is the last macro used by **\do@lineL**. It adds up the club, widow, and interline penalties, and puts a single penalty of the appropriate size back into the paragraph; these penalties get removed by the **\vsplit** operation. **\displaywidowpenalty** and **\brokenpenalty** are not restored, since we have no easy way to find out where we should insert them.

In the code below, which is a virtual copy of the original **\add@penalties**, **\num@lines** is the number of lines in the whole paragraph, and **\par@line** is the line we are working on at the moment. The count **\@l@dttempcnta** is used to calculate and accumulate the penalty; it is initially set to the value of **\ballast@count**, which has been worked out in **\do@ballast**. Finally, the penalty is checked to see that it does not go below  $-10000$ .

```

\newcommand*{\add@penaltiesR}{\@l@dttempcnta=\ballast@count
\ifnum\num@linesR>\@ne
\global\advance\par@lineR \@ne
\ifnum\par@lineR=\@ne
\advance\@l@dttempcnta by \clubpenalty
\fi
\@l@dttempcntb=\par@lineR \advance\@l@dttempcntb \@ne
\ifnum\@l@dttempcntb=\num@linesR
\advance\@l@dttempcnta by \widowpenalty
\fi
\ifnum\par@lineR<\num@linesR
\advance\@l@dttempcnta by \interlinepenalty

```

```

\fi
\fi
\ifnum\@l@dttempcnta=\z@
\relax
\else
\ifnum\@l@dttempcnta>-10000
\penalty\@l@dttempcnta
\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.

```

1588 \newcommand*{\add@penaltiesL}{%
1589 \newcommand*{\add@penaltiesR}{%
1590
1591 %

```

## 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.

```

1592 \newcommand*{\flush@notesR}{%
1593 \xloop
1594 \ifx\inserts@listR\empty \else
1595 \glp\inserts@listR\to\@insertR
1596 \@insertR
1597 \global\let\@insertR=\undefined
1598 \repeat}
1599
1600 %

```

## 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>:

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

- If notes are to be printed in both sides, we just proceed the usual way: print the foot starts for the series, then the foot group.
- If notes are to be printed in the left side, we do these prints only for even pages ; if notes are to be printed in the right side, we do these prints only for odd pages.
- However, that is not enough. Because the problem does not only consists in printing notes in any particular page. It is also not to put aside room for notes in the pages where we do not want to print them. To take an example: if some note in the left side is too long by 160pt to be printed in full in the left page, we do not want to put aside 160pt a space for it in the following right page.
- To solve this problem, we change the magnification factor associated with notes before going to the next page. If we start a page where no notes are supposed to be printed, the magnification counter is set to 0. 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 and skip, read *The TeXbook* p. 122-125).
- There still remains a last problem. This problem is quite complex to understand, so an example will speak for itself. Suppose we allow 10 lines of notes by page. Suppose a long note, be it 25 lines, which needs three pages to be printed. Suppose it must be printed only on left pages, namely odd pages.

On p. 2, the first 10 lines of the notes are printed. On p. 3, the box associated to the notes contains 10 lines. However, as we are in a right page, we do not void this box. So  $\TeX$  will keep its content for the pages to come. However, on p. 4 it will also add one line in the footnote box, because in any case,  $\TeX$  adds some content in the box when preparing the output routines, even if there is some content left in this box from the previous pages. So the lines in the note box at p. 4 will be  $10 + 1 = 11$ . There is one line which should not be there. Furthermore, as the box size is for 10 lines and not for 11 lines, this last line will be glued to the previous one.

To fix this double issue:

- For the pages where notes must be NOT printed, we allow to every note box one line less than it ought to be. In our example, that means that we allow  $\TeX$  to add only  $10 - 1 = 9$  line in the note box on p. 3. Before shifting to the pages where notes must be printed, we allow to every notes the expected number of lines. In our example, that means that we allow  $\TeX$  to add 10 lines in the note box on p. 4. As on p. 3 only 9 lines were allowed, that means note box of p. 4 will contain  $9 + 1 = 10$  lines. So the “one line too many” problem is solved.
- Still remains the “glue” problem. We solve it by recreating a clean note box. We split the one which is created by  $\TeX$  to get the next line printed. Then, we create the new box, by bringing together the first part and the last part of the split box, adding some skip between them. That is achieved

by `\correct@Xfootins@box` (or `\correct@footinsX@box` for familiar notes).

The code to print critical notes, when processing `\Pages`.

```
1601 \newcommand\print@Xnotes@forpages[1]{%
1602 %
```

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

```
1603 \ifcseempty{Xonlyside@#1}{%
1604 \csuse{#1footstart}{#1}%
1605 \csuse{#1footgroup}{#1}%
1606 }%
1607 %
```

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

```
1608 {%
1609 \ifboolexpr{%
1610 ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page
1611 }})%
1612 or%
1612 (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}}))%
1613 }%
1614 %
```

If we are in a page where notes must be printed, print the notes, after having made the corrections which are needed for boxes.

```
1615 {%
1616 \correct@Xfootins@box{#1}%
1617 \csuse{#1footstart}{#1}%
1618 \csuse{#1footgroup}{#1}%
1619 %
```

Then, say not to keep room for notes in the next page.

```
1620 \global\count\csuse{#1footins}=0%
1621 \global\skip\csuse{#1footins}=0pt%
1622 %
```

And also, allow one line less for notes in the next page.

```
1623 \csuse{Xnotefontsize@#1}%
1624 \global\advance\dimen\csuse{#1footins} by -\baselineskip%
1625 %
```

Now we have printed the notes. So we put aside this fact.

```
1626 \global\boolfalse{keepforXside@#1}%
1627 }%
1628 %
```



In case we are on a page where notes must NOT be printed. First, memorize that we have not printed the notes, despite having some to print.

```
1629 {%
1630 \global\booltrue{keepforXside@#1}%
1631 %
```

Then restore expected rooms for notes on the next page.

```
1632 \global\count\csuse{#1footins}=\csuse{default@#1footins}%
1633 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
1634 %
```

Last but not least, restore the normal line number allowed to notes for the following page.

```
1635 \bgroup%
1636 \csuse{Xnotefontsize@#1}%
1637 \global\advance\dimen\csuse{#1footins} by \baselineskip%
1638 \egroup%
1639 %
```

```
1640 % End of \protect\cs{print@Xnotes@forpages}.
1641 }%
1642 }%
1643 }%
1644 %
```

Now, \correct@Xfootins@box, to fix problem of last line being glued to the previous one.

```
1645 \newcommand{\correct@Xfootins@box}[1]{%
1646 %
```

We need to make correction only in case we have not printed any note in the previous page, although there was to be “normally” printed.

```
1647 \ifbool{keepforXside@#1}{%
1648 %
```

Some setting needed to do the right splitting.

```
1649 \csuse{Xnotefontsize@#1}%
1650 \splittopskip=0pt%
1651 %
```

And now, split the last line, and push in the right place.

```
1652 \global\setbox\csuse{#1footins}=\vbox{%
1653 \vsplit\csuse{#1footins} to \dimexpr\ht\csuse{#1footins}-1pt\relax%
1654 \vskip \dimexpr-0.5\baselineskip-0.5\lineskip-0.5pt\relax%
1655 \unvbox\csuse{#1footins}%
1656 }%
1657 %
```

End of the macro.

```

1658 }{}%
1659 }%
1660 %

```

And now, the same for familiar footnotes.

```

1661 \newcommand\print@notesX@forpages[1]{%
1662   \ifcseempty{onlysideX@#1}{%
1663     \csuse{footstart#1}{#1}%
1664     \csuse{footgroup#1}{#1}%
1665   }%
1666   {%
1667     \ifboolexpr{%
1668       ((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page
1669       }%
1670       (test {\ifcsstring{onlysideX@#1}{R}} and test{\ifnumodd{\c@page}}))%
1671     }%
1672     {%
1673       \correct@footinsX@box{#1}%
1674       \csuse{footstart#1}{#1}%
1675       \csuse{footgroup#1}{#1}%
1676       \global\count\csuse{footins#1}=0%
1677       \global\skip\csuse{footins#1}=0pt%
1678       \csuse{notefontsizeX@#1}%
1679       \global\advance\dimen\csuse{footins#1} by -\baselineskip%
1680       \global\boolfalse{keepforsideX@#1}%
1681     }%
1682     {%
1683       \global\booltrue{keepforsideX@#1}%
1684       \global\count\csuse{footins#1}=\csuse{default@footins#1}%
1685       \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
1686       \bgroup%
1687         \csuse{notefontsizeX@#1}%
1688         \global\advance\dimen\csuse{footins#1} by \baselineskip%
1689       \egroup%
1690     }%
1691   }%
1692 }%
1693 \newcommand{\correct@footinsX@box}[1]{%
1694   \ifbool{keepforsideX@#1}{%
1695     \csuse{notefontsizeX@#1}%
1696     \splittopskip=0pt%
1697     \global\setbox\csuse{footins#1}=\vbox{%
1698       \vsplit\csuse{footins#1} to \dimexpr\ht\csuse{footins#1}-1pt\relax%
1699       \vskip \dimexpr-0.5\baselineskip-0.5\lineskip-0.5pt\relax%
1700       \unvbox\csuse{footins#1}%
1701     }%
1702   }{}%
1703 }%
1704 %

```

## 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.

```
1705 \list@create{\labelref@listR}
1706
1707 %
```

`\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`.

```
1708 \def\l@dmake@labelsR#1|#2|#3|#4|#5{%
1709   \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
else
1710   \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
1711   \fi
1712   \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
{#1|#2|#3|#4|\@Rlineflag}%
1713   \ignorespaces}
1714 \AtBeginDocument{%
1715   \def\l@dmake@labelsR#1|#2|#3|#4|#5{%
1716   }
1717
1718 %
```

`\@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*
1719 \WithSuffix\newcommand\sidenotemargin*[1]{%
1720   \l@dgetsidenote@margin{#1}
1721   \global\sidenote@marginR=\@l@dttempcntb
1722   \global\sidenote@margin=\@l@dttempcntb
1723 }
1724 \newcount\sidenote@marginR
1725 \global\sidenote@margin=\@ne
1726
1727 %
```

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

```

1728 \newcommand*{\affixside@noteR}{%
1729   \def\sidenotecontent@{%
1730     \numgdef{\itemcount@}{0}%
1731     \def\do##1{%
1732       \ifnumequal{\itemcount@}{0}%
1733         {%
1734           \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
1735         {\appto\sidenotecontent@{\sidenotesep ##1}%
1736         }%
1737         \numgdef{\itemcount@}{\itemcount@+1}%
1738       }%
1739       \dolistloop{\l@dcstotetext}%
1740       \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
1741       \gdef\@templ@d{%
1742         \gdef\@templ@n{\l@dcstotetext\l@dcstotetext@l\l@dcstotetext@r}%
1743         \ifx\@templ@d\@templ@n \else%
1744           \if@twocolumn%
1745             \if@firstcolumn%
1746               \setl@dlp@rbox{##1}{\sidenotecontent@}%
1747             \else%
1748               \setl@drp@rbox{\sidenotecontent@}%
1749             \fi%
1750           \else%
1751             \l@dttempcntb=\sidenote@marginR%
1752             \ifnum\l@dttempcntb>\@ne%
1753               \advance\l@dttempcntb by\page@numR%
1754             \fi%
1755             \ifodd\l@dttempcntb%
1756               \setl@drp@rbox{\sidenotecontent@}%
1757             \gdef\sidenotecontent@{%
1758               \numdef{\itemcount@}{0}%
1759               \dolistloop{\l@dcstotetext@l}%
1760               \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
1761               \setl@dlp@rbox{\sidenotecontent@}%
1762             \else%
1763               \setl@dlp@rbox{\sidenotecontent@}%
1764               \gdef\sidenotecontent@{%
1765                 \numdef{\itemcount@}{0}%
1766                 \dolistloop{\l@dcstotetext@r}%
1767                 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
1768                 \setl@drp@rbox{\sidenotecontent@}%
1769               \fi%
1770             \fi%
1771           \fi%
1772   }
1773
1774 %

```

## 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.

```

\inserthangingsymbolL% \newif\ifinserthangingsymbolR
\inserthangingsymbolR% \newcommand{\inserthangingsymbolL}{%
1777 \ifinserthangingsymbol%
1778 \ifinstanzaL%
1779 \hskip \@ifundefined{sza@00}{0}{\expandafter%
1780 \noexpand\csname sza@00\endcsname}\stanzaindentbase%
1781 \@hangingsymbol%
1782 \fi%
1783 \fi%
1784 }%
1785 \newcommand{\inserthangingsymbolR}{%
1786 \ifinserthangingsymbolR%
1787 \ifinstanzaR%
1788 \hskip \@ifundefined{sza@00}{0}{\expandafter%
1789 \noexpand\csname sza@00\endcsname}\stanzaindentbase%
1790 \@hangingsymbol%
1791 \fi%
1792 \fi%
1793 }%
1794 %

```

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.

```

1795 \chardef\next=\catcode`\&
1796 \catcode`\&=\active
1797
1798 %

```

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

```

1799 \newenvironment{astanza}[1][\&]{%
1800 \catcode`\&=\active
1801 \global\stanza@count\@ne\stanza@modulo\@ne

```

```

1802 \ifnum\usernamecount{sza@0@}=\z@
1803 \let\stanza@hang\relax
1804 \let\endlock\relax
1805 \else
1806 \rightskip\z@ plus 1fil\relax
1807 \fi
1808 \ifnum\usernamecount{szp@0@}=\z@
1809 \let\sza@penalty\relax
1810 \fi
1811 \def&{%
1812 \endlock\mbox{}%
1813 \sza@penalty
1814 \global\advance\stanza@count\@ne
1815 \@astanza@line}%
1816 \def\&{\@stopastanza}%
1817 \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}} and test{\ifstrempy
1818 {#1}}}%
1819 {\pstart[\at@every@stanza]}%
1820 {\pstart[#1]}%
1821 \@astanza@line
1822 \let\par\relax\ignorespaces%No paragraph in verses
1823 }{}
1824 %

```

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

```

1825 \newcommandx{\@stopastanza}[1][1,usedefault]{%
1826 \endlock\mbox{}%
1827 \ifboolexpr{not test{\ifdefvoid{\at@every@stop@stanza}} and test{\
1828 ifstrempy{#1}}}%
1829 {\pend[\at@every@stop@stanza]}%
1830 {\pend[#1]}%
1831 }%
1832 %

```

**\@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.

```

1832 \newcommand*{\@astanza@line}{%
1833 \ifnum\value{stanzaindentrepetition}=0
1834 \parindent=\csname sza@\number\stanza@count
1835 @\endcsname\stanzaindentbase
1836 \else
1837 \parindent=\csname sza@\number\stanza@modulo
1838 @\endcsname\stanzaindentbase
1839 \managestanza@modulo
1840 \fi
1841 \endgraf
1842 \stanza@hang%

```

```

1843 \ignorespaces}
1844
1845 %

```

Lastly reset the modified category codes.

```

1846 \catcode`\&=\next
1847
1848 %

```

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

```

\thestanzaR
1849 \newcounter{stanzaL}
1850 \newcounter{stanzaR}
1851 \renewcommand{\thestanzaL}{%
1852   \textbf{\arabic{stanzaL}}%
1853 }
1854 \renewcommand{\thestanzaR}{%
1855   \textbf{\arabic{stanzaR}}%
1856 }
1857 %
1858 %

```

## XIV Naming macros

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

`\newnamebox` A set of macros for creating and using ‘named’ boxes; the macros are called after the regular box macros, but including the string ‘name’.

```

\setnamebox
\unhnamebox
\unvnamebox
\namebox
1859 \providecommand*\newnamebox}[1]{%
1860   \expandafter\newbox\csname #1\endcsname}
1861 \providecommand*\setnamebox}[1]{%
1862   \expandafter\setbox\csname #1\endcsname}
1863 \providecommand*\unhnamebox}[1]{%
1864   \expandafter\unhbox\csname #1\endcsname}
1865 \providecommand*\unvnamebox}[1]{%
1866   \expandafter\unvbox\csname #1\endcsname}
1867 \providecommand*\namebox}[1]{%
1868   \csname #1\endcsname}
1869
1870 %

```

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

```

1871 \providecommand*{\newnamecount}[1]{%
1872   \expandafter\newcount\csname #1\endcsname}
1873 \providecommand*{\usenamecount}[1]{%
1874   \csname #1\endcsname}
1875
1876 %

```

## XV 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
1877 \newif\ifl@dusedbabel
\l@dusedbabeltrue
1878 %

```

`\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 `\`.

```

1879 \patchcmd{\bbl@set@language}%
1880   {\select@language{\language}}%
1881   {\edef\language{#1}\select@language{\language}}%
1882   {}%
1883   {}%
1884
1885 %

```

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

```



```

1886 \newcommand*{\l@duselanguage}[1]{}
1887 \gdef\theledlanguageL{}
1888 \gdef\theledlanguageR{}
1889
1890 %

```

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

```

1891 \AtBeginDocument{%
1892   \ifundefined{xpg@main@language}{%
1893     \ifundefined{bbl@main@language}{%
1894       %

```

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

```

1895     \l@dusedbabelfalse
1896   }{%
1897   %

```

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.

```

1898     \l@dusedbabeltrue
1899     \let\l@doldselectlanguage\selectlanguage
1900     \let\l@doldbbl@set@language\bbl@set@language
1901     \renewcommand{\selectlanguage}[1]{%
1902       \l@doldselectlanguage{#1}%
1903       \ifledRcol \gdef\theledlanguageR{#1}%
1904       \else      \gdef\theledlanguageL{#1}%
1905       \fi}
1906   %

```

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

```

1907     \renewcommand*{\l@duselanguage}[1]{%
1908       \expandafter\l@doldselectlanguage\expandafter{#1}}
1909   %

```

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

```

1910     \gdef\theledlanguageL{\bbl@main@language}%
1911     \gdef\theledlanguageR{\bbl@main@language}%
1912   }%
1913 }
1914 %

```

If use polyglossia

```

1915 { \let\old@otherlanguage\otherlanguage%
1916   \renewcommand{\otherlanguage}[2] [] {%
1917     \selectlanguage{#1}{#2}%
1918     \ifledRcol \gdef\theledlanguageR{#2}%
1919     \else      \gdef\theledlanguageL{#2}%

```

```

1920 \fi}%
1921 \renewcommand{\l@duselanguage}[1]{%
1922 \csuse{no\language @numbers}\select@language{#1}%
1923 }%
1924 \gdef\theledlanguageL{\xpg@main@language}%
1925 \gdef\theledlanguageR{\xpg@main@language}%
1926 %
    That is it.
1927 }}
1928 %

```

## XVI 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.

```

1929 \newcount\l@dc@maxchunks
1930 \newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}
1931 \maxchunks{5120}
1932
1933 %

```

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

`\l@dnumpstartsR`

```

1934 \newcount\l@dnumpstartsR
1935
1936 %

```

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

`\l@pscR`

```

1937 \newcount\l@dpscL
1938 \newcount\l@dpscR
1939
1940 %

```

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

```

1941 \newcommand*\l@dsetuprawboxes{%
1942 \l@dttempcntb=\l@dc@maxchunks
1943 \loop\ifnum\l@dttempcntb>\z@
1944 \newnamebox{\l@dLcolrawbox\the\l@dttempcntb}

```

```

1945 \newnamebox{\l@dRcolrawbox\the\@l@tempcntb}
1946 \advance\@l@tempcntb \m@ne
1947 \repeat}
1948
1949 %

```

`\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.

```

1950 \newcommand*{\l@dsetupmaxlinecounts}{%
1951 \l@tempcntb=\l@dc@maxchunks
1952 \loop\ifnum\@l@tempcntb>\z@
1953 \newnamecount{\l@dmaxlinesinpar\the\@l@tempcntb}
1954 \advance\@l@tempcntb \m@ne
1955 \repeat}
1956 \newcommand*{\l@dzeromaxlinecounts}{%
1957 \begingroup
1958 \l@tempcntb=\l@dc@maxchunks
1959 \loop\ifnum\@l@tempcntb>\z@
1960 \global\usernamecount{\l@dmaxlinesinpar\the\@l@tempcntb}=\z@
1961 \advance\@l@tempcntb \m@ne
1962 \repeat
1963 \endgroup}
1964
1965 %

```

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

```

1966 \AtBeginDocument{%
1967 \l@dsetuprawboxes
1968 \l@dsetupmaxlinecounts
1969 \l@dzeromaxlinecounts
1970 \l@dnumpstartsL=\z@
1971 \l@dnumpstartsR=\z@
1972 \l@dpscL=\z@
1973 \l@dpscR=\z@}
1974
1975 %

```

## XVII Checking text to be processed

```

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

```

```

1978 \@startsfalse
1979 \ifnum\l@dnumpstartsL>\l@dpscL
1980 \@startstrue
1981 \else
1982 \ifnum\l@dnumpstartsR>\l@dpscR
1983 \@startstrue
1984 \fi
1985 \fi
1986 }
1987
1988 %

```

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

```

\checkraw@text
1989 \newif\ifaraw@text
1990 \newcommand*{\checkraw@text}{%
1991 \araw@textfalse
1992 \ifvbox\namebox{1@dLcolrawbox\the\l@dpscL}
1993 \araw@texttrue
1994 \else
1995 \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}
1996 \araw@texttrue
1997 \fi
1998 \fi
1999 }
2000
2001 %

```

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

```

2002 \newcommand*{\@writelinesinparL}{%
2003 \edef\next{%
2004 \write\linenum@out{\string\@pend[\the\@donereallinesL]}}%
2005 \next
2006 \global\@donereallinesL \z@}
2007 \newcommand*{\@writelinesinparR}{%
2008 \edef\next{%
2009 \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}%
2010 \next
2011 \global\@donereallinesR \z@}
2012
2013 %

```

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

```

\@writepageofparR
2014 \newcommand*{\@writepageofparL}[0]{%
2015 \ifnum\@donereallinesL=\z@%

```

```

2016 \edef\next{%
2017 \write\linenum@out{\string\@pstart{\the\l@dpscl}{\the\c@page}{\the\
numpagelinesL}}%
2018 }%
2019 \next%
2020 \fi%
2021 }%
2022 \newcommand*{\@writepageofparR}[0]{%
2023 \ifnum\@donereallinesR=\z@%
2024 \edef\next{%
2025 \write\linenum@outR{\string\@pstartR{\the\l@dpscR}{\the\c@page}{\the\
numpagelinesR}}%
2026 }%
2027 \next%
2028 \fi%
2029 }%
2030 %

```

## XVIII Parallel columns

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

```

\@eledsectionR
2031 \newsavebox{\@eledsectionL}%
2032 \newsavebox{\@eledsectionR}%
2033 %

```

`\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.

```

2034 \newcommand*{\Columns}{%
2035 \ifl@dpairing%
2036 \led@err@Columns@InsideEnv%
2037 \fi%
2038 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
2039 \led@err@Columns@WithoutEnv%
2040 \else%
2041 \global\l@dprintingcolumnstrue%
2042 \eledsection@correcting@skip=-\baselineskip% Correction for sections'
titles
2043 \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else
2044 \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
2045 \fi
2046 %

```

Start a group and zero counters, etc.

```

2047 \begingroup
2048 \l@dzeropenalties
2049 \endgraf\global\num@lines=\prevgraf
2050 \global\num@linesR=\prevgraf

```

```

2051 \global\par@line=\z@
2052 \global\par@lineR=\z@
2053 \global\l@dpscL=\z@
2054 \global\l@dpscR=\z@
2055 \get@familiarfootnote@number%
2056 %

```

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

```

2057 \check@pstarts
2058 \loop\if@pstarts
2059 \global\pstartnumtrue
2060 \global\pstartnumRtrue
2061 %

```

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.

```

2062 \global\advance\l@dpscL \@ne
2063 \global\advance\l@dpscR \@ne
2064 \restore@pstartL@pc%
2065 \restore@pstartR@pc%
2066 %

```

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

```

2067 \Columns@print@before@pstart%
2068 %

```

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

```

2069 \checkraw@text
2070 { \loop\ifaraw@text
2071 %

```

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

```

2072 \l@duselanguage{\theledlanguageL}%
2073 \do@lineL
2074 \xifinlist{\the\l@dpscL}{\eled@sections@@}
2075 {%
2076 \ifdefstring{\@eledsectmark}{L}%
2077 {\csuse{eled@sectmark@the\l@dpscL}%
2078 }{}}%
2079 \global\csundef{eled@sectmark@the\l@dpscL}%
2080 \savebox{\@eledsectionL}{\parbox[t]{}[t]{\Lcolwidth}{\vbox
{} \print@eledsectionL}}%\vbox{}-> prevent alignment troubles with RTL
language
2081 }%
2082 {}%
2083 \l@duselanguage{\theledlanguageR}%

```

```

2084 \do@lineR
2085 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}
2086 {%
2087 \ifdefstring{\@eledsectmark}{R}%
2088 {\csuse{eled@sectmark@\the\l@dpscR R}%
2089 }{}%
2090 \global\csundef{eled@sectmark@\the\l@dpscR R}%
2091 \savebox{\@eledsectionR}{\parbox[t]{}[t]{\Rcolwidth}{\vbox
{} \print@eledsectionR}}}%\vbox{}-> prevent alignment troubles with RTL
language
2092 {}%
2093 \hb@xt@ \hsize{%
2094 \ifdefstring{\columns@position}{L}{\hfill }%
2095 \unhbox\l@leftbox%
2096 \ifhbox\@eledsectionL%
2097 \usebox{\@eledsectionL}%
2098 \fi%
2099 \print@columnseparator%
2100 \unhbox\l@drighthbox%
2101 \ifhbox\@eledsectionR%
2102 \usebox{\@eledsectionR}%
2103 \fi%
2104 \ifdefstring{\columns@position}{R}{\hfill}%
2105 }%
2106 \checkraw@text
2107 \checkverseL
2108 \checkverseR
2109 \checkpb@columns
2110 \repeat}
2111 %

```

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.

```

2112 \@writelinesinparL
2113 \@writelinesinparR
2114 \check@pstarts
2115 \ifbypstart@%
2116 \write\linenum@out{\string\@set[1]}
2117 \resetprevline@
2118 \fi
2119 \ifbypstart@R
2120 \write\linenum@outR{\string\@set[1]}
2121 \resetprevline@
2122 \fi
2123 \Columns@print@after@pend%
2124 \repeat
2125 %

```

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.

```

2126 \flush@notes
2127 \flush@notesR
2128 \endgroup
2129 %

2130 \global\l@dpscL=\z@
2131 \global\l@dpscR=\z@
2132 \global\l@dnpstartsL=\z@
2133 \global\l@dnpstartsR=\z@
2134 \global\l@dprintingcolumnsfalse%
2135 \ignorespaces
2136 \global\instanzaLfalse%
2137 \global\instanzaRfalse%
2138 \fi}
2139
2140 %

```

**\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).

```

2141 \def\print@columnseparator{%
2142 \ifdim\beforecolumnseparator<0pt%
2143 \hfill%
2144 \else%
2145 \hspace{\beforecolumnseparator}%
2146 \fi%
2147 \columnseparator%
2148 \ifdim\aftercolumnseparator<0pt%
2149 \hfill%
2150 \else%
2151 \hspace{\beforecolumnseparator}%
2152 \fi%
2153 }%
2154 %

```

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

```

2155
2156 \newcommand{\checkpb@columns}{%
2157 \newif\if@pb
2158 \newif\if@nopb
2159 \IfStrEq{\led@pb@setting}{before}{
2160 \numdef{\next@absline}{\the\absline@num+1}%
2161 \numdef{\next@abslineR}{\the\absline@numR+1}%
2162 \xifinlists{\next@absline}{\l@prev@pb}{\@pbtrue}{}%

```



```

2163 \xifinlistcs{\next@abslineR}{l@prev@pbR}{\@pbtrue}{%
2164 \xifinlistcs{\next@absline}{l@prev@nopb}{\@nopbtrue}{}%
2165 \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\@nopbtrue}{%
2166 }{}
2167 \IfStrEq{\led@pb@setting}{after}{
2168 \xifinlistcs{\the\absline@num}{l@prev@pb}{\@pbtrue}{}%
2169 \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\@pbtrue}{%
2170 \xifinlistcs{\the\absline@num}{l@prev@nopb}{\@nopbtrue}{}%
2171 \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\@nopbtrue}{%
2172 }{}
2173 \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
2174 \if@pb\pagebreak[4]\fi
2175 }
2176 %

```

**\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).

```

2177 \newcommand*{\columnseparator}{%
2178 \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}}
2179 \newdimen\columnrulewidth
2180 \columnrulewidth=\z@
2181
2182 %

```

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

```

2183 \newcommand*{\columnspan}[1]{%
2184 \xdef\columns@position{#1}%
2185 }%
2186 \xdef\columns@position{R}%
2187 %

```

**\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`.

```

2188 \newlength{\beforecolumnseparator}%
2189 \setlength{\beforecolumnseparator}{-2pt}%
2190
2191 \newlength{\aftercolumnseparator}%
2192 \setlength{\aftercolumnseparator}{-2pt}%
2193
2194 %

```

**setwidthliketwocolumns@L** `\setwidth...` macros are called in `\beginnumbing` in a **non-parallel** typesetting context, to fix the width of the lines to be vertically aligned with parallel columns. They

```

\setwidthliketwocolumns@L
\setwidthliketwocolumns@C
\setwidthliketwocolumns@R
\setwidthliketwocolumns@L
\setwidthliketwocolumns@C
\setwidthliketwocolumns@R

```

are also called at the beginning of a note's group, if some options are enabled. The `\setposition...` 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.

```
2195 \newcommand{\setwidthliketwocolumns@L}{%
2196 % Temporary dimension, initially equal to the standard hsize, i.e. text
    width
2197 %   \begin{macrocode}
2198   \newdimen\temp%
2199   \temp=\hsize%
2200 %
```

Hsize : Left + Right width

```
2201   \hsize=\Lcolwidth%
2202   \advance\hsize\Rcolwidth%
2203 %
```

Now, calculating the remaining space

```
2204   \advance\temp-\hsize%
2205 %
```

And multiply the hsize by 2/3 of this space

```
2206   \multiply\temp by 2%
2207   \divide\temp by 3%
2208   \advance\hsize\temp%
2209 }%
2210
2211 \newcommand{\setpositionliketwocolumns@L}{%
2212   \renewcommand{\ledrlfill}{\hfill}%
2213 }%
2214
2215 \newcommand{\setnotespositionliketwocolumns@L}{%
2216 }%
2217
2218
2219 %
```

```
2220 \newcommand{\setwidthliketwocolumns@C}{%
2221 % Temporary dimension, initially equal to the standard hsize, i.e. text
    width
2222 %
```

```
2223   \newdimen\temp%
2224   \temp=\hsize%
2225 % Hsize : Left + Right width
2226 %
```

```

2227 \hsize=\Lcolwidth%
2228 \advance\hsize\Rcolwidth%
2229 % Now, calculating the remaining space
2230 %

```

```

2231 \advance\temp-\hsize%
2232 %

```

And multiply the hsize by 1/2 of this space

```

2233 \divide\temp by 2%
2234 \advance\hsize\temp%
2235 }%
2236
2237 \newcommand{\setpositionliketwocolumns@C}{%
2238 \doinsidelinehook{\hfill}%
2239 \renewcommand{\ledrlfill}{\hfill}%
2240 }%
2241
2242 \newcommand{\setnotespositionliketwocolumns@C}{%
2243 \newdimen\temp%
2244 \newdimen\tempa%
2245 \temp=\hsize%
2246 \tempa=\Lcolwidth%
2247 \advance\tempa\Rcolwidth%
2248 \advance\temp-\tempa%
2249 \divide\temp by 2%
2250 \leftskip=\temp%
2251 \rightskip=-\temp%
2252 }%
2253
2254 \newcommand{\setwidthliketwocolumns@R}{%
2255 %

```

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

```

2256 \newdimen\temp%
2257 \temp=\hsize%
2258 %

```

Hsize : Left + Right width

```

2259 \hsize=\Lcolwidth%
2260 \advance\hsize\Rcolwidth%
2261 %

```

Now, calculating the remaining space

```

2262 \advance\temp-\hsize%
2263 %

```

And multiply the hsize by 2/3 of this space

```

2264 \multiply\temp by 2%
2265 \divide\temp by 3%
2266 \advance\hsize\temp%
2267 }%
2268
2269 \newcommand{\setpositionliketwocolumns@R}{%
2270 \doinsidelinehook{\hfill}%
2271 }%
2272
2273 \newcommand{\setnotespositionliketwocolumns@R}{%
2274 \newdimen\temp%
2275 \newdimen\tempa%
2276 \temp=\hsize%
2277 \tempa=\Lcolwidth%
2278 \advance\tempa\Rcolwidth%
2279 \advance\temp-\tempa%
2280 \divide\temp by 2%
2281 \leftskip=\temp%
2282 \rightskip=-\temp%
2283 }%
2284
2285 %

```

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

`\Columns@print@after@pend`

```

2286 \newcommand{\Columns@print@before@pstart}{%
2287 \ifboolexpr{%
2288 test{\ifcsstring{before@pstartL@the\l@dpscl}{\at@every@pstart}}%
2289 and test {\ifcsstring{before@pstartR@the\l@dpscl}{\at@every@pstart}}%
2290 and test {\ifdefempty{\at@every@pstart}}}%
2291 {}%
2292 {%
2293 \hb@xt@ \hsize{%
2294 \ifdefstring{\columns@position}{L}{\hfill }%
2295 \par\parbox[t] [] [t]{\Lcolwidth}{%
2296 \csuse{before@pstartL@the\l@dpscl}%
2297 }%
2298 \print@columnseparator%
2299 \parbox[t] [] [t]{\Rcolwidth}{%
2300 \set@sectcountR%
2301 \csuse{before@pstartR@the\l@dpscl}%
2302 }%
2303 \ifdefstring{\columns@position}{R}{\hfill}%
2304 }%
2305 }%
2306 \global\csundef{before@pstartL@the\l@dpscl}%
2307 \global\csundef{before@pstartR@the\l@dpscl}%
2308 }%

```

```

2309 \newcommand{\Columns@print@after@pend}{%
2310   \ifboolexpr{%
2311     test{\ifcsstring{after@pendL@the\l@dpscL}{\at@every@pend}}%
2312     and test {\ifcsstring{after@pendR@the\l@dpscR}{\at@every@pend}}%
2313     and test {\ifdefempty{\at@every@pend}}}%
2314     {}%
2315     {%
2316       \hb@xt@ \hsize{%
2317         \ifdefstring{\columns@position}{L}{\hfill }%
2318         \parbox[t] [] [t]{\Lcolwidth}{%
2319           \csuse{after@pendL@the\l@dpscL}%
2320         }%
2321         \print@columnseparator%
2322         \parbox[t] [] [t]{\Rcolwidth}{%
2323           \set@sectcountR%
2324           \csuse{after@pendR@the\l@dpscR}%
2325         }%
2326         \ifdefstring{\columns@position}{R}{\hfill}%
2327       }%
2328     }%
2329   \global\csundef{after@pendL@the\l@dpscL}%
2330   \global\csundef{after@pendR@the\l@dpscR}%
2331 }%
2332 %

```

## XIX Parallel pages

This is considerably more complicated than parallel columns.

### XIX.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`

```

2333 \newcount\numpagelinesL
2334 \newcount\numpagelinesR
2335 \newcount\l@dminpagelines
2336
2337 %

```

### XIX.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.

```

2338 \newcommandx*{\Pages}[1][1,usedefault]{%
2339   \ifl@dpairing%

```

```

2340 \led@err@Pages@InsideEnv%
2341 \fi%
2342 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
2343 \led@err@Pages@WithoutEnv%
2344 \else%
2345 \ifstrequal{#1}{mainmatter}{\Pages@mainmattertrue}{\Pages@mainmatterfalse
}%
2346 \eledsection@correcting@skip=-2\baselineskip% line correcting for section
titles.
2347 \parledgroup@notespacing@set@correction%
2348 \typeout{}%
2349 \typeout{***** PAGES *****}%
2350 \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else%
2351 \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
2352 \fi%
2353 %

```

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

```

2354 \cleartol@devenpage%
2355 \global\l@dprintingpagetrue%
2356 \begingroup%
2357 %

```

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

```

2358 \setlength{\Lcolwidth}{\textwidth}%
2359 \setlength{\Rcolwidth}{\textwidth}%
2360 %

```

```

2361 \l@dzeropenalties%
2362 \endgraf\global\num@lines=\prevgraf%
2363 \global\num@linesR=\prevgraf%
2364 \global\par@line=\z@%
2365 \global\par@lineR=\z@%
2366 \global\l@dpscL=\z@%
2367 \global\l@dpscR=\z@%
2368 \writtenlinesLfalse%
2369 \writtenlinesRfalse%
2370 \get@familiarfootnote@number%
2371 %

```

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

```

2372 \let\print@Xnotes\print@Xnotes@forpages%
2373 \let\print@notesX\print@notesX@forpages%
2374 %

```

Check if there are chunks to be processed.

```

2375 \check@pstarts%

```

```
2376 \loop\if@pstarts%
2377 %
```

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

```
2378 \global\advance\l@dpscL \@ne%
2379 \global\advance\l@dpscR \@ne%
2380 %
```

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

```
2381 \getlinesfromparlistL%
2382 \getlinesfromparlistR%
2383 \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}%
2384 {\usernamecount{\l@dmaxlinesinpar\the\l@dpscL}}%
2385 \check@pstarts%
2386 \repeat%
2387 %
```

Zero the counts again, ready for the next bit.

```
2388 \global\l@dpscL=\z@%
2389 \global\l@dpscR=\z@%
2390 %
```

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

```
2391 \getlinesfrompagelistL%
2392 \getlinesfrompagelistR%
2393 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
2394 {\l@dminpagelines}%
2395 %
```

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.

```
2396 \check@pstarts%
2397 \if@pstarts%
2398 %
```

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

```
2399 \global\advance\l@dpscL \@ne%
2400 \global\advance\l@dpscR \@ne%
2401 \restore@pstartL@pc%
2402 \restore@pstartR@pc%
2403 %
```

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

```

2404 \global\@donereallinesL=\z@%
2405 \global\@donetotallinesL=\z@%
2406 \global\@donereallinesR=\z@%
2407 \global\@donetotallinesR=\z@%
2408 %

```

Start a loop over the boxes (chunks).

```

2409 \checkraw@text%
2410 %
2411 % \begingroup
2412 { \loop\ifaraw@text%
2413 %

```

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

```

2414 \checkpageL%
2415 \l@duselanguage{\theledlanguageL}%
2416 { \loop\ifl@dsamepage%
2417 %

```

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

```

2418 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}%
2419 \csuse{before@pstartL@the\l@dpscl}%
2420 \global\csundef{before@pstartL@the\l@dpscl}%
2421 \do@lineL%
2422 \xifinlist{\the\l@dpscl}{\eled@sections@@}
2423 {\print@eledsectionL}%
2424 {}%
2425 \advance\numpagelinesL \@ne%
2426 %

```

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`

```

2427 \ifshiftedpstarts%
2428 \ifdim\ht\l@dleftbox>0pt%
2429 \parledgroup@correction@notespacing{L}%
2430 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2431 \else%
2432 \unless\ifadvancedshiftedpstarts%
2433 \dimen0=\pagetotal%
2434 \advance\dimen0 by \baselineskip%
2435 \global\pagetotal=\dimen0%
2436 \else%

```



```

2437         \ifnomaxlines%
2438         \numdef{\@tmp}{\the\l@dpscL+1}%
2439         \ifcsdef{minpage@pstart@\@tmp}{%
2440         \ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}}%
2441         {\dimen0=\pagetotal%
2442         \advance\dimen0 by \baselineskip%
2443         \global\pagetotal=\dimen0%
2444         }%
2445         {}%
2446         }{}%
2447         \fi%
2448         \fi%
2449         \fi%
2450     \else%
2451     \parledgroup@correction@notespacing{L}%
2452     \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2453     \fi%
2454 %

```

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

```

2455     \get@nextboxL%
2456     \global\l@dskipversenumberfalse%
2457     \ifprint@last@after@pendL%
2458     \csuse{after@pendL@\the\l@dpscL}%
2459     \global\csundef{after@pendL@\the\l@dpscL}%
2460     \fi%
2461     \checkpageL%
2462     \checkverseL%
2463     \checkpbL%
2464     \repeat%
2465 %

```

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.

```

2466     \ifl@dpagefull%
2467     \@writelinesonpageL{\the\numpagelinesL}%
2468     \else%
2469     \@writelinesonpageL{1000}%
2470     \fi%
2471 %

```

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.

```

2472 \numpagelinesL \z%
2473 \parledgroup@correction@notespacing@init%
2474 \clearl@dleftpage }%
2475 %

```

Now do the same for the right text.

```

2476 \checkpageR%
2477 \l@duselanguage{\theledlanguageR}%
2478 {
2479 \loop\ifl@dsamepage%
2480 \set@sectcountR%
2481 \ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{}%
2482 \csuse{before@pstartR@the\l@dpscR}%
2483 \global\csundef{before@pstartR@the\l@dpscR}%
2484 \do@lineR%
2485 \xifinlist{the\l@dpscR}{\eled@sectionsR@@}%
2486 {\print@eledsectionR}%
2487 {}%
2488 \advance\numpagelinesR \@ne%
2489 \ifshiftedpstarts%
2490 \ifdim\ht\l@drightbox>0pt%
2491 \parledgroup@correction@notespacing{R}%
2492 \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2493 \else%
2494 \unless\ifadvancedshiftedpstarts%
2495 \dimen0=\pagetotal%
2496 \advance\dimen0 by \baselineskip%
2497 \global\pagetotal=\dimen0%
2498 \else%
2499 \ifnomaxlines%
2500 \numdef{\@tmp}{\the\l@dpscR+1}%
2501 \ifcsdef{minpage@pstart@\@tmp}{%
minpage@pstart@\@tmp}}%
2502 {\dimen0=\pagetotal%
2503 \advance\dimen0 by \baselineskip%
2504 \global\pagetotal=\dimen0%
2505 }%
2506 {}%
2507 }{}%
2508 \fi%
2509 \fi%
2510 \fi%
2511 \else%
2512 \parledgroup@correction@notespacing{R}%
2513 \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2514 \fi%
2515 \get@nextboxR%
2516 \global\l@dskipversenumberRfalse%
2517 \ifprint@last@after@pendR%
2518 \csuse{after@pendR@the\l@dpscR}%

```

```

2519         \global\csundef{after@pendR@the\l@dpscR}%
2520         \fi%
2521         \checkpageR%
2522         \checkverseR%
2523         \checkpbR%
2524         \repeat%
2525         \ifl@pagefull%
2526         \@writelinesonpageR{\the\numpagelinesR}%
2527         \else%
2528         \@writelinesonpageR{1000}%
2529         \fi%
2530         \numpagelinesR=\z@%
2531         \parledgroup@correction@notespacing@init%
2532 %

```

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

```

2533         \clearl@drightpage}%
2534 %

```

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.

```

2535         \checkraw@text%
2536         \ifaraw@text%
2537         \getlinesfrompagelistL%
2538         \getlinesfrompagelistR%
2539         \l@dcalcm@minoftwo{\cs@linesonpageL}{\cs@linesonpageR}%
2540         {\l@dminpagelines}%
2541         \fi%
2542         \repeat}%
2543 %

```

We have now output the text from all the chunks.

```

2544         \fi%
2545 %

```

Make sure that there are no inserts hanging around.

```

2546         \flush@notes%
2547         \flush@notesR%
2548         \endgroup%
2549 %

```

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

```

2550         \global\l@dpscL=\z@%
2551         \global\l@dpscR=\z@%
2552         \global\l@dnumpststartsL=\z@%
2553         \global\l@dnumpststartsR=\z@%
2554         \global\instanzaLfalse%
2555         \global\instanzaRfalse%

```

```

2556 \global\l@dprintingpagesfalse%
2557 \finish@Pages@notes%Needed to prevent final notes overlap line number
2558 \ignorespaces\fi}
2559
2560
2561 %

```

### XIX.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.

```

2562 \newcommand{\finish@Pages@notes}{%
2563 \def\do##1{%
2564 %

```

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

```

2565 \ifnocritical@%
2566 \global\newnamebox{##1footins}
2567 \fi
2568 \ifnofamiliar@%
2569 \global\newnamebox{footins##1}
2570 \fi
2571 %

```

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

```

2572 \ifvoid\csuse{##1footins}%
2573 \ifvoid\csuse{footins##1}\else%
2574 \newpage\null%
2575 \listbreak%
2576 \fi%
2577 \else%
2578 \newpage\null%
2579 \listbreak%
2580 \fi%
2581 }%
2582 \dolistloop{@series}%
2583 }%
2584 %

```

### XIX.4 Struts

**\ledstrutL** Struts inserted into leftand right text lines.

```

\ledstrutR
2585 \newcommand*{\ledstrutL}{}
2586 \newcommand*{\ledstrutR}{}
2587
2588 %

```

## XIX.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.

```

2589 \providecommand{\cleartoevenpage}[1][\@empty]{%
2590   \clearpage
2591   \ifodd\c@page\hbox{#1}\clearpage\fi}
2592
2593 \newcommand*{\cleartol@devenpage}{%
2594   \ifdim\pagetotal<\topskip% on an empty page
2595   \else
2596     \clearpage
2597     \Pages@mainmatter%
2598   \fi
2599   \ifodd\c@page%
2600     \ifprevpgnotnumbered%
2601       \addtocounter{par@page}{-1}%
2602       \ifdef\prevpgstyle{\thispagestyle{\prevpgstyle}}}%
2603     \fi%
2604     \hbox{}\clearpage%
2605   \fi%
2606 }%
2607 %

```

`\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`.

```

2608 \newcommand*{\clearl@dleftpage}{%
2609   \ifdim\pagetotal=0pt\hbox{}\fi%
2610   \newpage%
2611   \ifodd\c@page\else
2612     \led@err@LeftOnRightPage
2613     \hbox{}\fi%
2614     \cleardoublepage
2615   \fi}
2616
2617 \newcommand*{\clearl@drightpage}{%
2618   \ifdim\pagetotal=0pt\hbox{}\fi%
2619   \newpage%
2620   \ifodd\c@page
2621     \led@err@RightOnLeftPage
2622     \hbox{}\fi%
2623     \cleartoevenpage
2624   \fi}

```

```
2625
2626 %
```

## XIX.6 Lines managing

`\getlinesfromparlistL` `\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 `\getlinesfromparlistR` for `\getlinesfromparlistR`.

```
\@cs@linesinparR
2627 \newcommand*\getlinesfromparlistL{%
2628   \ifx\linesinpar@listL\empty
2629     \gdef\@cs@linesinparL{0}%
2630   \else
2631     \gl@p\linesinpar@listL\to\@cs@linesinparL
2632   \fi}
2633 \newcommand*\getlinesfromparlistR{%
2634   \ifx\linesinpar@listR\empty
2635     \gdef\@cs@linesinparR{0}%
2636   \else
2637     \gl@p\linesinpar@listR\to\@cs@linesinparR
2638   \fi}
2639
2640 %
```

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

```
\@cs@linesonpageR
2641 \newcommand*\getlinesfrompagelistL{%
2642   \ifx\linesonpage@listL\empty
2643     \gdef\@cs@linesonpageL{1000}%
2644   \else
2645     \gl@p\linesonpage@listL\to\@cs@linesonpageL
2646   \fi}
2647 \newcommand*\getlinesfrompagelistR{%
2648   \ifx\linesonpage@listR\empty
2649     \gdef\@cs@linesonpageR{1000}%
2650   \else
2651     \gl@p\linesonpage@listR\to\@cs@linesonpageR
2652   \fi}
2653
2654 %
```

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

```
2655 \newcommand*\@writelinesonpageL}[1]{%
2656   \edef\next{\write\linenum@out{\string\@lopL{#1}}}%
2657   \next}
```

```

2658 \newcommand*{\@writelinesonpageR}[1]{%
2659   \edef\next{\write\linenum@outR{\string\@lopR{#1}}}%
2660   \next}
2661
2662 %

```

`\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⟩`.

```

2663 \newcommand*{\l@dcalc@maxoftwo}[3]{%
2664   \ifnum #2>#1\relax
2665     #3=#2\relax
2666   \else
2667     #3=#1\relax
2668   \fi}
2669 \newcommand*{\l@dcalc@minoftwo}[3]{%
2670   \ifnum #2<#1\relax
2671     #3=#2\relax
2672   \else
2673     #3=#1\relax
2674   \fi}
2675
2676 %

```

## XIX.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
2677 \newif\ifl@dsamepage
2678 \l@dsamepagetrue
2679 \newif\ifl@dpagfull
2680
2681 \newcommand*{\checkpageL}{%
2682   \l@dpagfulltrue
2683   \l@dsamepagetrue
2684   \check@goal
2685   \ifdim\pagetotal<\ledthegoal
2686     \ifnum\umpagelinesL<\l@minpagelines
2687     \else
2688       \ifnomaxlines%
2689       \else%
2690       \l@dsamepagefalse%

```

```

2691         \l@dpagfullfalse%
2692         \fi%
2693     \fi
2694 \else
2695     \l@dsamepagefalse
2696     \l@dpagfulltrue
2697 \fi%
2698 \ifprint@last@after@pendL%
2699     \l@dpagfullfalse%
2700     \l@dsamepagefalse%
2701     \print@last@after@pendLfalse%
2702 \fi%
2703 }%
2704
2705 \newcommand*{\checkpageR}{%
2706     \l@dpagfulltrue
2707     \l@dsamepagetrue
2708     \check@goal
2709     \ifdim\pagetotal<\ledthegoal
2710         \ifnum\numpagelinesR<\l@dminpagelines
2711             \else
2712                 \ifnomaxlines%
2713                 \else%
2714                     \l@dsamepagefalse%
2715                     \l@dpagfullfalse%
2716                 \fi%
2717             \fi
2718         \else
2719             \l@dsamepagefalse
2720             \l@dpagfulltrue
2721         \fi%
2722         \ifprint@last@after@pendR%
2723             \l@dpagfullfalse%
2724             \l@dsamepagefalse%
2725             \print@last@after@pendRfalse%
2726         \fi%
2727     }%
2728
2729 %

```

`\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`.

```

2730 \newcommand{\checkpbL}{
2731     \IfStrEq{\led@pb@setting}{after}{
2732         \xifinlistcs{\the\absline@num}{l@prev@pb}{\l@dpagfulltrue\
l@dsamepagefalse}{
2733         \xifinlistcs{\the\absline@num}{l@prev@nopb}{\l@dpagfullfalse\
l@dsamepagetrue}{
2734     }{}

```



```

2735 \IfStrEq{\led@pb@setting}{before}{
2736   \numdef{\next@absline}{\the\absline@num+1}
2737   \xifinlistcs{\next@absline}{l@prev@pb}{\l@dpagetrue\
l@dsamepagefalse}{
2738     \xifinlistcs{\next@absline}{l@prev@nopb}{\l@dpagetrue\
l@dsamepagetrue}{
2739   }{}
2740 }
2741
2742 \newcommand{\checkpbR}{
2743   \IfStrEq{\led@pb@setting}{after}{
2744     \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\l@dpagetrue\
l@dsamepagefalse}{
2745       \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\l@dpagetrue\
l@dsamepagetrue}{
2746     }{}
2747   }
2748   \IfStrEq{\led@pb@setting}{before}{
2749     \numdef{\next@abslineR}{\the\absline@numR+1}
2750     \xifinlistcs{\next@abslineR}{l@prev@pbR}{\l@dpagetrue\
l@dsamepagefalse}{
2751       \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\l@dpagetrue\
l@dsamepagetrue}{
2752     }{}
2753   }
2754 }
2755 %

```

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

```

2754 \newcommand{\checkverseL}{
2755   \ifinstanzaL
2756     \iflednopbinverse
2757       \ifinserthangingsymbol
2758         \numdef{\prev@abslineverse}{\the\absline@num-1}
2759         \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{}
2760         \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesL<3\ledpbnum{\
prev@abslineverse}\fi}{
2761       \fi
2762     }
2763   \fi
2764 }
2765 \newcommand{\checkverseR}{
2766   \ifinstanzaR
2767     \iflednopbinverse
2768       \ifinserthangingsymbolR
2769         \numdef{\prev@abslineverse}{\the\absline@numR-1}
2770         \IfStrEq{\led@pb@setting}{after}{\lednopbnumR{\prev@abslineverse}}{}
2771         \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesR<3\ledpbnumR{\
prev@abslineverse}\fi}{
2772       \fi
2773     }
2774   \fi
2775 }

```

```

2773 \fi
2774 \fi
2775 }
2776 %

```

`\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
2777 \newdimen\ledthegoal
2778 \ifshiftedpstarts
2779     \newcommand*{\@goalfraction}{0.95}
2780 \else
2781     \newcommand*{\@goalfraction}{0.9}
2782 \fi
2783
2784 \newcommand*{\check@goal}{%
2785     \ledthegoal=\@goalfraction\pagegoal}
2786 \newcommand{\setgoalfraction}[1]{%
2787     \xdef\@goalfraction{#1}%
2788 }
2789 %

```

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

```

\ifwrittenlinesL
2790 \newif\ifwrittenlinesL
2791 \newif\ifwrittenlinesR
2792
2793 %

```

## XIX.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.

```

2794 \newif\if@getnextbox%
2795 %

```

`\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.

```

2796 \newcommand*{\get@nextboxL}{%
2797     \ifvbox\namebox{1@dLcolrawbox\the\1@dpscl}% box is not empty
2798 %

```

The current box is not empty; do nothing.

```
2799 \else%                                box is empty
2800 %
```

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

```
2801 \@getnextboxtrue%Should be local, but be cautious
2802 %
```

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

```
2803 \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscl}>\@donetotallinesL
2804 \parledgroup@notes@endL%
2805 \unless\ifnosyncpstarts%
2806 \getnextboxfalse%
2807 %
```

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.

```
2808 \ifnomaxlines%
2809 \ifdim\pagetotal<\ledthegoal%
2810 \numdef{\@tmp}{\l@dpscl+1}%
2811 \ifcsdef{afterlines@pstart@\@tmp R}{%
2812 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp R}}
2813 %
2814 {}%
2815 {\ifcsdef{minpage@pstart@\@tmp}%
2816 {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2817 {\ifnum\numpagelinesL=\l@dminpagelines%
2818 \getnextboxtrue%
2819 \fi%
2820 }%
2821 {\@getnextboxtrue}}%
2822 {\@getnextboxtrue}}%
2823 }%
2824 {}%
2825 \fi%
2826 \fi%
2827 \fi%
2828 \else%
2829 \ifnomaxlines%
2830 \numdef{\@tmp}{\the\l@dpscl+1}%
2831 \ifcsdef{minpage@pstart@\@tmp}%
2832 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2833 {\ifdimgreater{\pagetotal}{\ledthegoal}%
2834 {\@getnextboxtrue}%
2835 {\@getnextboxfalse}%
2836 }%
2837 }
```

```

2837         {\@getnextboxtrue}%
2838         }{}%
2839         \fi%
2840     \fi%
2841 %

```

Sufficient lines have been output.

```

2842     \if@getnextbox%
2843         \ifnum\usernamecount{1@dmaxlinesinpar\the\1@dpscL}=\@donetotallinesL
2844             \parledgroup@notes@endL
2845         \fi
2846         \ifwrittenlinesL\else
2847 %

```

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

```

2848         \@writelinesinparL
2849         \writtenlinesLtrue
2850     \fi
2851     \ifnum\1@dnumstartsL>\1@dpscL
2852 %

```

There are still unprocessed boxes. Recalculate the maximum number of lines needed, and move onto the next box (by incrementing \1@dpscL). If needed, restart the line numbering.

```

2853         \writtenlinesLfalse
2854         \ifbypstart@
2855             \global\line@num=0%
2856             \resetprevline@%
2857         \fi
2858 % Add the content of the optional argument of the previous \protect\cs{pend
2859 % \begin{macrocode}
2860         \csuse{after@pendL@the\1@dpscL}%
2861         \global\csundef{after@pendL@the\1@dpscL}%
2862 %

```

Check the number of lines

```

2863         \1@dcalc@maxoftwo{\the\usernamecount{1@dmaxlinesinpar\the\1@dpscL}}%
2864         {\the\@donetotallinesL}%
2865         {\usernamecount{1@dmaxlinesinpar\the\1@dpscL}}%
2866         \global\@donetotallinesL \z@
2867 %

```

Go to the next pstart

```

2868         \global\advance\1@dpscL \@ne
2869         \global\pstartnumtrue%
2870         \restore@pstartL@pc%
2871 %

```

Add notes of parallel ledgroup.

```

2872 \parledgroup@notes@endL
2873 \parledgroup@correction@notes@spacing@final{L}
2874 \else
2875 %

2876 \print@last@after@pendLtrue%
2877 \fi
2878 \fi
2879 \fi}
2880 %

2881 \newcommand*{\get@nextboxR}{%
2882 \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}% box is not empty
2883 \else% box is empty
2884 \@getnextboxtrue%
2885 \ifnum\usenamecount{1@dmaxlinesinpar\the\l@dpscR}>\donetotallinesR
2886 \parledgroup@notes@endR
2887 \unless\ifnosyncpstarts%
2888 \getnextboxfalse%
2889 \ifnomaxlines%
2890 \ifdim\pagetotal<\ledthegoal%
2891 \numdef{\@tmp}{\l@dpscR+1}%
2892 \ifcsdef{afterlines@pstart@\@tmp L}{%
2893 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp L}}
2894 %
2895 {}}%
2896 {\ifcsdef{minpage@pstart@\@tmp}%
2897 {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
2898 {\ifnum\numpagelinesR=1@dminpagelines%
2899 \getnextboxtrue%
2900 \fi%
2901 }%
2902 {\@getnextboxtrue}}}%
2903 {\@getnextboxtrue}}}%
2904 }%
2905 {}}%
2906 \fi%
2907 \fi%
2908 \fi%
2909 \else%
2910 \ifnomaxlines%
2911 \numdef{\@tmp}{\the\l@dpscR+1}%
2912 \ifcsdef{minpage@pstart@\@tmp}{%
2913 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
2914 {\ifdimgreater{\pagetotal}{\ledthegoal}%
2915 {\@getnextboxtrue}%
2916 {\@getnextboxfalse}%
2917 }%
2918 {\@getnextboxtrue}%

```

```

2919     }{}
2920     \fi%
2921 \fi%
2922 \if@getnextbox%
2923     \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscR}=\@donetotallinesR
2924     \parledgroup@notes@endR
2925     \fi
2926     \ifwrittenlinesR\else
2927     \@writelinesinparR
2928     \writtenlinesRtrue
2929     \fi
2930     \ifnum\l@dnumpstartsR>\l@dpscR
2931     \writtenlinesRfalse
2932     \ifbypstart@R
2933         \global\line@numR=0%
2934         \resetprevline@%
2935     \fi
2936     \csuse{after@pendR@\the\l@dpscR}%
2937     \global\csundef{after@pendR@\the\l@dpscR}%
2938     \l@dcalc@maxoftwo{\the\usernamecount{1@dmaxlinesinpar\the\l@dpscR}}%
2939                     {\the\@donetotallinesR}%
2940                     {\usernamecount{1@dmaxlinesinpar\the\l@dpscR}}%
2941     \global\@donetotallinesR \z@
2942     \global\advance\l@dpscR \@ne
2943     \global\pstartnumRtrue%
2944     \restore@pstartR@pc%
2945     \parledgroup@notes@endR
2946     \parledgroup@correction@notes@spacing@final{R}
2947 \else
2948     \print@last@after@pendRtrue%
2949 \fi
2950 \fi
2951 \fi}
2952
2953 %

```

## XX Page numbering

### XX.1 Global options

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

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

This counter will be increased in a patched version of the  $\LaTeX$ '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.

```

2954 \newcommand{\par@patch@thepage}{%
2955   \ifboolexpr{%
2956     bool{sameparallelpagenunder}%
2957     or bool{prevpgnotnumbered}%
2958   }%
2959   {%
2960     \patchcmd{\thepage}%
2961       {page}{par@page}%
2962       {}%
2963       {\led@error@fail@patch@thepage}%
2964   }{}%
2965 }%
2966
2967 \newcommand{\par@patch@pagenumbering}{%
2968   \ifboolexpr{%
2969     bool{sameparallelpagenunder}%
2970     or bool{prevpgnotnumbered}%
2971   }%
2972   {%
2973     \setcounter{par@page}{1}%
2974   }%
2975   {}%
2976   \par@patch@thepage%
2977 }%
2978
2979 \ifl@dmemoir%
2980   \apptocmd{\@mempnum}%
2981     {\par@patch@pagenumbering}%
2982     {}%
2983     {\led@error@fail@patch@@mempnum}%
2984
2985 \else%
2986   \apptocmd{\pagenumbering}%
2987     {\par@patch@pagenumbering}%
2988     {}%
2989     {\led@error@fail@patch@pagenumbering}%
2990

```

```

2991 \fi%
2992
2993 \AtBeginDocument{\par@patch@thepage}%
2994 %

```

**\@outputpage** As its name says, \@outputpage is a  $\TeX$ '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.

```

2995 \AtBeginDocument{%
2996   \apptocmd{\@outputpage}{%
2997     \ifsameparallelpagelapnumber%
2998       \ifl@dprintingpages%
2999         \ifodd\c@page\else%
3000           \stepcounter{par@page}%
3001         \fi%
3002       \else%
3003         \stepcounter{par@page}%
3004       \fi%
3005     \else%
3006       \stepcounter{par@page}%
3007     \fi%
3008   }%
3009   {}%
3010   {\led@error@fail@patch@@@outputpage}%
3011 }
3012 %

```

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

```

3013 \newcounter{par@page}%
3014 \setcounter{par@page}{1}%
3015 %

```

## XX.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 3016 \newif\ifPages@mainmatter
\Pages@mainmatter 3017 \newcommand{\Pages@mainmatter}{%
3018   \ifPages@mainmatter%
3019     \pagenumbering{arabic}%
3020     \addtocounter{page}{1}%
3021     \addtocounter{par@page}{-1}%
3022     \patchcmd{\thepage}{page}{par@page}{}{}%
3023   \fi%
3024 }
3025 %

```



## XXI 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 3026 \pretocmd{\chapter}{%
3027 \ifl@dprintingpages%
3028 \vbox{}%
3029 \fi%
3030 }%
3031 {}%
3032 {}%
3033 %
```

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

```
3034 \newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}}
3035 \eledsectnotoc{R}
3036 %
```

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

```
3037 \newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}}
3038 \eledsectmark{L}
3039 %
```

`\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`.

```
3040 \newskip\eledsection@correcting@skip
3041 %
```

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

```
3042 \newwrite\eled@sectioningR@out
3043 %
```

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

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

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

```
3044 \def\l@prev@pbR{}
3045 \def\l@prev@nopbR{}
3046 %
```

`\ledpbR` The `\ledpbR` macro writes the call to `\led@pbR` in line-list file. The `\ledpbnR` macro writes the call to `\led@pbnR` in line-list file. The `\lednopbR` macro writes the call to `\led@nopbR` in line-list file. The `\lednopbnR` macro writes the call to `\led@nopbnR` in line-list file.

```
3047 \newcommand{\ledpbR}{\write\linenum@outR{\string\led@pbR}}
3048 \newcommand{\ledpbnR}[1]{\write\linenum@outR{\string\led@pbnR{#1}}}
3049 \newcommand{\lednopbR}{\write\linenum@outR{\string\led@nopbR}}
3050 \newcommand{\lednopbnR}[1]{\write\linenum@outR{\string\led@nopbnR{#1}}}
3051 %
```

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

```
3052 \newcommand{\led@pbR}{\listxadd{\l@prev@pbR}{\the\absline@numR}}
3053 \newcommand{\led@pbnR}[1]{\listxadd{\l@prev@pbR}{#1}}
3054 \newcommand{\led@nopbR}{\listxadd{\l@prev@nopbR}{\the\absline@numR}}
3055 \newcommand{\led@nopbnR}[1]{\listxadd{\l@prev@nopbR}{#1}}
3056 %
```

## XXIII Parallel ledgroup

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

```
3057 \newmarks\parledgroup@
3058 \newmarks\parledgroup@series
3059 \newmarks\parledgroup@type
3060 %
```

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

```
3061 \newcommand{\parledgroup@notes@startL}{%
3062   \ifnum\usernamecount{1dmaxlinesinpar}\the\l@dpscL>0%
3063   \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
bhooknoteX@\splitfirstmarks\parledgroup@series}}{}%
```

```

3064 \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
bhookXnote@\splitfirstmarks\parledgroup@series}}{}%
3065 \fi%
3066 \global\ledgroupnotesL@true%
3067 \insert@noterule@ledgroup{L}%
3068 }
3069 \newcommand{\parledgroup@notes@startR}{%
3070 \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscR}>0%
3071 \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
bhooknoteX@\splitfirstmarks\parledgroup@series}}{}%
3072 \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
bhookXnote@\splitfirstmarks\parledgroup@series}}{}%
3073 \fi%
3074 \global\ledgroupnotesR@true%
3075 \insert@noterule@ledgroup{R}%
3076 }
3077 %

```

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

```

3078 \newcommand{\parledgroup@notes@endL}{%
3079 \global\ledgroupnotesL@false%
3080 }
3081 \newcommand{\parledgroup@notes@endR}{%
3082 \global\ledgroupnotesR@false%
3083 }
3084 %

```

`\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`

```

3085 \newcommand{\insert@noterule@ledgroup}[1]{
3086 \IfStrEq{\splitbotmarks\parledgroup@}{begin}{%
3087 \IfStrEq{\splitbotmarks\parledgroup@type}{Xfootnote}{
3088 \csuse{ifledgroupnotes#1@}
3089 \vskip\skip\csuse{mp\splitbotmarks\parledgroup@series footins}
3090 \csuse{\splitbotmarks\parledgroup@series footnoterule}
3091 \fi
3092 }
3093 {}
3094 \IfStrEq{\splitbotmarks\parledgroup@type}{footnoteX}{
3095 \csuse{ifledgroupnotes#1@}
3096 \vskip\skip\csuse{mpfootins\splitbotmarks\parledgroup@series}
3097 \csuse{footnoterule\splitbotmarks\parledgroup@series}
3098 \fi
3099 }{}
3100 }
3101 {}
3102 }

```

3103 %

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

```
3104 \newcommand{\setparledgroupnotespacing}[1]{\gdef\@parledgroupnotespacing
      {#1}}
3105 \newcommand{\@parledgroupnotespacing}{}
3106 %
```

`\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`.

```
3107 \dimdef{\parledgroup@notespacing@correction}{0pt}
3108 \newcommand{\parledgroup@notespacing@set@correction}{%
3109   {\@getfirstseries\csuse{Xnotefontsize@}\@firstseries}%We suppose all the
      series has the same footnote size setup
3110   \@parledgroupnotespacing\dimgdef{\temp@spacing}{\baselineskip}}%
3111   \dimgdef{\parledgroup@notespacing@correction}{\baselineskip-\temp@spacing
      }%
3112 }
3113 %
```

`\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.

```
3114 \newcommand{\parledgroup@correction@notespacing@init}{
3115   \dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}
3116   \dimdef{\parledgroup@notespacing@correction@modulo}{0pt}
3117 }
3118 \parledgroup@correction@notespacing@init
3119 %
```

`\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`.

```
3120 \newcommand{\parledgroup@correction@notespacing@final}[1]{
3121   \ifparledgroup
3122   \vspace{\parledgroup@notespacing@correction@accumulated}
3123   \parledgroup@correction@notespacing@init%
3124   \ifstrequal{#1}{L}{
3125     \numdef{\@checking}{\the\l@dpscl-1}
3126   }{
3127     \numdef{\@checking}{\the\l@dpscr-1}
3128   }
}
```

```

3129 \dimdef{\@beforenotes@current@diff}{\csuse{@parledgroup@beforenotes@
@checking L}-\csuse{@parledgroup@beforenotes@}@checking R}}%
3130 \ifstrequal{#1}{L}%
3131 {% Left
3132 \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{-\
@beforenotes@current@diff}}%
3133 }%
3134 {% Right
3135 \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{\
@beforenotes@current@diff}}{}
3136 }%
3137 \fi
3138 }
3139 %

```

`\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@correction` and `\parledgroup@notespacing@correction@modulo`. If `\parledgroup@notespacing@correction@modulo` 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.

```

3140 \newcommand{\parledgroup@correction@notespacing}[1]{%
3141 \csuse{ifledgroupnotes#1@}%
3142 \vspace{-\parledgroup@notespacing@correction}%
3143 \dimdef{\parledgroup@notespacing@correction@accumulated}{\
parledgroup@notespacing@correction@accumulated+\
parledgroup@notespacing@correction}%
3144 \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo+\
parledgroup@notespacing@correction}%
3145 \ifdimless{\parledgroup@notespacing@correction@modulo}{\baselineskip
}{\advance\numpagelinesL -\@ne%
3146 \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo-\baselineskip}%
3147 }% mean greater than equal
3148 \fi%
3149 }
3150 %

```

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

```

3151 \dimdef\parledgroup@beforenotesL{0pt}
3152 \dimdef\parledgroup@beforenotesR{0pt}
3153 %

```

**\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.

```

3154 \newcommand{\parledgroup@beforenotes@save}[1]{
3155   \ifparledgroup
3156     \csdimgdef{@parledgroup@beforenotes@the\csuse{1@dnumstarts#1}#1}{\
csuse{parledgroup@beforenotes#1}}
3157     \csdimgdef{parledgroup@beforenotes#1}{0pt}
3158   \fi
3159 }
3160 %

```

## XXIV Compatibility with eledmac

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

```

3161 \ifeledmaccompat%
3162
3163
3164   \unless\ifnocritical@
3165     \let\onlyXside\Xonlyside
3166   \fi
3167 \fi
3168 %

```

## XXV 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 `\setstanzaindents` 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. 335.

## Appendix A.7 Migration to `reledpar` 2.6.0

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

## Appendix A.8 Migration to `reledpar` 2.6.1

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



## References

- [LW90] John Lavagnino and Dominik Wujastyk. ‘An overview of edmac: a PLAIN TeX format for critical editions’. *TUGboat*, **11**, 4, pp. 623–643, November 1990. (Code available from CTAN in macros/plain/contrib/edmac)
- [Wil02] Peter Wilson. *The memoir class for configurable typesetting*. November 2002. (Available from CTAN in macros/latex/contrib/memoir)
- [Wil04] Peter Wilson and Maïeul Rouquette. *eledmac A presumptuous attempt to port EDMAC, TABMAC and EDSTANZA to LaTeX*. December 2004. (Available from CTAN in macros/latex/contrib/eledmac)

## Index

### Symbols

<code>\@adv</code>	1
<code>\@astanza@line</code>	1
<code>\@cs@linesinparL</code>	1
<code>\@cs@linesinparR</code>	1
<code>\@cs@linesonpageL</code>	1
<code>\@cs@linesonpageR</code>	1
<code>\@donereallinesL</code>	1
<code>\@donereallinesR</code>	1
<code>\@donetotallinesL</code>	1
<code>\@donetotallinesR</code>	1
<code>\@eledsectionL</code>	1
<code>\@eledsectionR</code>	1
<code>\@lab</code>	1
<code>\@lopL</code>	1
<code>\@lopR</code>	1
<code>\@nl</code>	1
<code>\@nl@regR</code>	1
<code>\@outputpage</code>	1
<code>\@par@sync@option</code>	1
<code>\@par@this@sync@option</code>	1
<code>\@parledgroupnotespacing</code>	1
<code>\@pend</code>	1
<code>\@pendR</code>	1
<code>\@pstart</code>	1
<code>\@pstartR</code>	1
<code>\@pstartsfalse</code>	1
<code>\@pstartstrue</code>	1
<code>\@ref</code>	1
<code>\@ref@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	111
COMMAND\+	47
COMMAND\@Rlineflag	75, 145
COMMAND\@adv	38, 144
COMMAND\@cs@linesinparL	102
COMMAND\@cs@linesonpageL	102
COMMAND\@eledsectionL	85
COMMAND\@eledsectionR	85
COMMAND\@eledsectmark	113
COMMAND\@eledsectnotoc	113
COMMAND\@footnotetext	77
COMMAND\@goalfraction	12, 106
COMMAND\@l@dttempcnta	69
COMMAND\@lab	36, 75, 144
COMMAND\@lopL	42, 96, 102
COMMAND\@lopR	42, 102
COMMAND\@mempnum	111
COMMAND\@namedef	79
COMMAND\@namuse	79
COMMAND\@nl	36, 43, 75, 144
COMMAND\@nl@regR	36
COMMAND\@outputpage	111, 112
COMMAND\@page	75
COMMAND\@par@sync@option	36
COMMAND\@parledgroupnotespacing	116
COMMAND\@pend	41
COMMAND\@pendR	41
COMMAND\@pstart	41
COMMAND\@pstartstrue	83
COMMAND\@ref	39–41, 43, 144
COMMAND\@ref@regR	40
COMMAND\@set	38, 144
COMMAND\@sw	40
COMMAND\AtBeginPairs	8, 49, 142
COMMAND\AtEveryPend	142, 144
COMMAND\AtEveryPstart	2, 17, 56, 86, 142, 144
COMMAND\AtEveryPstartCall	2, 17, 56, 143
COMMAND\AtEveryStanza	146
COMMAND\AtEveryStopStanza	146
COMMAND\Clear the right lines for \@read@linelist	36
COMMAND\Columns	7, 8, 17, 48, 85, 89, 139–141, 144, 145
COMMAND\Columns@print@after@pend	92
COMMAND\Columns@print@before@pstart	92
COMMAND\Lcolwidth	8, 10, 94
COMMAND\Leftsidehook	139

COMMAND\Leftsidehookend	139
COMMAND\MakePerPage	13
COMMAND\Pages	4, 7, 10, 12, 17, 47, 48, 67, 70, 72, 93, 94, 100, 110, 112, 116, 139, 141, 143–145
COMMAND\Pages@mainmatter	112
COMMAND\Pairs	47
COMMAND\Rcolwidth	8, 10, 94
COMMAND\Rightsidehook	139
COMMAND\Rightsidehookend	139
COMMAND\Rlineflag	119
COMMAND\Xendlineflag	14, 145
COMMAND\Xlineflag	14, 120, 145
COMMAND\Xmaxhnotes	13
COMMAND\Xnoteswidthliketwocolumns	9, 142
COMMAND\Xonlyside	14, 70, 120
COMMAND\Xtxtbeforenotes	146
COMMAND\&	18
COMMAND\absline@numR	35
COMMAND\add@penalties	69
COMMAND\add@penaltiesL	69
COMMAND\advanceline	38, 43, 144
COMMAND\affixline@num	64
COMMAND\affixline@numR	64, 139, 140
COMMAND\affixpstart@num	67
COMMAND\affixpstart@numR	67
COMMAND\affixside@note	76
COMMAND\aftercolumnseparator	9, 89, 142
COMMAND\araw@textfalse	84
COMMAND\araw@texttrue	84
COMMAND\at@begin@pairs	49
COMMAND\autopar	17
COMMAND\ballast@count	69
COMMAND\baselineskip	89, 117
COMMAND\bbl@set@language	80, 81, 144
COMMAND\beforecolumnseparator	9, 89, 142
COMMAND\begin	18
COMMAND\beginnumbering	15–17, 26, 36, 89, 90, 140, 142, 144
COMMAND\beginnumberingR	31, 43
COMMAND\bf	140
COMMAND\bfseries	140
COMMAND\brokenpenalty	69
COMMAND\chapter	113, 139
COMMAND\check@goal	106
COMMAND\check@pstarts	83
COMMAND\checkpageL	96, 103
COMMAND\checkpb@columns	88
COMMAND\checkpbL	104
COMMAND\checkpbR	104
COMMAND\checkraw@text	84
COMMAND\checkverseL	105
COMMAND\checkverseR	105

COMMAND\clear(double)page	101
COMMAND\clearl@dleftpage	101
COMMAND\clearl@drighpage	101
COMMAND\clearpage	101, 143
COMMAND\cleartoevenpage	101
COMMAND\cleartol@devenpage	101, 112
COMMAND\columnrulewidth	8, 89
COMMAND\columns@position	89
COMMAND\columnseparator	8, 9
COMMAND\columnspostion	9, 142
COMMAND\correct@Xfootins@box	72, 73, 143
COMMAND\correct@footinsX@box	72, 143
COMMAND\critext	143
COMMAND\csname	48
COMMAND\displaywidowpenalty	69
COMMAND\do@actions	63
COMMAND\do@actions@fixedcode	139
COMMAND\do@actions@nextR	63
COMMAND\do@actionsR	63, 139
COMMAND\do@ballast	69
COMMAND\do@ballastR	63
COMMAND\do@insidelineLhook	141
COMMAND\do@insidelineRhook	141
COMMAND\do@line	57
COMMAND\do@line(L/R)	60
COMMAND\do@lineL	57, 69, 139, 140
COMMAND\do@lineLhook	139
COMMAND\do@lineR	60, 139, 140, 142
COMMAND\do@lineRhook	139
COMMAND\do@lockoff	144
COMMAND\do@lockoffR	39
COMMAND\do@lockon	144
COMMAND\do@lockonR	38
COMMAND\doinsidelineLhook	142
COMMAND\doinsidelineRhook	142
COMMAND\dolineLhook	142
COMMAND\dolineRhook	142
COMMAND\edindex	143
COMMAND\edlabel	140, 143
COMMAND\edtext	39, 43, 44, 142, 143
COMMAND\eled@sectioningR@out	113
COMMAND\eledchapter	143
COMMAND\eledsection	142, 143, 145
COMMAND\eledsection@correcting@skip	113
COMMAND\eledsectmark	20, 113
COMMAND\eledsectnotoc	20, 113
COMMAND\eledxxx	142
COMMAND\end	18
COMMAND\endgraf	55
COMMAND\endlock	44, 144

COMMAND\endnumbering	9, 15, 17, 27, 144
COMMAND\endsub	43, 144
COMMAND\endumbering	15
COMMAND\expandafter	46
COMMAND\extensionchars	26
COMMAND\firstlinenum	16, 141, 144, 145
COMMAND\firstsublinenum	141, 144, 145
COMMAND\fix@page	37, 144
COMMAND\flag@end	43, 139, 142
COMMAND\flag@start	43, 142
COMMAND\flush@notesR	70
COMMAND\footnote	48
COMMAND\footnoteX	45, 46
COMMAND\footnoteX@reading	146
COMMAND\footnoteXmk	14
COMMAND\footnoteXnomk	14, 45
COMMAND\frontmatter	12, 20
COMMAND\get@nextboxL	140
COMMAND\get@nextboxR	140
COMMAND\getline@numL	62
COMMAND\getline@numR	62
COMMAND\getlinesfrompagelistL	102
COMMAND\getlinesfrompagelistR	102
COMMAND\getlinesfromparlistL	102
COMMAND\getlinesfromparlistR	102
COMMAND\gl@p	46
COMMAND\goalfraction	119
COMMAND\hangingsymbol	119, 140
COMMAND\hfill	88, 89
COMMAND\hidenumbering	16, 144
COMMAND\if@getnextbox	106
COMMAND\ifPages@mainmatter	112
COMMAND\ifbypage@	145
COMMAND\ifbypstart@R	145
COMMAND\ifdim	88
COMMAND\ifinserthangingsymbol	77
COMMAND\ifinserthangingsymbolR	77
COMMAND\ifl@dpagefull	103
COMMAND\ifl@dpaging	24, 142
COMMAND\ifl@dpairing	24, 139
COMMAND\ifl@dsamelang	142
COMMAND\ifl@dsamepage	103
COMMAND\ifl@pagefull	96
COMMAND\ifledRcol	24
COMMAND\iflinenumberLevenifblank	34
COMMAND\iflinenumberRevenifblank	34
COMMAND\iflledRcol	140
COMMAND\ifnumberedpar@	52
COMMAND\ifnumberingR	140
COMMAND\ifnumberpstart	48

COMMAND\ifpst@rtedL	26, 27, 53, 139
COMMAND\ifpst@rtedR	26
COMMAND\ifsublines@	38
COMMAND\insert@countR	39
COMMAND\insert@noterule@ledgroup	115
COMMAND\insertlines@list	39
COMMAND\insertlines@listR	40
COMMAND\inserts@list	52
COMMAND\inserts@listR	68
COMMAND\l@d@nums	44
COMMAND\l@d@set	38, 44, 144
COMMAND\l@dLcolrawbox	52
COMMAND\l@dLcolrawbox1	82
COMMAND\l@dLcolrawbox2	82
COMMAND\l@dRcolrawbox	52
COMMAND\l@dbfnote	77, 144
COMMAND\l@dcalc@maxoftwo	103
COMMAND\l@dcalc@minoftwo	103
COMMAND\l@dchecklang	139, 141
COMMAND\l@dcsnote	142
COMMAND\l@dleftbox	57, 96, 143
COMMAND\l@dlinenumR	34, 139
COMMAND\l@dlsnote	142
COMMAND\l@dmake@labels	75
COMMAND\l@dmaxlinesinpar	95
COMMAND\l@dmaxlinesinpar1	83
COMMAND\l@dminpagelines	95, 139
COMMAND\l@dnumpsstartsL	82, 139
COMMAND\l@dprintingcolumnstrue	143
COMMAND\l@dprintingpagestrue	143
COMMAND\l@dpscL	86, 95, 108
COMMAND\l@dpscR	86, 95
COMMAND\l@drsnote	142
COMMAND\l@dsetupmaxlinecounts	83
COMMAND\l@duselanguage	80, 81, 139
COMMAND\l@dzeromaxlinecounts	83
COMMAND\l@prev@nopbR	114
COMMAND\l@prev@pbR	114
COMMAND\labelstarttrue	140
COMMAND\labelref@list	75
COMMAND\labelref@listR	75
COMMAND\lang	80
COMMAND\last@page@numR	37
COMMAND\led	140
COMMAND\led@nopbR	114
COMMAND\led@nopbnumR	114
COMMAND\led@pbR	114
COMMAND\led@pbnumR	114
COMMAND\ledinnerrote	19
COMMAND\ledleftnote	19

COMMAND\lednopb	18, 88, 104
COMMAND\lednopbR	114
COMMAND\lednopbnumR	114
COMMAND\ledouterote	19
COMMAND\ledpb	88, 104
COMMAND\ledpbR	114
COMMAND\ledpbnumR	114
COMMAND\ledrightnote	19
COMMAND\ledsidenote	19
COMMAND\ledstrutL	139
COMMAND\ledstrutR	139, 145
COMMAND\ledthegoal	106
COMMAND\ledtrutL	139, 145
COMMAND\leftlinenumR	34, 139
COMMAND\let	46
COMMAND\line@list@R	40
COMMAND\line@list@stuff	36, 43
COMMAND\line@margin	31
COMMAND\line@marginR	31, 139
COMMAND\line@numR	35
COMMAND\lineation	16, 143
COMMAND\lineation*	16, 31, 142
COMMAND\lineationR	16, 30, 143
COMMAND\linenum@out	75
COMMAND\linenum@outR	42
COMMAND\linenumberLevenifblanktrue	17, 146
COMMAND\linenumberRevenifblank	146
COMMAND\linenumberRevenifblanktrue	17
COMMAND\linenumberstyle	16, 120
COMMAND\linenumberstyle*	120
COMMAND\linenumberstyleR	16, 120
COMMAND\linenumincrement	16, 141, 144, 145
COMMAND\linenummargin	16, 31, 139, 144–146
COMMAND\linenummargin*	16, 31, 145
COMMAND\linenummarginR	16, 31, 145
COMMAND\linenumrepR	33, 139
COMMAND\linesinpar@listL	41, 102
COMMAND\linesonpage@listL	42, 102
COMMAND\lock@off	39
COMMAND\lock@on	38
COMMAND\mainmatter	2, 12, 20, 145
COMMAND\makeatletter	60
COMMAND\maxchunks	7, 18, 82, 83
COMMAND\maxhnotesX	13
COMMAND\memorydump	15, 29
COMMAND\n@num	143
COMMAND\new@lineL	43
COMMAND\new@lineR	43
COMMAND\newhookcommand@series	47
COMMAND\newif	143

COMMAND\newpage	100, 101, 143
COMMAND\newseries	48
COMMAND\newseries@	44
COMMAND\newseries@par	44, 47, 48
COMMAND\noledxxx	142
COMMAND\nomark@	45
COMMAND\nomaxlines	41
COMMAND\normalbfnoteX	139, 144
COMMAND\notesXwidthliketwocolumns	9, 142
COMMAND\num@lines	69
COMMAND\num@lines(R)	52
COMMAND\numberingR	28
COMMAND\numberlinefalse	7
COMMAND\numberonlyfirstinline	140
COMMAND\numberpstartfalse	16
COMMAND\numberpstarttrue	16, 140, 145
COMMAND\one@line	52, 77
COMMAND\one@lineR	52
COMMAND\onlyXside	120
COMMAND\onlysideX	14, 70, 144
COMMAND\otherlanguage	144
COMMAND\page@action	38, 144
COMMAND\pagenumbering	111, 145
COMMAND\pages	12
COMMAND\pagetotal	96, 143
COMMAND\par@line	69
COMMAND\par@line(R)	52
COMMAND\par@patch@pagenumbering	111
COMMAND\par@patch@thepage	111
COMMAND\par@sync@option	22
COMMAND\parledgroup@	114
COMMAND\parledgroup@beforenotes@save	118
COMMAND\parledgroup@beforenotesL	117
COMMAND\parledgroup@beforenotesR	117
COMMAND\parledgroup@correction@notespacing	117
COMMAND\parledgroup@correction@notespacing@final	116
COMMAND\parledgroup@correction@notespacing@init	116
COMMAND\parledgroup@notes@endL	115
COMMAND\parledgroup@notes@endR	115
COMMAND\parledgroup@notes@startL	114
COMMAND\parledgroup@notes@startR	114
COMMAND\parledgroup@notespacing@correction	116, 117
COMMAND\parledgroup@notespacing@correction@accumulated	117
COMMAND\parledgroup@notespacing@correction@modulo	117
COMMAND\parledgroup@notespacing@set@correction	116
COMMAND\parledgroup@series	114
COMMAND\parledgroup@type	114
COMMAND\parledgroupnotespacing	119
COMMAND\parledgrouptrue	19
COMMAND\patchcmd	145



COMMAND\pausenumbering	9, 29, 146
COMMAND\pend	3, 7, 10, 17–19, 48, 52, 55, 56, 82, 92, 142, 144, 145
COMMAND\pendL	141
COMMAND\pendR	141
COMMAND\pends	17
COMMAND\perpage	13
COMMAND\prev@nopbR	114
COMMAND\prev@pbR	114
COMMAND\prevpgstyle	23
COMMAND\print@Xnotes	70
COMMAND\print@Xnotes@forpages	70, 143
COMMAND\print@columnseparator	88, 141
COMMAND\print@eledsectionL	59
COMMAND\print@line	59
COMMAND\print@lineL	59
COMMAND\print@notesX@forpages	143
COMMAND\printlinenumR	120
COMMAND\printlinesR	145
COMMAND\pstart	3, 7, 10, 16–19, 30, 44, 48, 52, 53, 55, 56, 82, 86, 92, 140, 142, 144, 145
COMMAND\pstartL	56, 141
COMMAND\pstartR	56, 140, 141
COMMAND\pstartinfootnote	143
COMMAND\raw@text	82
COMMAND\read@linelist	36, 122, 144
COMMAND\ref@reg	39
COMMAND\ref@regR	39, 144
COMMAND\relax	120
COMMAND\reledmac	144
COMMAND\renewcommand	119
COMMAND\resumenumbering	9, 29, 142, 146
COMMAND\resumenumberingR	31, 142
COMMAND\rightlinenumR	34, 139
COMMAND\section	139
COMMAND\section@num	26
COMMAND\selectlanguage	17, 80, 81
COMMAND\set@continuousnumberingforR	31
COMMAND\set@line	44, 144
COMMAND\set@line@action	38, 144
COMMAND\setRlineflag	16, 119
COMMAND\setgoalfraction	12, 119
COMMAND\sethangingsymbol	18
COMMAND\setline	38, 44, 144
COMMAND\setlinenum	38, 44, 144
COMMAND\setnoteposition...	90
COMMAND\setparledgroupnotespacing	119, 145
COMMAND\setposition...	90
COMMAND\setstanzaindents	9, 18, 119
COMMAND\setwidth...	89
COMMAND\sidenotemargin	19, 141
COMMAND\sidenotemargin*	19, 142

COMMAND\skipnumbering	16, 143
COMMAND\sloppy	8
COMMAND\stanza	7, 9, 16, 18, 50, 77, 120, 140
COMMAND\stanzanumtrue	18
COMMAND\startlock	44, 144
COMMAND\startsub	43, 144
COMMAND\sub@action	38, 144
COMMAND\sub@off	75
COMMAND\sub@on	75
COMMAND\subline@numR	35
COMMAND\sublinenumberstyle	16, 120
COMMAND\sublinenumberstyle*	120
COMMAND\sublinenumberstyleR	16, 120
COMMAND\sublinenumincrement	141, 144, 145
COMMAND\sublinenumrepR	33, 139
COMMAND\syntaxonly	146
COMMAND\sza@0@	18
COMMAND\textheight	13
COMMAND\textwidth	50
COMMAND\the@labelX	145
COMMAND\thefootnoteX	141
COMMAND\theledlanguageL	80, 81
COMMAND\theledlanguageR	80, 81
COMMAND\thepage	20, 110, 111
COMMAND\thepstartL	16, 140
COMMAND\thepstartR	16, 140
COMMAND\thestanzaL	18
COMMAND\thestanzaR	18
COMMAND\ vbox	53
COMMAND\vl@dbfnote	77
COMMAND\vskip	115
COMMAND\vsplit	69
COMMAND\widthliketwocolumns	9
COMMAND\widthliketwocolumnsfalse	9
COMMAND\widthliketwocolumnstrue	9
COMMAND\xflagref	145
COMMAND\xright@appenditem	46
COMMAND\xspace	21
COMMAND\xxxfootstart	90
COMMAND\xxxmatter	111
ENVIRONMENTLeftside	50
ENVIRONMENTRightside	51
ENVIRONMENTastanza	18, 77, 78, 120, 145
ENVIRONMENTcolumns	28, 146
ENVIRONMENTledgroup	6, 146
ENVIRONMENTleft	16
ENVIRONMENTpages	28, 49, 145, 146
ENVIRONMENTpairs	49, 145
PACKAGEEDMAC	121
PACKAGEEDSTANZA	121

PACKAGEEledmac	44, 143
PACKAGEEledpar	143
PACKAGETABMAC	121
PACKAGEbabel	17, 80, 81, 146
PACKAGEedmac	121
PACKAGEeledmac	4, 82, 118, 119, 121, 141, 142, 144, 145
PACKAGEeledpar	5, 6, 13, 32, 119, 120, 141–143
PACKAGEetoolbox	88, 114
PACKAGEledmac	6
PACKAGEledpar	1, 6
PACKAGEMemoir	121
PACKAGEMusixtex	142
PACKAGEperpage	2, 13, 120, 146
PACKAGEpolyglossia	17, 80, 81, 146
PACKAGEreledmac	1, 3, 5–7, 9, 13, 16–21, 23, 24, 26, 27, 31, 32, 35–39, 41–45, 47, 48, 59, 67, 75, 77, 94, 100, 119, 120, 144–146
PACKAGEreledpar	1, 3, 5–7, 9–12, 18–23, 30, 35, 41, 42, 44, 47, 48, 75, 119, 120, 145
PACKAGEsetspace	2, 20
PACKAGESyntonly	146
PACKAGEXkeyval	21
PACKAGEXr	146

## A

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

## B

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

\do@lockonR .....	1
\doinsidelineLhook .....	1
\doinsidelineRhook .....	1
\dolineLhook .....	1
\dolineRhook .....	1
\dump@pstartL@pc .....	1
\dump@pstartR@pc .....	1

## E

\edlabel .....	1
\edtext .....	1
\eled@sectioningR@out .....	1
\eledsection@correcting@skip .....	1
\eledsectmark .....	1, 20
\eledsectnotoc .....	1, 20
\endlock .....	1
\endnumbering .....	1, 15
\endnumberingR .....	1
\endsub .....	1
environments:	
astanza .....	18
Leftside .....	15
pages .....	9
pairs .....	8
Rightside .....	15

## F

\f@x@l@cksR .....	1
\finish@Pages@notes .....	1
\first@linenum@out@Rfalse .....	1
\first@linenum@out@Rtrue .....	1
\firstlinenum .....	1, 16
\firstlinenum* .....	1, 16
\firstlinenumR .....	1, 16
\firstsublinenum .....	1, 16
\firstsublinenum* .....	1, 16
\firstsublinenumR .....	1, 16
\fix@page .....	1
\flag@end .....	1
\flag@start .....	1
\flush@notesR .....	1
\footnote@reading .....	1
\footnote@typeset .....	1
\footnoteXmk .....	14
\footnoteXnomk .....	14

## G

\get@familiarfootnote@number .....	1
\get@nextboxL .....	1
\get@nextboxR .....	1

\getline@numR .....	1
\getlinesfrompagelistL .....	1
\getlinesfrompagelistR .....	1
\getlinesfromparlistL .....	1
\getlinesfromparlistR .....	1
\goalfraction .....	1

## H

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

## I

\if@getnextbox .....	1
\if@pstarts .....	1
\ifaraw@text .....	1
\iffirst@linenum@out@R .....	1
\ifistanzaL .....	1
\ifistanzaR .....	1
\ifl@dpagfull .....	1
\ifl@dpaging .....	1
\ifl@dpairing .....	1
\ifl@dsamepage .....	1
\ifl@dusedbabel .....	1
\ifledRcol .....	1
\iflinenumberLevenifblank .....	1
\iflinenumberRevenifblank .....	1
\ifnomaxlines .....	1
\ifnosyncpstarts .....	1
\ifPages@mainmatter .....	1
\ifprevpgnotnumbered .....	1
\ifprint@last@after@pendL .....	1
\ifprint@last@after@pendR .....	1
\ifpst@rtedL .....	1
\ifpst@rtedR .....	1
\ifpstartnumR .....	1
\ifsameparallelpagenumber .....	1
\ifshiftedpstarts .....	1
\ifwidthliketwocolumns .....	1
\ifwrittenlinesL .....	1
\init@series@par .....	1
\initnumbering@sectcountR .....	1
\insert@countR .....	1
\insert@noterule@ledgroup .....	1
\inserthangingsymbolL .....	1
\inserthangingsymbolR .....	1
\insertlines@listR .....	1
\inserts@listR .....	1

## L

\l@d@set .....	1
\l@dbfnote .....	1

<code>\l@dc@maxchunks</code>	1
<code>\l@dcalc@maxoftwo</code>	1
<code>\l@dcalc@minoftwo</code>	1
<code>\l@dcalcnun</code>	1
<code>\l@dchecklang</code>	1
<code>\l@dleftbox</code>	1
<code>\l@dlinenumR</code>	1
<code>\l@dmake@labelsR</code>	1
<code>\l@dminpagelines</code>	1
<code>\l@dnumpstartsL</code>	1
<code>\l@dnumpstartsR</code>	1
<code>\l@dpagfullfalse</code>	1
<code>\l@dpagfulltrue</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@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
<code>Leftside (environment)</code> .....	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> .....	16
<code>\linenumberRevenifblanktrue</code> .....	16
<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>optioncontinuousnumberingwithcolumns</code> .....	9, 146
<code>optionnomaxlines</code> .....	10, 11, 22
<code>optionnosyncpstarts</code> .....	12, 22, 106
<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>\parledgrouptype@</code> .....	1
<code>\pausenumberingR</code> .....	1
<code>\pend</code> .....	17
<code>\pendL</code> .....	1
<code>\pendR</code> .....	1
<code>\prev@nopbR</code> .....	1
<code>\prev@pbR</code> .....	1
<code>\prevpgstyle</code> .....	1

\print@columnseparator	1
\print@eledsectionL	1
\print@eledsectionR	1
\print@lineL	1
\print@lineR	1
\print@notesX@forpages	1
\print@Xnotes@forpages	1
\pstart	17
\pstartL	1
\pstartR	1

## R

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

## S

\save@familiarfootnote@number	1
\save@section@number	1
\section@numR	1
\selectlanguage	1
\set@continuousnumberingforR	1
\set@line	1
\set@line@action	1
\set@sectcountR	1
\setgoalfraction	1, 12
\sethangingsymbol	18
\setline	1
\setlinenum	1
\setnamebox	1
\setnotepositionliketwocolumns@C	1
\setnotepositionliketwocolumns@L	1
\setnotepositionliketwocolumns@R	1
\setpositionliketwocolumns@C	1
\setpositionliketwocolumns@L	1
\setpositionliketwocolumns@R	1
\setRlineflag	16
\setwidthliketwocolumns@C	1
\setwidthliketwocolumns@L	1

<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>

## T

<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> , 18
<code>\thestanzaR</code> .....	<u>1</u> , 18

## U

<code>\unhnamebox</code> .....	<u>1</u>
<code>\unvnamebox</code> .....	<u>1</u>
<code>\usernamecount</code> .....	<u>1</u>

## W

<code>\widthliketwocolumns</code> .....	9
---	---

## X

<code>\Xendlineflag</code> .....	14
<code>\Xlineflag</code> .....	14
<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	86
\Pages: Added \l@duselanguage to \Pages	96
General: Added section of babel related code	80
Fix babel problems	1
v0.3.0.	
\Pages: Added \ledstrutL to \Pages	96
Added \ledstrutR to \Pages	98
\Rightsidehookend: Added \Leftsidehook, \Leftsidehookend, \Rightsidehook and \Rightsidehookend	51
\affixline@numR: Changed \affixline@numR to match neweledmac	64
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR	63
\do@lineL: Added \do@lineLhook to \do@lineL	57
Simplified \do@lineL by using macros for some common code	57
\do@lineR: Changed \do@lineR similarly to \do@lineL	60
\flag@end: Removed extraneous spaces from \flag@end	43
\ifledRcol: Moved \ifl@dpairing toeledmac	24
\ifpst@rtedR: Moved \ifpst@rtedL toeledmac	26
\l@dlinenumR: Simplified \leftlinenumR and \rightlinenumR by introducing \l@dlinenumR	34
\l@dnumpstartsR: Moved \l@dnumpstartsL toeledmac	82
\ledstrutR: Added \ledtrutL and \ledstrutR	100
\normalbfnoteX: Removed extraneous spaces from \normalbfnoteX	77
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR	33
General: Added \do@lineLhook and \do@lineRhook	60
Added hooks into Leftside environment	50
Reorganize for ledarab	1
v0.3.a.	
\line@marginR: Do not just set \line@marginR in \linenummargin	31
General: Minor \linenummargin fix	1
v0.3.b.	
\Pages: Added \l@dminpagelines calculation for succeeding page pairs	99
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 use \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
Possibilty 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> . . . . .	87
<code>\affixline@numR</code> : Changed <code>\affixline@numR</code> to allow to disable line numbering (like in <code>eledmac 0.15</code> ). . . . .	64
<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> ). . . . .	106
<code>Pstart</code> number can be printed in side . . . . .	108
<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> ). . . . .	57
Lineation can be by <code>pstart</code> (like in <code>eledmac 0.15</code> ). . . . .	30
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> . . . . .	30
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> ). . . . .	52
General: <code>Shiftedverses</code> becomes <code>shiftedpstarts</code> . . . . .	1
v1.1.1.	
<code>\pstartR</code> : Correct <code>\pstartR</code> bug introduced by 1.1. . . . .	52
v1.1.2.	
<code>\affixside@noteR</code> : Remove spurious space between line number and line content . .	76
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. . . . .	28

v1.3.0.	
<code>\edtext</code> : Manage RTL language. . . . .	44
v1.3.1.	
<code>\l@dbfnote</code> : Compatibility of standard footnotes with <code>eledmac</code> when theses footnotes contain any commands. . . . .	77
v1.3.2.	
General: Debug with some classes. . . . .	1
v1.3.3.	
<code>\l@dbfnote</code> : Spurious space with footnote in right column. . . . .	77
General: Debugging the left notes of the right column. . . . .	76
v1.3.4.	
General: Allow use of commands in sidenotes, as introduced by <code>eledmac</code> 1.0. . . . .	76
v1.3.5.	
<code>\normalbfnoteX</code> : Allows one to redefine <code>\thefootnoteX</code> with <code>\alph</code> when some packages are loaded. . . . .	77
v1.4.0.	
General: Added <code>\do@insidelineLhook</code> and <code>\do@insidelineRhook</code> . . . . .	60
v1.4.1.	
<code>\normalbfnoteX</code> : Fix bug with normal familiar footnotes when mixing RTL and LTR text. . . . .	77
General: Enable the use of <code>stanzaindentsrepetition</code> within <code>astanza</code> environment. . . . .	77
v1.4.3.	
<code>\inserthangingsymbolR</code> : Hanging verse is no longer automatically flush right. . . . .	77
<code>\pendL</code> : Spurious spaces in <code>\pendL</code> . . . . .	55
<code>\pendR</code> : Spurious spaces in <code>\pstartR</code> . . . . .	56
<code>\pstartR</code> : Spurious spaces in <code>\pstartL</code> and <code>\pstartR</code> . . . . .	52
General: Corrects a false hanging verse when a verse is exactly the length of a line. . . . .	1
v1.5.0.	
<code>\sublinenumincrement*</code> : Add starred version of <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> to change both Left and Rightside. . . . .	32
General: Add, as in <code>eledmac</code> , 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 <code>eledmac</code> , features to make crossrefs with <code>pstart</code> numbers. . . . .	1
v1.8.0.	
<code>\Columns</code> : Modify <code>\Columns</code> to enable to add section's title. . . . .	85
Suppress <code>\l@dchecklang</code> from <code>\Columns</code> . . . . .	86
<code>\Pages</code> : Modify <code>\Pages</code> to enable to add section's title. . . . .	93
<code>\l@dchecklang</code> : Suppress <code>\l@dchecklang</code> which did not work and was not logical, because both columns could have the same language but not the main language of the document. . . . .	80
<code>\pendL</code> : As in <code>eledmac</code> , <code>\pendL</code> can have an optional argument. . . . .	55
<code>\pendR</code> : As in <code>eledmac</code> , <code>\pendR</code> can have an optional argument. . . . .	56
<code>\print@columnseparator</code> : Move some code of <code>\Columns</code> to <code>\print@columnseparator</code> . . . . .	88
<code>\pstartR</code> : As in <code>eledmac</code> , <code>\pendL</code> and <code>\pendR</code> can have an optional argument. . . . .	52
<code>\sidenotemargin*</code> : <code>\sidenotemargin</code> is now directly defined in <code>eledmac</code> to be able to manage <code>eledpar</code> . . . . .	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> . ....	26
<code>\l@disnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in <code>eledmac</code> . ....	76
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. ....	57
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. ....	80
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) (?). ....	57
<code>\do@lineR</code> : Fix a bug with critical notes at the beginning of a page, added by v1.8.0 (?). ....	60
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> .....	43
<code>\lineation*</code> : Add <code>\lineation*</code> .....	31
<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> . ....	93
<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> .....	29
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). ....	93
Debug wrong parallel pages synchronization when an <code>\edtext</code> falls across two pages. ....	93
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. . . . .	44
General: Compatibility of standard footnotes with some biblatex styles. . . . .	1
v1.12.0.	
\Columns: Add \l@dprintingcolumnstrue . . . . .	85
\Pages: Add \l@dprintingpagestrue . . . . .	93
\edlabel: \edlabel and \edindex works now with hyperref when using eledpar. . .	75
\edlabel is now defined only one time for both eledmac and eledpar . . . . .	75
\print@eledsectionL: Compatibility with Lua $\TeX$ RTL languages. . . . .	59
\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. . . . .	59
v1.13.0.	
\Pages: Prevent false overfull hboxes when using \Pages outside of pages environment.	94
When using shiftedpstarts option, a \l@dleftbox with a null height will advance the	
\pagetotal in any case. . . . .	93
\clearl@drightpage: Use \newpage instead of \clearpage. . . . .	101
\ifledRcol: Remove false boolean settings which are not needed. . . . .	24
General: Enable the use of optional argument of & in astanza 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) . . . . .	93
\correct@footinsX@box: Call \correct@footinsX@box and	
\correct@Xfootins@box directly in \print@notesX@forpages and	
\print@Xnotes@forpages. . . . .	70
Correct \correct@footinsX@box and \correct@Xfootins@box . . . . .	70
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 . . . . .	63
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting	
when a verse is hanging . . . . .	77
\lineationR: As \lineation, \lineationR automatically set the	
\pstartinfootnote. . . . .	30
\n@num: \n@num defined only one time for both Eledmac and Eledpar. . . . .	39
\skipnumbering: \skipnumbering defined only one time for both Eledmac and	
Eledpar . . . . .	44
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> .	44
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>\hidenumbers</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> .	39
<code>\@set</code> : <code>\@set</code> defined only in <code>reledmac</code> .	38
<code>\advanceline</code> : <code>\advanceline</code> defined only in <code>reledmac</code> .	43
<code>\bbl@set@language</code> : Patch <code>\bbl@set@language</code> instead of redefining it	80
<code>\do@lockonR</code> : <code>\do@lockon</code> defined only in <code>reledmac</code> .	38
<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> .	43
<code>\fix@page</code> : <code>\fix@page</code> is defined only once in <code>reledmac</code>	37
<code>\l@d@set</code> : <code>\l@d@set</code> defined only in <code>reledmac</code> .	38
<code>\l@dbfnote</code> : <code>\l@dbfnote</code> defined only in <code>reledmac</code> .	77
<code>\line@marginR</code> : <code>\linenummargin</code> now defined only once time in <code>reledmac</code> .	31
<code>\normalbfnoteX</code> : <code>\normalbfnoteX</code> defined only in <code>reledmac</code> .	77
<code>\page@action</code> : <code>\page@action</code> defined only in <code>reledmac</code> .	38
<code>\read@linelist</code> : <code>\read@linelist</code> is defined only once time in <code>\reledmac</code> .	36
<code>\set@line</code> : <code>\set@line</code> defined only in <code>reledmac</code> .	44
<code>\set@line@action</code> : <code>\set@line@action</code> defined only in <code>reledmac</code> .	38
<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> .	38
<code>\sublinenumincrement*</code> : <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> are now defined only in <code>reledmac</code> .	32
<code>\theledlanguageR</code> : Patch <code>\otherlanguage</code> instead of redefining it.	81
General: <code>\@nl</code> is now defined only in <code>reledmac</code> .	36

\ifbypage@ and \ifbypstart@R defined in eledmac. . . . .	30
Fix some bugs with ‘sameparallelpagenunder’ option. . . . .	1
Many code refactored and moved to reledmac. . . . .	1
Package’s name becomes reledpar. . . . .	1
Totally new implementation of ‘sameparallelpagenunder’ option. . . . .	1
chapterinpages: Deleting the old system of managing parallel chapter, keep only the new one with \patchcmd. . . . .	50
v2.1.0.	
General: Fix bug when using \eledsection and related on right pages when page width is short. . . . .	1
Fix bug when using \pagenumbering with memoir (bug added in v2.0.0). . . . .	1
Fix bug with \setparledgroupnotespacing with the shiftedpstarts option. . . . .	1
Fix incompatibility between optional argument of \pstart and \numberpstarttrue	1
Options to custom empty right page before \Pages. . . . .	1
v2.2.0.	
General: astanza environment can take an optional argument, which will be the optional argument of \pstart started by this environment. . . . .	1
New tools to number stanza . . . . .	1
v2.2.1.	
General: Fix bug with optional argument of last left \pend . . . . .	1
v2.3.0.	
\Pages: Fix bug when calling \Columns after a \Pages (bug added in v1.13.0). . . . .	94
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.	
\ledstrutR: Deleted \ledstrutL and \ledstrutR . . . . .	100
Fix bug with dotted letter . . . . .	100
General: New way of (not) synchronizing the parallel pages. . . . .	1
Option to switch to \mainmatter when calling \Pages . . . . .	1
v2.5.0.	
General: Disable empty lines as paragraph in astanza. . . . .	1
Fix bug introduced in v1.15.0 which made hanging indentation in verse not work anymore. . . . .	1
New commands \linenummarginR and \linenummargin* . . . . .	1
v2.5.1.	
General: Fix spurious space when using optional argument of astanza environment (introduced in v2.5.0). . . . .	1
v2.5.2.	
General: Fix bug introduced in v2.5.0 with \linenummargin, \firstlinenum, \linenumincrement, \firstsublinenum, \sublinenumincrement. . . . .	1
v2.6.0.	
\l@dmake@labelsR: \@Rlineflag is not stored directly after the line number, but as a fifth argument of \the@labelX. Can be retrieved by \xflagref. . . . .	75
General: \Xlineflag and \Xendlineflag added . . . . .	1
\printlinesR deleted . . . . .	1
Error message when calling \Pages or \Columns 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 <code>pages</code> or <code>columns</code> environments. . . . .	1
Fix compatibility with <code>babel</code> (broken in v2.0.0). . . . .	1
No error messages about ends of left / right page when using the <code>\syntaxonly</code> command of the <code>syntonly</code> package. . . . .	1
v2.6.1.	
General: Fix bug, introduced in v2.6.0, with footnote numbering when using <code>perpage</code> package. . . . .	1
v2.6.2.	
<code>\newseries@par</code> : The $\TeX$ counter <code>\footnoteX@reading</code> is defined in <code>reledmac</code> . . .	45
General: Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.6.0). . . . .	1
Fix bug (added in v2.6.0) with footnote numbering in parallel typesetting while using <code>polyglossia</code> with specific numbering systems (like Greek). . . . .	1
v2.6.3.	
General: Fix spurious dot when using <code>\linenummargin</code> on right side (introduced in v2.5.0). . . . .	1
v2.7.0.	
General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package. . . . .	1
v2.7.1.	
General: Fix bug added in <code>reledmac</code> 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 <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> options. . . . .	1
Add <code>\linenumberLevenifblanktrue</code> and <code>\linenumberRevenifblank</code> 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 <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> . . . . .	1
More specific error messages. . . . .	1
v2.9.1.	
General: Prevent <code>\Xtxtbeforenotes</code> hook from causing notes to go beyond the bottom margin . . . . .	1
v2.10.0.	
<code>\do@actions@nextR</code> : Add action 1010 . . . . .	63
General: Add new tools to make apparatuses of manuscripts . . . . .	1