



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

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

## Abstract

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

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

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

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

## Contents

<b>1 Introduction</b>	<b>6</b>
1.1 Aim of this package . . . . .	6
1.2 Historical overview . . . . .	7
<b>2 Options</b>	<b>7</b>
2.1 Synchronization’s options . . . . .	7
2.2 Other options . . . . .	8
<b>3 General</b>	<b>8</b>

---

\*This file (reledpar.dtx) has version number v2.25.2, last revised 2021/09/27.

<sup>†</sup>maieul at maieul dot net

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

<b>4 Parallel columns</b>	<b>9</b>
4.1 Basic use	9
4.2 Setting	10
4.2.1 Column's width	10
4.2.2 Column's separator	10
4.2.3 Column's positions	10
4.2.4 Mixing two columns and one column texts	11
4.2.5 Schemas of column typesetting	11
4.2.6 Continuous numbering between single and double column texts	11
4.3 Note about <code>\AtEveryPstart*</code> and <code>\AtEveryPend*</code>	12
<b>5 Keeping translation in the outside/inside column</b>	<b>12</b>
<b>6 Facing pages</b>	<b>12</b>
6.1 Basic usage	12
6.2 Setting	13
6.2.1 Text width	13
6.2.2 Way of synchronizing	13
6.2.3 Page number	15
6.2.4 Page breaking	15
6.2.5 Right page before <code>\Pages</code>	15
6.2.6 Notes about <code>\mainmatter</code>	15
6.3 Critical and familiar footnotes	15
6.3.1 Notes height setting	16
6.3.2 About the numbering of familiar footnotes	16
6.3.3 Using <code>perpage</code> package	16
6.3.4 Notes for one side only	17
6.3.5 Familiar notes called on the right side, but to be printed on the left side	17
6.4 Critical notes called on leftside but printed on right side	17
6.5 Using line flag	18
<b>7 Left and right texts</b>	<b>18</b>
7.1 Environments	18
7.2 Numbering text lines and paragraphs	18
7.3 First line number and line number increment	19
7.4 Lination system	20
7.5 Line number style	20
7.6 Line number margin	20
7.7 Line flag	21
7.8 Chunks	21
7.9 <code>\AtEveryPstart</code> and <code>\AtEveryPstartCall</code>	22
7.10 Language setting	22
7.11 Executing code at each line	22
7.12 Executing code at each page	22

<b>8 Verse</b>	<b>23</b>
<b>9 Side notes</b>	<b>23</b>
<b>10 Parallel ledgroups</b>	<b>24</b>
10.1 General	24
10.2 Parallel ledgroups and <code>setspace</code> package	25
<b>11 Sectioning commands</b>	<b>25</b>
<b>12 Notes about page number</b>	<b>25</b>
<b>I Implementation overview</b>	<b>26</b>
<b>II Preliminaries</b>	<b>26</b>
II.1 Package's meta-data	26
II.2 Package's requirement	26
II.3 Package's options	26
II.4 Package's options	27
II.4.1 Synchronization's options	27
II.4.2 Position options	28
II.4.3 Other options	28
II.5 Determining side and category of parallel processing	29
II.6 Text's width	29
II.7 Messages	29
II.8 Naming macros	33
<b>III Sectioning commands</b>	<b>34</b>
<b>IV Line counting</b>	<b>38</b>
IV.1 Setting lineation reset	38
IV.2 Setting line number margin	39
IV.3 Setting lineation start and step	41
IV.4 Setting line flag	42
IV.5 Setting line number style	42
IV.6 Print marginal line number	43
IV.7 Line-number counters and lists	44
IV.7.1 Correspond to those in <code>reledmac</code> for regular or left text	44
IV.7.2 Specific to <code>reledpar</code>	45
IV.8 Reading the line-list file	45
IV.9 Commands within the line-list file	46
IV.10 Writing to the line-list file	52

<i>Contents</i>	5
<b>V Marking text for notes</b>	<b>54</b>
V.1 Specific hooks and commands for notes . . . . .	55
V.1.1 Notes to be printed on one side only . . . . .	55
V.1.2 Tools specific to familiar footnotes . . . . .	55
V.1.3 Get correct footnote number . . . . .	57
V.2 Create hooks . . . . .	58
V.3 Init standards series (A,B,C,D,E,Z) . . . . .	58
V.4 Tools specific to $\LaTeX$ 's classical footnotes . . . . .	58
<b>VI Pstart numbers dumping and restoration</b>	<b>58</b>
<b>VII Parallel environments</b>	<b>59</b>
<b>VIII Paragraph decomposition and reassembly</b>	<b>62</b>
VIII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code> . . . . .	62
VIII.2 Processing one line . . . . .	69
VIII.3 Line and page number computation . . . . .	75
VIII.4 Line number printing . . . . .	78
VIII.5 Pstart number printing in side . . . . .	81
VIII.6 Add insertions to the vertical list . . . . .	83
VIII.7 Penalties . . . . .	84
VIII.8 Printing leftover notes . . . . .	85
<b>IX Footnotes</b>	<b>85</b>
IX.1 Footnotes output specific to <code>\Pages</code> . . . . .	85
IX.2 Critical footnote printed in right side . . . . .	88
<b>X Cross referencing</b>	<b>91</b>
<b>XI Sidenotes</b>	<b>91</b>
<b>XII Verse</b>	<b>94</b>
<b>XIII Fixing babel and polyglossia</b>	<b>96</b>
<b>XIV Counts and boxes for parallel texts</b>	<b>99</b>
<b>XV Checking text to be processed</b>	<b>100</b>
<b>XVI Parallel columns</b>	<b>102</b>
<b>XVII Parallel pages</b>	<b>113</b>
XVII.1 Specific counters . . . . .	113
XVII.2 Main macro . . . . .	113
XVII.3 Ensure all notes are printed at the end of parallel pages . . . . .	120
XVII.4 Struts . . . . .	121
XVII.5 Page clearing . . . . .	121
XVII.6 Lines managing . . . . .	123

XVII.7 Page break managing . . . . .	124
XVII.8 Getting boxes content . . . . .	127
<b>XVIII The mainmatter option of \Pages</b>	<b>131</b>
<b>XIX Sections' titles' commands</b>	<b>132</b>
<b>XX Page break/no page break, depending on the specific line</b>	<b>133</b>
<b>XXI Parallel ledgroup</b>	<b>133</b>
<b>XXII Compatibility with eledmac</b>	<b>137</b>
<b>XXIII The End</b>	<b>137</b>
<b>Appendix A Some things to do when changing version</b>	<b>138</b>
A.1 Migration to eledpar 1.4.3 . . . . .	138
A.2 Migration from eledpar to reledpar . . . . .	138
A.2.1 Deprecated options . . . . .	138
A.2.2 \renewcommand replaced with command . . . . .	138
A.2.3 Commands the names of which have changed . . . . .	139
A.3 Migration to reledpar 2.2.0 . . . . .	139
A.4 Migration to reledpar 2.3.0 . . . . .	139
A.5 Migration to reledpar 2.4.0 . . . . .	139
A.6 Migration to reledpar 2.5.0 . . . . .	139
A.7 Migration to reledpar 2.6.0 . . . . .	139
A.8 Migration to reledpar 2.6.1 . . . . .	139
A.9 Migration to reledpar 2.14.0 . . . . .	140
A.10 Migration to reledpar 2.14.1 . . . . .	140
A.11 Migration to reledpar v. 2.17.2 . . . . .	140
A.12 Migration to reledpar v. 2.24.3 . . . . .	140
<b>References</b>	<b>140</b>
<b>Index</b>	<b>140</b>
<b>Change History</b>	<b>160</b>

## 1 Introduction

### 1.1 Aim of this package

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

The package has to try and coerce  $\TeX$  into paths it was not designed for. Use of the package, therefore, may produce some surprising results. In this case, please reports them to the author via github's issues: <https://github.com/maieul/ledmac/issues/>.

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

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

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

## 1.2 Historical overview

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

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

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

## 2 Options

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

### 2.1 Synchronization's options

Please read the paragraph on synchronization's option on 6.2.2 p. 13 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>

**continuousnumberingwithcolumns** allow you to alternate between normal typesetting and parallel column typesetting without resetting the line number (see 4.2.6 p. 11).

**sidenotesmarginpage** makes the sidenotes be placed on the margin of the page and not on the margin of the columns.

**sameparallelpagenumber** 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.

**movecolumnsonrightpage** make the left column on the right page become the right column, and the left column become the right column. It allows to have a text running on the inner column, and an other one on the outer column.

## 3 General

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

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

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

<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`.

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



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 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 7.

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

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

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

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

There is no required pagebreak before or after the columns.

## 4.2 Setting

### 4.2.1 Column's width

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

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

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

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

### 4.2.2 Column's separator

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

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

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

### 4.2.3 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 `\setstanzaindents` outside the `Leftside` or `Rightside` environment.

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

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

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

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

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

#### 4.2.4 Mixing two columns and one column texts

`\widthliketwocolumns`

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

`\Xnoteswidthliketwocolumns`  
`\notesXwidthliketwocolumns`

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

#### 4.2.5 Schemas of column typesetting

Domenico Cufalo wrote a schema of the parallel column typesetting parameters, with explanations. You can read the file `doc-more/page-typesetting-columns.pdf`.

#### 4.2.6 Continuous numbering between single and double column texts

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` in addition to `\beginnumbering... \endnumbering` (see 5.2.7 p. 21).

The best way to understand the behaviour of `\pausenumbering` and `\resumenumbering` is to think of the left-column text as a continuation of single-column text (or, vice versa, to think of single-column text as left-column text), and of the right-column as independent text, which needs to be synchronised with the left one manually. The numbering commands in a right column are interconnected with each other, and not with the respective left-column commands.

This means that you must switch between single-column and left-column text using `\beginnumbering... \pausenumbering... \resumenumbering... \pausenumbering... \resumenumbering... \endnumbering`.

On the other side, the first right column must always begin with `\beginnumbering`, no matter if the corresponding left-column started with `\resumenumbering` or not; this is because numbering in the right column has not been started yet. Then, for consecutive chunks of right-column text you should use `\pausenumbering... \resumenumbering`.

You must use `\endnumbering` in a right column in the following cases:

- there is also a `\endnumbering` in the left column, because the line numbering ends with a two columns mode (e.g. at the end of a document);
- there will be a `\endnumbering` in a text after the columns, in a single column mode, because the line numbering finishes with single column mode text.

Without this, you will get wrong numbering when a right column reappears next time.

The file `examples/4-reledpar_column_mix_with_not_column-continuous-numbering.tex` covers all these cases.

### 4.3 Note about `\AtEveryPstart*` and `\AtEveryPend*`

The content of `\AtEveryPstart` / `\AtEveryPend` (without star) is added before every `\pstart` / after every `\pend`, distinguishing between the left and the right side.

The content of `\AtEveryPstart*` / `\AtEveryPend*` (with star) is added before every `\pstart` / after every `\pend`, but there is no distinction between the left and right sides: the content is inserted “for the two columns as a whole”.

However, as this content is also added when we typeset pages in parallel, `reledmac` must add it once for left `\pstart` and once for right `\pstart`. So if you use it to add vertical spacing, it will be problematic in parallel columns, as you will have the vertical spacing two times (vertical spacing will be twice as large as expected). A solution is to add a test inside to distinguish between parallel typesetting in columns or on pages.

```
\AtEveryPstart*{%
\ifl@dprintingcolumns
  \vspace{0.125\baselineskip}
\else
  \vspace{0.25\baselineskip}
\fi
}
```

## 5 Keeping translation in the outside/inside column

By default, `reledpar` works with left and right columns. However, it is possible to work with inner and outer column. In this case, use the `movecolumnsonrightpage` at loading time of the package.

The `Leftside` environment will correspond to the inner column, and the `Rightside` environment will correspond to the outer column.

## 6 Facing pages

### 6.1 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.

## 6.2 Setting

### 6.2.1 Text width

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

### 6.2.2 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 `advancedshiftpstarts` 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:

- To ensure that left pages contain what is to be on left side and that right pages contain what is to be on right side.
- 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:

---

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

- 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.

### 6.2.3 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 `sameparallelpagewater` 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 12 p. 25.

### 6.2.4 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>}`.

### 6.2.5 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 12 p. 25.

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

### 6.2.6 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.

## 6.3 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.

### 6.3.1 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  $\TeX$ '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 (7.12.6 p. 56). As an advisable setting:

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

### 6.3.2 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.

### 6.3.3 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}
\MakePerPage{footnoteB}
```

But set:

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

---

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



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

### 6.3.4 Notes for one side only

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

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

### 6.3.5 Familiar notes called on the right side, but to be printed on the left side

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

To achieve this, we call `\footnoteXnomk{ $\langle notecontent \rangle$ }` 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}
```

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

Sometimes, you need to print the critical notes on the right side, despite the fact that they refer to the lemmas on the left side. In this case, you must use `\edtextlater` and `\edtextnow`. The `\edtextlater{ $\langle lemma \rangle$ }{ $\langle footnote command \rangle$ }` command must be called on the same side as the lemma. It is similar to the standard `\edtext` command, but does not add the footnote immediately on the page, keeping it for later. `\edtextnow`

`\edtextlater`  
`\edtextnow`

must be called on the side on which you want to print the lemma, approximately at the point corresponding to the equivalent `\edtextlater` command. It will add the notes stored by the equivalent `\edtextlater` command.

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

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

## 6.5 Using line flag

`\Xlineflag` Use `\Xlineflag[⟨s⟩]` to add right line flag (7.7 p. 21) to right critical footnotes and  
`\Xendlineflag` `\Xendlineflag[⟨s⟩]` to add it to right critical endnotes.

# 7 Left and right texts

## 7.1 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.

## 7.2 Numbering text lines and paragraphs

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

```
\beginnumbering
⟨text⟩
\endnumbering
```

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

The `\beginnumbering` macro resets the line number to zero, reads an auxiliary file called `⟨jobname⟩.nn` (where `⟨jobname⟩` is the name of the main input file for this job, and `nn` is 1 for the first numbered section, 2 for the second section, and so on), and then creates a new version of this auxiliary file to collect information during this run. Separate auxiliary files are maintained for right hand texts and these are named `⟨jobname⟩.nnR`, using the “R” to distinguish them from the left hand and serial (non-parallel) texts.

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

```
\begin{pages}
```

```

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

```

`\numberstarttrue`  
`\numberstartfalse` It is possible to insert a number at every `\pstart` command. You must use the `\numberstarttrue` command to have it. You can stop the numbering with `\numberstartfalse`.

You can modify the number by changing the values of the `pstartL` and `pstartR` counters (for left and right side, respectively). For example:

```
\setcounter{pstartL}{0}% To reset the counter of pstart for the left side.
```

`\thepstartL`  
`\thepstartR` You can redefine the commands `\thepstartL` and `\thepstartR` to change style. The numbering restarts on each `\beginnumbering`.

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

`\hidenumbering` 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.

### 7.3 First line number and line number increment

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

`\firstsublinenum`  
`\sublinenumincrement` 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**

```

\firstlinenum*
\linenumincrement*
\firstsublinenum*
\sublinenumincrement*
\firstlinenumR
\linenumincrementR
\firstsublinenumR
\sublinenumincrementR
\lineationR
\lineation*

```

**practice is to add all setting commands to the preamble.** 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.

## 7.4 Lineation system

`\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.

`reledmac` allows you to define a `\linenumberlist` to explicitly define (5.3.2 p. 22) in which line the line number will be printed. In parallel typesetting, this command affects only left side. For right side, you have to define `\linenumberlistR`.

## 7.5 Line number style

```

\linenumberstyleR
\sublinenumberstyleR
\linenumberstyle*
\sublinenumberstyle*

```

`\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.

## 7.6 Line number margin

```

\linenummarginR
\linenummargin*
\linenummarginColumns
\linenummarginColumnsR
\linenummarginColumns*

```

`\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.

Suppose you typeset texts both in parallel pages (or in normal typesetting) and in parallel columns. In this case, your setting of line margin in parallel pages (or in normal typesetting) could be different from your setting of line margin in parallel columns. For example, you could want to have line numbers on the right when your are in parallel pages (or in normal typesetting), but when you are in parallel columns, to have them on the left for the left column and on the right for the right column.

In this case, you can use `\linenummarginColumns`, which overrides the default setting for the left column, `\linenummarginColumnsR` which overrides the default setting for the right column, and `\linenummarginColumns*`, which overrides the default for both left and right columns.

```

\linenumOnlyPagesForColumns
\linenumOnlyPagesForColumnsR

```

Another usual case is when you have text in parallel columns and want to get line numbers only on the outer margin. In this case, you need to get line numbers on the left side of the left column on left pages and on the right side of the right column on the right pages.

Reciprocally, if you want line numbers only in the inner margin, you need to get line numbers on the right side of the right column on left pages and on the left side of the left column on the right pages.

In both cases, you need, for one column, to have line numbers only on one page of a double page.

You can use `\linenumOnlyPagesForColumns` and `\linenumOnlyPagesForColumnsR` to get this result:

```
\linenummargin{left}
\linenummarginR{right}
% To get only on the outer margin
\linenumOnlyPagesForColumns{left}
\linenumOnlyPagesForColumnsR{right}
% To get only on the inner margin
\linenumOnlyPagesForColumns{right}
\linenumOnlyPagesForColumnsR{left}
% To get on the inner or the outer margin
\linenumOnlyPagesForColumns{}
\linenumOnlyPagesForColumnsR{}
```

Note that these settings only apply when we are typesetting columns (as, if we are not typesetting columns, the “inner” and “outer” setting of `\linenummargin` and `\linenummarginR` are enough).

By default, when a blank line is printed on one side, no line number is printed for the sake of synchronizing with the other side, . However, line numbers can be printed for blank lines as well. Use `\linenumberLevenifblanktrue` to enable this on the left side and `\linenumberRevenifblanktrue` for the right side.

## 7.7 Line flag

`\setRlineflag` 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.

## 7.8 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... \pend`. Please read `reledmac`'s handbook (5.2.2 p. 19).

### 7.9 `\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. 20) 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`.

### 7.10 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.

### 7.11 Executing code at each line

`\dolineLhook` `\dolineLhook` and `\dolineRhook` are the equivalent to the `reledmac` `\dolinehook`, respectively for left and right side.

`\doinsidelineLhook` `\doinsidelineLhook` and `\doinsidelineRhook` are the equivalent to the `reledmac` `\doinsidelinehook`, for the left and the right sides respectively.

`\doinsidelineRhook` About these two hooks, read `reledmac`'s handbook (5.6 p. 28).

### 7.12 Executing code at each page

`\Rightpagehook` `\Rightpagehook` and `\Leftpagehook` are executed each time `reledpar` switches to Left / Right page.

`\Leftpagehook` These macro are initially void. You can redefine them using `\renewcommand`.

## 8 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 latter syncs verse by verse, while the environment syncs stanza by stanza.

If you get an error message along the lines of “Missing number, treated as zero `\sza@0@`” it is because you have forgotten to use `\setstanzaindent`s to set the stanza indents.

As `astanza` is a specific type of `\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 `\stanz anumtrue` (9.10 p. 62) 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.

## 9 Side notes

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

The `\sidenotemargin` defines the margin of the sidenote for either left or right side, depending on the current environment.

The `\sidenotemarginR` defines the margin of the sidenote for the right side.

You can use `\sidenotemargin*` to define it for both sides.

When typesetting parallel columns, the margin where a sidenote is placed is one of the margins of the column the sidenote is called. However, you can load `reledpar` with the `sidenotesmarginpage` to use the margin of the page, and not of the column.

## 10 Parallel ledgroups

### 10.1 General

You can also make parallel ledgroups (see the documentation of `reledmac` about ledgroups, 10 p. 63). 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

```

## 10.2 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.*

## 11 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).

## 12 Notes about page number

If you use `sameparallepagenumber` option (6.2.3 p. 15) or `prevpgnotnumbered` option (6.2.5 p. 15), please read the following paragraph if you want to manipulate page numbers manually.

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

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

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

## I Implementation overview

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

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

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

## II Preliminaries

### II.1 Package’s meta-data

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

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledpar}[2021/09/27 v2.25.2 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@this@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 `shiftedpstarts` option 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 `advancedshiftedpstarts` option 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 Position options

`\ifmovecolumnsonrightpage` The `\ifmovecolumnsonrightpage` boolean is set to true if, for a right page, we want the left column become a right column and the right column become a left column.

```

39 \newif\ifmovecolumnsonrightpage%
40 \DeclareOptionX{movecolumnsonrightpage}{%
41   \movecolumnsonrightpagetrue%
42 }%
43 %

```

`if@sidenotesmarginpage` With the `sidenotesmarginpage`, a left sidenote called in a right column will be printed at the left of the page, and a right sidenote called in a left column will be printed at the right of the page.

```

44 \newif\if@sidenotesmarginpage%
45 \DeclareOptionX{sidenotesmarginpage}{%
46   \@sidenotesmarginpagetrue%
47 }%
48 %

```

#### II.4.3 Other options

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

```

49 \DeclareOptionX{parledgroup}{\parledgrouptrue}
50 %

```

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

```

51 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
52 \DeclareOptionX{continuousnumberingwithcolumns}{\
53   continuousnumberingwithcolumnstrue}%
54 %

```

Options related to page numbering. The booleans are defined in `reledmac`.

```

54 \DeclareOptionX{sameparallelpagenumber}{\sameparallelpagenumbertrue}
55 \DeclareOptionX{prevpnotnumbered}{\prevpnotnumberedtrue}
56 %

```

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

```

57 \DeclareOptionX{prevpgstyle}{\gdef\prevpgstyle{#1}}%
58 %

```

```

59 \ProcessOptionsX%
60 %

```

## 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` `\Rcolwidth` The widths of the left and right parallel columns (or pages). The suffixed versions are not manipulated directly by user, but we use them to be certain to have the same `coldwidth` in the `pages` environment and in the `\Pages` command.

```

61 \newdimen\Lcolwidth
62 \Lcolwidth=0.45\textwidth
63 \newdimen\Rcolwidth
64 \Rcolwidth=0.45\textwidth
65 \newdimen\Lcolwidth@pages
66 \newdimen\Rcolwidth@pages
67 %

```

## II.7 Messages

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

```

\reledpar@error 68 \newcommand{\reledpar@error}[2]{\PackageError{reledpar}{#1}{#2}}
69 %

```

```
\reledpar@warning70 \newcommand{\reledpar@warning}[1]{\PackageWarning{reledpar}{#1}}%
71 %
```

```
\led@err@TooManyPstarts72 \newcommand*\led@err@TooManyPstarts}{%
73   \reledpar@error{Too many \string\pstart\space without printing.
74               Some text will be lost}{\@ehc}}
75 %
```

```
\led@err@polyglossiaTooOld76 \newcommand{\led@err@polyglossiaTooOld}{%
77   \reledpar@error{You use the polyglossia package.%
78   \MessageBreak However, reledpar requires version 2020/04/08 v1.49 or
later of the polyglossia package.%
79   \MessageBreak Please update your polyglossia version}{\@ehc}%
80 }%
81 %
```

```
\led@err@BadLeftRightPstarts82 \newcommand*\led@err@BadLeftRightPstarts}[2]{%
83   \reledpar@error{The numbers of left (#1) and right (#2)
84               \string\pstart s do not match}{\@ehc}}
85 %
```

```
\led@err@LeftOnRightPage86 \providebool{syntax@}
\led@err@RightOnLeftPage87 \newcommand*\led@err@LeftOnRightPage}{%
88   \notbool{syntax@}%
89   {\reledpar@error{The left page has ended on a right page}{\@ehc}}%
90   {}%
91 }
92 \newcommand*\led@err@RightOnLeftPage}{%
93   \notbool{syntax@}%
94   {\reledpar@error{The right page has ended on a left page}{\@ehc}}
95   {}%
96 }%
97 %
```

```
\led@err@Leftside@PreviousNotPrinted98 \newcommand*\led@err@Leftside@PreviousNotPrinted}{%
\led@err@Rightside@PreviousNotPrinted99   \reledpar@error{You call a new Leftside environment while the previous
one has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}
100 \newcommand*\led@err@Rightside@PreviousNotPrinted}{%
101   \reledpar@error{You call a new Rightside environment while the previous
one has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}
102 %
```

```

\led@err@Pages@InsideEnv 003 \newcommand*\led@err@Pages@InsideEnv}{%
\led@err@Columns@InsideEnv 004 \reledpar@error{\string\Pages\space must be called *outside* of the `
pages` environment}{\@ehc}}
105 \newcommand*\led@err@Columns@InsideEnv}{%
106 \reledpar@error{\string\Columns\space must be called *outside* of the `
pairs` environment}{\@ehc}}
107 %

\led@err@Pages@WithoutEnv 008 \newcommand*\led@err@Pages@WithoutEnv}{%
\led@err@Columns@WithoutEnv 009 \reledpar@error{\string\Pages\space called without previous `pages`
environment}{\@ehc}}
110 \newcommand*\led@err@Columns@WithoutEnv}{%
111 \reledpar@error{\string\Columns\space called without previous `pairs`
environment}{\@ehc}}
112 %

\led@err@Pages@WithoutRightside 113 \newcommand*\led@err@Pages@WithoutRightside}{%
\led@err@Pages@WithoutLeftside 114 \reledpar@error{\string\Pages\space called without any preceding `
@Columns@WithoutRightside 115 Rightside` environment (or environment without content)}{\@ehc}}%
\led@err@Pages@WithoutLeftside 116 \newcommand*\led@err@Pages@WithoutLeftside}{%
117 \reledpar@error{\string\Pages\space called without any preceding `
@Columns@WithoutLeftside 118 Leftside` environment (or environment without content)}{\@ehc}}%
119 \newcommand*\led@err@Columns@WithoutRightside}{%
120 \reledpar@error{\string\Columns\space called without any preceding `
@Columns@WithoutLeftside 121 Leftside` environment (or environment without content)}{\@ehc}}%
122 %

\led@err@fail@patch@thepage 222 \newcommand*\led@err@fail@patch@thepage}{%
123 \reledpar@error{Fail to patch \string\@thepage\space command.}{\@ehc}}%
124 }%
125 %

\led@err@fail@patch@pagenumbering 26 \newcommand*\led@err@fail@patch@pagenumbering}{%
127 \reledpar@error{Fail to patch \string\pagenumbering\space command.}{\@ehc}
128 }%
129 %

\led@err@note@called@onrightside 30 \newcommand*\led@err@note@called@onrightside}[1]{%
\led@err@note@called@onleftside 31 \reledpar@error{#1 called on right side, despite your configuring it to
be for the left side only}{\@ehc}}%

```

```

132 }%
133 \newcommand{\led@error@note@called@onleftside}[1]{%
134   \reledpar@error{#1 called on left side, despite your configuring it to be
      for the right side only}{\@ehc}%
135 }%
136 %

```

```

\led@error@fail@patch@mempnum137 \newcommand{\led@error@fail@patch@mempnum}{%
138   \reledpar@error{Fail to patch \string@mempnum\space command.}{\@ehc}%
139 }%
140 %

```

```

\led@error@fail@patch@outputpage141 \newcommand{\led@error@fail@patch@outputpage}{%
142   \reledpar@error{Fail to patch \string@outputpage\space command.}{\@ehc}%
143 }%
144 %

```

```

\led@error@edtext@later@now145 \newcommand{\led@error@edtext@later@now}{%
146   \reledpar@error{You call \the\edtext@later\space \string\edtextlater\
      space commands, but \the\edtext@now\space \string\edtextnow\space commands
      }{\@ehc}% (did I follow your intent correctly here?)
147 }%
148 %

```

```

\led@warn@ChangeSyncOption149 \newcommand*\led@warn@ChangeSyncOption[1]{%
150   \reledpar@warning{You have changed synchronization's options since the
      last run. We have not read line-list file #1. Please run LaTeX again.}%
151 }%
152 %

```

```

\led@warn@setting@in@rightside153 \newcommand{\led@warn@setting@in@rightside}[1]{%
154   \reledpar@warning{You use #1 inside rightside environment.\MessageBreak%
155   Such behavior is deprecated.\MessageBreak%
156   Use instead #1R or #1* in your preamble.}%
157 }
158 %

```

```

\led@error@missing@numbering159 \newcommand{\led@error@missing@numbering}[1]{%
160   \reledpar@error{Missing \string\pstart\space ... \space\string\pend\space
      inside `#1` environment}{\@ehc}%
161 }%
162 %

```



## II.8 Naming macros

The L<sup>A</sup>T<sub>E</sub>X 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
163 \providecommand*\newnamebox}[1]{%
164   \expandafter\newbox\csname #1\endcsname}
165 \providecommand*\setnamebox}[1]{%
166   \expandafter\setbox\csname #1\endcsname}
167 \providecommand*\unhnamebox}[1]{%
168   \expandafter\unhbox\csname #1\endcsname}
169 \providecommand*\unvnamebox}[1]{%
170   \expandafter\unvbox\csname #1\endcsname}
171 \providecommand*\namebox}[1]{%
172     \csname #1\endcsname}
173
174 %

```

`\ifcsboxvoid` The `\ifcsboxvoid` command comes on top of the `etoolbox` package. The first argument is a box name, the second is the *⟨true⟩* value, the third the *⟨false⟩* value. If the first argument refers to a non-existing box, or if it refers to a box that is void, the *⟨true⟩* value is returned. otherwise, *⟨false⟩* is returned.

```

175 \newcommand*\ifcsboxvoid}[1]{%
176   \ifcsname#1\endcsname%
177   \expandafter\ifvoid\csname #1\endcsname%
178   \expandafter\expandafter%
179   \expandafter\@firstoftwo%
180   \else%
181   \expandafter\expandafter%
182   \expandafter\@secondoftwo%
183   \fi%
184   \else%
185   \expandafter\@firstoftwo%
186   \fi%
187 }%
188 %

```

`\newnamecount` Macros for creating and using “named” counts.

```

\usenamecount
189 \providecommand*\newnamecount}[1]{%
190   \expandafter\newcount\csname #1\endcsname}
191 \providecommand*\usenamecount}[1]{%
192     \csname #1\endcsname}
193
194 %

```

### III Sectioning commands

`\section@numR` This is the right side equivalent of `\section@num`.

Each section will read and write an associated “line-list file”, containing information used to do the numbering. Normally the file will be called `\jobname.nn`, where `nn` is the section number. However, for right side texts the file is called `\jobname.nnR`. The `\extensionchars` applies to the right side files just as it does to the normal files.

```
195 \newcount\section@numR
196 \section@numR=\z@
197 %
```

`\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`.

```
198 \pst@rtedLfalse
199 \newif\ifpst@rtedR
200
201 %
```

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

```
202 \newcommand*\beginnumberingR}{%
203 \ifnumberingR
204 \led@err@NumberingStarted
205 \endnumberingR
206 \fi
207 \global\l@dnumpstartsR \z@
208 \global\pst@rtedRfalse
209 \global\numberingRtrue
210 \global\afterendnumberingRfalse%
211 \global\advance\section@numR \@ne
212 \global\absline@numR \z@
213 \gdef\normal@page@breakR{}
214 \gdef\l@prev@pbR{}
215 \gdef\l@prev@nopbR{}
216 \global\line@numR \z@
217 \global\@lockR \z@
218 \global\sub@lockR \z@
219 \global\sublines@Rfalse
220 \global\let\next@page@numR\relax
221 \global\let\this@section@next@page@numR\relax%
222 \global\let\sub@change\relax
223 \global\last@page@numR=-10000%
224 \global\stopmsdata@inserted@true%
225 \global\let\@msdata@list\relax%
226 \global\csundef\@msdata@\add@msd@cR @dataR}%
227 \ifcontinuousnumberingwithcolumns%
```

```

228 \ifnumbering%
229   \global\resumenumberingR@starttrue%
230 \fi%
231 \ifnumberpstart%
232   \addtocounter{pstartR}{1}%
233 \fi%
234 \else%
235   \setcounter{pstartR}{1}%
236 \fi%
237 \message{Section \the\section@numR R }%
238 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
239 \l@dend@stuff
240 \beginingroup
241 \initnumbering@sectcountR
242 \gdef\eled@sectionsR@{}%
243 \if@noeled@sec\else%
244   \makeatletter%
245   \InputIfFileExists%
246   {\l@auxdir\jobname.eledsec\the\section@numR R}%
247   {}%
248   {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@numR R}}%
249   \makeatother%
250   \immediate\openout\eled@sectioningR@out=\l@auxdir\jobname.eledsec\the\
section@numR R\relax%
251 \fi%
252 }
253 %

```

`\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.

```

254 \def\endnumberingR{%
255   \ifnumberingR
256     \global\numberingRfalse
257     \global\afterendnumberingRtrue%
258     \normal@pars
259     \ifnum\l@dnumpstartsR=0%
260       \led@err@NumberingWithoutPstart%
261     \fi%
262     \global\page@numR=\this@section@page@numR%
263     \global\last@page@numR=\this@section@last@page@numR%
264     \global\let\next@page@numR\this@section@next@page@numR%
265     \ifl@dpairing
266       \global\pst@rtedRfalse
267     \else
268       \ifx\insertlines@listR\empty\else

```

```

269     \global\noteschanged@true
270     \fi
271     \ifx\line@listR\empty\else
272     \global\noteschanged@true
273     \fi
274     \fi
275     \ifnoteschanged@
276     \led@mess@NotesChanged
277     \fi
278     \else
279     \led@err@NumberingNotStarted
280     \fi
281     \endgroup
282     \if@noeled@sec\else%
283     \immediate\closeout\eled@sectioningR@out%
284     \fi%
285     }
286
287 %

```

`\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`.

```

288 \newcounter{chapterR}
289 \newcounter{sectionR}
290 \newcounter{subsectionR}
291 \newcounter{subsubsectionR}
292
293 \newcount\old@chapter%
294 \newcount\old@section%
295 \newcount\old@subsection%
296 \newcount\old@subsubsection%
297 \newcommand{\save@section@number}{%
298   \ifdefined\c@chapter%
299   \global\old@chapter\value{chapter}%
300   \fi%
301   \global\old@section\value{section}%
302   \global\old@subsection\value{subsection}%
303   \global\old@subsubsection\value{subsubsection}%
304 }%
305 \newcommand{\initnumbering@sectcountR}{
306   \ifdefined\c@chapter%
307   \setcounter{chapterR}{\old@chapter}%
308   \fi%
309   \setcounter{sectionR}{\old@section}%
310   \setcounter{subsectionR}{\old@subsection}%
311   \setcounter{subsubsectionR}{\old@subsubsection}%
312   \set@sectcountR%
313 }

```

```

314 \newcommand{\set@sectcountR}{%
315   \let\c@chapter\c@chapterR%
316   \let\c@section\c@sectionR%
317   \let\c@subsection\c@subsectionR%
318   \let\c@subsubsection\c@subsubsectionR%
319 }%
320 %

```

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

```

\resumenumberingR
\ifresumenumberingR@start
\pausenumbering@page@numR
321 \newcount\pausenumbering@page@numR%
322 \newcommand*{\pausenumberingR}{%
323   \ifx\this@section@next@page@numR\relax%
324     \global\pausenumbering@page@numR=0%
325   \else%
326     \global\pausenumbering@page@numR=\this@section@next@page@numR%
327   \fi%
328   \endnumberingR%
329   \global\numberingRtrue%
330   \global\afterendnumberingRfalse%
331 }%
332 \newif\ifresumenumberingR@start%
333 \newcommand*{\resumenumberingR}{%
334   \ifnumberingR
335     \global\pst@rtedRtrue
336     \global\advance\section@numR \@ne
337     \global\resumenumberingR@starttrue%
338     \led@mess@SectionContinued{\the\section@numR R}%
339     \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
340     \l@dend@stuff
341     \begingroup%
342     \initnumbering@sectcountR%
343   \else
344     \led@err@NumberingShouldHaveStarted
345     \endnumberingR
346     \beginnumberingR
347   \fi}
348 %
349 %

```

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

```

350 \newcommand*{\memorydumpL}{%
351   \endnumbering
352   \numberingtrue
353   \global\pst@rtedLtrue
354   \global\advance\section@num \@ne
355   \led@mess@SectionContinued{\the\section@num}%
356   \line@list@stuff{\jobname.\extensionchars\the\section@num}%

```

```

357 \l@dend@stuff}
358
359 \newcommand*\memorydumpR}{%
360 \endnumberingR
361 \numberingRtrue
362 \global\pst@rtedRtrue
363 \global\advance\section@numR \@ne
364 \led@mess@SectionContinued{\the\section@numR R}%
365 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
366 \l@dend@stuff}
367
368 %

```

## 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`.

```

369 \newcommand*\lineationR}[1]{%
370 \ifnumbering
371 \led@err@LineationInNumbered
372 \else
373 \def\@tempa{#1}\def\@tempb{page}%
374 \ifx\@tempa\@tempb
375 \global\bypage@Rtrue
376 \global\bypstart@Rfalse
377 \unless\ifnocritical@%
378 \Xpstart[] [false]%
379 \fi%
380 \else
381 \def\@tempb{pstart}%
382 \ifx\@tempa\@tempb
383 \global\bypage@Rfalse
384 \global\bypstart@Rtrue
385 \unless\ifnocritical@%
386 \Xpstart%
387 \fi%
388 \else
389 \def\@tempb{section}
390 \ifx\@tempa\@tempb
391 \global\bypage@Rfalse%
392 \global\bypstart@Rfalse%

```

```

393         \unless\ifnocritical@%
394             \Xpstart[] [false]%
395         \fi%
396     \else
397         \led@warn@BadLineation
398     \fi%
399 \fi
400 \fi
401 \fi}}
402 %

```

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

```

403 \WithSuffix\newcommand\lineation*[1]{%
404     \lineation{#1}%
405     \lineationR{#1}%
406 }%
407 %

```

## 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`.

```

408 \newcount\line@marginR
409 %

```

By default put right text numbers at the right.

```

410 \line@marginR=\@one
411
412 %

```

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

```

413 \newcommand{\linenummarginR}[1]{%
414     \l@dgetline@margin{#1}%
415     \ifnum\@l@dtmpcntb>\m@ne%
416         \global\line@marginR=\@l@dtmpcntb%
417     \fi%
418 }
419 \WithSuffix\newcommand\linenummargin*[1]{%

```

```

420 \l@dgetline@margin{#1}%
421 \ifnum\@l@dttempcntb>\m@ne%
422   \global\line@marginR=\@l@dttempcntb%
423   \global\line@margin=\@l@dttempcntb%
424   \fi%
425 }
426 \ifmovecolumnsonrightpage%
427   \linenummargin{inner}%
428   \linenummarginR{outer}%
429 \fi%
430 %

```

`\linenummarginColumns*` The `\linenummarginColumns` macro overrides the definition of the line margin when typesetting in parallel columns.

`\linenummarginColumns`

`\linenummarginColumnsR`

```

431
432 \newcount\line@margin@columnsR%
433 \line@margin@columnsR=\m@ne%
434
435 \newcommand{\linenummarginColumns}[1]{%
436   \l@dgetline@margin{#1}%
437   \ifnum\@l@dttempcntb>\m@ne%
438     \global\line@margin@columns=\@l@dttempcntb%
439     \fi%
440 }%
441
442 \WithSuffix\newcommand\linenummarginColumns*[1]{%
443   \l@dgetline@margin{#1}%
444   \ifnum\@l@dttempcntb>\m@ne%
445     \global\line@margin@columns=\@l@dttempcntb%
446     \global\line@margin@columnsR=\@l@dttempcntb%
447   \fi%
448 }%
449
450 \newcommand{\linenummarginColumnsR}[1]{%
451   \l@dgetline@margin{#1}%
452   \ifnum\@l@dttempcntb>\m@ne%
453     \global\line@margin@columnsR=\@l@dttempcntb%
454   \fi%
455 }%
456
457 %

```

`\linenumOnlyPagesForColumns` `\linenumOnlyPagesForColumns` and `\linenumOnlyPagesForColumnsR` make the line numbers be printed only on left/right page for the left/right column.

`\linenumOnlyPagesForColumnsR`

```

458 \def\linenum@OnlyPages@ForColumns{}%
459 \newcommand{\linenumOnlyPagesForColumns}[1]{%
460   \gdef\linenum@OnlyPages@ForColumns{#1}%
461 }%

```



```

462 \def\linenum@OnlyPages@ForColumnsR{%
463 \newcommand{\linenumOnlyPagesForColumnsR}[1]{%
464 \gdef\linenum@OnlyPages@ForColumnsR{#1}%
465 }%
466 %
467 %

```

### 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.

```

468 \newcounter{firstlinenumR}
469 \setcounter{firstlinenumR}{5}
470 \newcounter{linenumincrementR}
471 \setcounter{linenumincrementR}{5}
472 %

```

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

```

473 \newcounter{firstsublinenumR}
474 \setcounter{firstsublinenumR}{5}
475 \newcounter{sublinenumincrementR}
476 \setcounter{sublinenumincrementR}{5}
477 %
478 %

```

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

```

\firstlinenum
\linenumincrement
\firstsublinenum
\sublinenumincrement
\firstlinenum*
\linenumincrement*
\firstsublinenum*
\sublinenumincrement*
\firstlinenum* [1]{%
\setcounter{firstlinenumR}{#1}%
\setcounter{firstlinenum}{#1}%
\linenumincrement*
}
\firstsublinenum* [1]{%
\setcounter{linenumincrementR}{#1}%
\setcounter{linenumincrement}{#1}%
}
\firstsublinenum* [1]{%
\setcounter{firstsublinenumR}{#1}%
\setcounter{firstsublinenum}{#1}%
}
\sublinenumincrement* [1]{%
\setcounter{sublinenumincrementR}{#1}%
\setcounter{sublinenumincrement}{#1}%

```

```
494 }
495 %
```

```

\firstlinenumR And the “R” suffixed version.
\linenumincrementR
\firstsublinenumR
\sublinenumincrementR
496 \newcommand\firstlinenumR[1]{%
497   \setcounter{firstlinenumR}{#1}%
498 }
499 \newcommand\linenumincrementR[1]{%
500   \setcounter{linenumincrementR}{#1}%
501 }
502 \newcommand\firstsublinenumR[1]{%
503   \setcounter{subfirstlinenumR}{#1}%
504 }
505 \newcommand\sublinenumincrementR[1]{%
506   \setcounter{sublinenumincrementR}{#1}%
507 }
508 %
```

#### IV.4 Setting line flag

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

```
509 \newcommand{\setRlineflag}[1]{%
510   \gdef\Rlineflag{#1}%
511 }
512 \setRlineflag{R}
513 %
```

#### IV.5 Setting line number style

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

```
514 \newcommand*\linenumrepR[1]{\@arabic{#1}}
515 \newcommand*\sublinenumrepR[1]{\@arabic{#1}}
516
517 %
```

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

```
518 \newcommand*\linenumberstyleR[1]{%
519   \def\linenumrepR##1{\@nameuse{@#1}{##1}}
520 \newcommand*\sublinenumberstyleR[1]{%
521   \def\sublinenumrepR##1{\@nameuse{@#1}{##1}}
522 %
```

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

```

523 \WithSuffix\newcommand\linenumberstyle*[1]{%
524   \linenumberstyle{#1}%
525   \linenumberstyleR{#1}%
526 }%
527
528 \WithSuffix\newcommand\sublinenumberstyle*[1]{%
529   \sublinenumberstyle{#1}%
530   \sublinenumberstyleR{#1}%
531 }%
532 %
533 %

```

## IV.6 Print marginal line number

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

```

534 \newif\iflinenumberLevenifblank
535 \newif\iflinenumberRevenifblank
536 %

```

`\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`.

```

537 \newcommand*\leftlinenumR{%
538   \l@dlinenumR%
539   \kern\linenumsep%
540 }%
541 \newcommand*\rightlinenumR{%
542   \kern\linenumsep%
543   \l@dlinenumR%
544 }%
545
546
547 \newcommand*\l@dlinenumR{%
548   \numlabfont%
549   \unless\iflinenumannotationotherside%
550     \ifdefstring{\Xlinenumannotationposition@side}{before}{%
551       \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
552         \csuse{annotR@the\absline@numR @the\section@numR}%
553       }%
554     }%
555   \fi%
556   \linenumrepR{\line@numR}\@Rlineflag%
557   \ifsublines@R
558     \ifnum\subline@numR>\z@
559       \unskip\fullstop\sublinenumrepR{\subline@numR}%
560     \fi

```

```

561 \fi%
562 \unless\iflinenumannotationotherside%
563   \ifdefstring{\Xlinenumannotationposition@side}{after}{%
564     \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
565       \csuse{annot@the\absline@numR @the\section@numR}%
566     }%
567   }%
568 \fi%
569 }%
570
571 %

```

`\leftlinenumannotationR` `\rightlinenumannotationR` and `\leftlinenumannotationR` are the same as not suffixed version, but for right side.

```

572 \newcommand*\leftlinenumannotationR{%
573   \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
574     \csuse{annot@the\absline@numR @the\section@numR}%
575   }%
576   \kern\linenumsep%
577 }%
578 \newcommand*\rightlinenumannotationR{%
579   \kern\linenumsep%
580   \l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{%
581     \csuse{annot@the\absline@numR @the\section@numR}%
582   }%
583 }%
584 %

```

## 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.

```

585 \newcount\line@numR
586 \newcount\subline@numR
587 \newcount\absline@numR
588
589 %

```

`\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:  
`\annot@listR`

```
590 \list@create{\line@listR}
591 \list@create{\insertlines@listR}
592 \list@create{\actionlines@listR}
593 \list@create{\actions@listR}
594 \list@create{\annot@listR}%
595 %
```

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

```
\this@section@page@numR
596 \newcount\page@numR
597 \newcount\this@section@page@numR%
598 %
```

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

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

```
599 \list@create{\linesinpar@listL}
600 \list@create{\linesinpar@listR}
601 \list@create{\maxlinesinpar@list}
602
603 %
```

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

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

```
604 \newcommand{\list@clearing@regR}{%
605   \list@clear{\line@listR}%
606   \list@clear{\insertlines@listR}%
607   \list@clear{\actionlines@listR}%
608   \list@clear{\actions@listR}%
609   \list@clear{\linesinpar@listR}%
610   \list@clear{\linesonpage@listR}
611 }
612 %
```

`\@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.

```
613 \newcommand{\@par@sync@option}[1]{%
614   \IfStrEq{#1}{\@par@this@sync@option}%
```

```

615 {}%
616 {\ifledRcol%
617   \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
618 %
619   \else%
620   \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
621 %
622   \fi%
623   \endinput%
624 }%

```

`\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.

```

625 \newcommand{\@nl@regR}{%
626   \ifx\l@dchset@num\relax \else
627   \advance\absline@numR \@ne
628   \csgdef{l@dchset@numR\the\absline@numR}{}%To remember this line have
629   \set@line@action
630   \let\l@dchset@num\relax
631   \advance\absline@numR \m@ne
632   \advance\line@numR \m@ne%   % do we need this?
633   \fi
634   \reset@current@annot%
635   \advance\absline@numR \@ne
636   \ifx\next@page@numR\relax \else
637   \page@action
638   \let\next@page@numR\relax
639   \fi
640   \ifx\sub@change\relax \else
641   \ifnum\sub@change>\z@
642   \sublines@Rtrue
643   \else
644   \sublines@Rfalse

```

```

645 \fi
646 \sub@action
647 \let\sub@change\relax
648 \fi
649 \ifcase\@lockR
650 \or
651 \@lockR \tw@
652 \or\or
653 \@lockR \z@
654 \fi
655 \ifcase\sub@lockR
656 \or
657 \sub@lockR \tw@
658 \or\or
659 \sub@lockR \z@
660 \fi
661 \ifsublines@R
662 \ifnum\sub@lockR<\tw@
663 \advance\subline@numR \@ne
664 \fi
665 \else
666 \ifnum\@lockR<\tw@
667 \advance\line@numR \@ne \subline@numR \z@
668 \fi
669 \fi}
670
671 %
672 %

```

`\last@page@numR` holds the page number of the last right page. `\this@section@last@page@numR` holds the page number of the last right page of the current section. Its value is modified globally, contrary to `\last@page@numR`. Both are modified by `\fix@page`, defined by `reledmac`.

```

673 \newcount\last@page@numR
674 \last@page@numR=-10000
675 \newcount\this@section@last@page@numR%
676 \this@section@last@page@numR=-10000%
677
678 %

```

- `\@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`.

```

679 \newcount\@lockR
680 \newcount\sub@lockR
681
682 \newcommand*{\do@lockonR}{%
683   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
684   \ifsublines@R
685     \xright@appenditem{-1005}\to\actions@listR
686     \ifnum\sub@lockR=\z@
687       \sub@lockR \@ne
688     \else
689       \ifnum\sub@lockR=\thr@@
690         \sub@lockR \@ne
691       \fi
692     \fi
693   \else
694     \xright@appenditem{-1003}\to\actions@listR
695     \ifnum\@lockR=\z@
696       \@lockR \@ne
697     \else
698       \ifnum\@lockR=\thr@@
699         \@lockR \@ne
700       \fi
701     \fi
702   \fi}
703
704 %

```

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

`\do@lockoffR` `\do@lockoffR`

`\skip@lockoff`

```

705
706

```



```

707 \newcommand{\do@lockoffR}{%
708   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
709   \ifsublines@R
710     \xright@appenditem{-1006}\to\actions@listR
711     \ifnum\sub@lockR=\tw@
712       \sub@lockR \thr@@
713     \else
714       \sub@lockR \z@
715     \fi
716   \else
717     \xright@appenditem{-1004}\to\actions@listR
718     \ifnum\@lockR=\tw@
719       \@lockR \thr@@
720     \else
721       \@lockR \z@
722     \fi
723   \fi}
724
725 %
726 %

```

`\n@num`

`\@ref`

`\@ref@regR`

`\insert@countR`

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

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

```

727 \newcount\insert@countR
728 %

```

- #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.

```

729 \newcommand*{\@ref@regR}[2]{%
730   \global\advance\@edtext@level by 1%
731   \global\insert@countR=#1\relax
732   \loop\ifnum\insert@countR>\z@
733     \xright@appenditem{\the\absline@numR}\to\insertlines@listR
734     \global\advance\insert@countR \m@ne
735   \repeat
736 %

```

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.

```

737 \begingroup
738   \let\@ref=\dummy@ref
739   \let\@lopR\@gobble
740   \let\page@action=\relax
741   \let\sub@action=\relax
742   \let\set@line@action=\relax
743   \let\@lab=\relax
744   \let\@lemma=\relax
745   \let\@sw\@gobblethree%
746   \let\store@annot@to@absline\@gobble%
747   #2
748   \global\endpage@num=\page@numR
749   \global\endline@num=\line@numR
750   \global\endsubline@num=\subline@numR
751   \global\let\endcurrent@annot=\current@annot%
752 \endgroup
753 %

```

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

```

754 \xright@appenditem%
755   {\the\page@numR|\the\line@numR|%
756   \ifsublinesR \the\subline@numR \else 0\fi|%
757   \the\endpage@num|\the\endline@num|%
758   \ifsublinesR \the\endsubline@num \else 0\fi}\to\line@listR
759 \xright@appenditem%
760   {\current@annot|\endcurrent@annot}\to\annot@listR%
761 %

```

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

```

762 \expandafter\list@create\expandafter{\cscname sw@list@edtext@tmp@\the\
@edtext@level\endcscname}%
763 %

```

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

```

764 \@ref@reg@parseR{#2}%
765 %

```

Decrease edtext level counter.

```

766 \global\advance\@edtext@level by -1%
767 }
768 %

```

```

\@ref@reg@parseR69 \newcommand{\@ref@reg@parseR}[1]{%
770 %

Declare and init boolean for lemma in this level.

771     \providebool{lemmacommand@the\@edtext@level}%
772     \boolfalse{lemmacommand@the\@edtext@level}%
773 %

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

774     #1%
775 % Now, we store the list of \protect\cs{@sw} of this current \protect\cs{
edtext} as an element of
776 % the global list of list of \protect\cs{@sw} for a \protect\cs{edtext}
depth.
777 % \begin{macrocode}
778     \ifnum\@edtext@level>0%
779     \def\create@this@edtext@level{\expandafter\list@create\expandafter{
csname sw@list@edtextR@the\@edtext@level\endcsname}}%
780     \ifcsundef{sw@list@edtextR@the\@edtext@level}{\
create@this@edtext@level}{}%
781     \letcs{\@tmp}{sw@list@edtextR@the\@edtext@level}%
782     \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}%
783     \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
784     \global\cslet{sw@list@edtextR@the\@edtext@level}{\@tmp}%
785     \fi%
786 }%
787 %

```

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

```

788 \renewcommand*{\@pend}[1]{%
789     \ifbypstart@%
790     \unless\ifinstanza%
791     \global\line@num=0%
792     \fi%
793     \fi%
794     \xright@appenditem{#1}\to\linesinpar@listL%
795 }%
796 \renewcommand*{\@pendR}[1]{%
797     \ifbypstartR%
798     \unless\ifinstanzaR%
799     \global\line@numR=0%
800     \fi%
801     \fi%
802     \xright@appenditem{#1}\to\linesinpar@listR%
803 }%

```

```

804
805 %
\@pstart \@pstart and cs@pstartR allows us to know, when using \nomaxlines option, in
\@pstartR 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
806 \newcommand{\@pstart}[3]{%
807   \ifcsdef{minpage@pstart@#1}%
808     {\ifnumgreater{#2}{\csuse{minpage@pstart@#1}}%
809       {\csnumgdef{minpage@pstart@#1}{#2}}%
810       {}}%
811   }%
812   {\csnumgdef{minpage@pstart@#1}{#2}}
813   \csnumgdef{afterlines@pstart@#1L}{#3}%
814   }%
815
816 \newcommand{\@pstartR}[3]{%
817   \numdef{\@tmp}{#2-1}%Because we have not to know in which page the pstart
818   starts, but in which pair of facing page
819   \ifcsdef{minpage@pstart@#1}%
820     {\ifnumgreater{\@tmp}{\csuse{minpage@pstart@#1}}%
821       {\csnumgdef{minpage@pstart@#1}{\@tmp}}%
822       {}}%
823     {\csnumgdef{minpage@pstart@#1}{\@tmp}}
824     \csnumgdef{afterlines@pstart@#1R}{#3}%
825   }%
826   %

```

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

```

827 \renewcommand*{\@lopL}[1]{%
828   \xright@appenditem{#1}\to\linesonpage@listL}
829 \renewcommand*{\@lopR}[1]{%
830   \xright@appenditem{#1}\to\linesonpage@listR}
831
832 %

```

## 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`.

```
833 \newwrite\linenum@outR
834 %
```

`\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
835 \newif\iffirst@linenum@out@R
836 \first@linenum@out@Rtrue
837 %
```

`\line@list@stuffR` This is the right text version of the `\line@list@stuff{<file>}` macro. It is called by `\next@line@list@stuffR` and performs all the line-list operations needed at the start of a section. Its argument is the name of the line-list file. `reledmac` and `reledpar` can fill the `\next@line@list@stuffR` hook between a `\endnumberingR` (associated with numbered file  $n$ ) and a `\beginnumberingR` (associated with numbered file  $n + 1$ ). It allows adding content to the numbered file  $n + 1R$  and not  $nR$ .

```
838
839 \let\next@line@list@stuffR\relax%
840 \newcommand*{\line@list@stuffR}[1]{%
841 \global\newtoggle{notfirstrun@#1}%
842 \IfFileExists{\l@auxdir#1}%
843 {\global\toggletrue{notfirstrun@#1}}%
844 {\global\togglefalse{notfirstrun@#1}}%
845 \next@line@list@stuffR%
846 \global\let\next@line@list@stuffR\relax%
847 \read@linelist{#1}%
848 \iffirst@linenum@out@R
849 \global\first@linenum@out@Rfalse
850 \immediate\openout\linenum@outR=\l@auxdir#1%
851 \immediate\write\linenum@outR{\string\line@list@version{\
this@line@list@version}}%
852 \ifl@dpaging%
853 \immediate\write\linenum@outR{\string\@par@sync@option{\
@par@this@sync@option}}%
854 \fi%
855 \else
856 \if@minipage%
857 \leavevmode%
858 \fi%
859 \closeout\linenum@outR%
860 \openout\linenum@outR=\l@auxdir#1%
861 \fi%
862 }%
863
864 %
```

`\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.

```

865 \newcommand*{\new@lineL}{%
866   \ifnumberline%
867     \write\linenum@out{\string\@nl [\the\@this@c@page] [\thepage]}%
868   \fi%
869 }%
870 %

```

`\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.

```

871 \newcommand*{\new@lineR}{%
872   \ifnumberline%
873     \write\linenum@outR{\string\@nl [\the\@this@c@page] [\thepage]}%
874   \fi%
875 }%
876 %

```

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

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

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

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

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

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

`\skipnumbering`

## V Marking text for notes

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

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

## V.1 Specific hooks and commands for notes

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

```
\newseries@par877 \newcommand{\newseries@par}[1]{%
878 %
```

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

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

```
879 \unless\ifnofamiliar@%
880 \csgdef{onlysideX@#1}{}%
881 \newnamebox{footins#1@kept}%
882 \fi%
883 \unless\ifnocritical@%
884 \csgdef{Xonlyside@#1}{}%
885 \newnamebox{#1footins@kept}%
886 \fi%
887 %
```

### V.1.2 Tools specific to familiar footnotes

```
888 \unless\ifnofamiliar@%
889 %
```

**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 directly used in the footnote, and is stored in the aux files to be used on the next run for the main text.

So here, we only defined the new counter.

```
890 \newcounter{footnote#1@typeset}%
891 %
```

**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.

```
892 \expandafter\list@create\csname footnote#1@mk\endcsname%
893 %
```

Then, declare the `\footnoteXnomk` command.

```
894 \expandafter\newcommand\csname footnote#1nomk\endcsname[1]{%
895 %
```

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

```
896 \toggletrue{nomk@}%
897 \csuse{footnote#1}{##1}%
898 \togglefalse{nomk@}%
899 %
```

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`.

```
900 \letcs{\@tmp}{footnote#1@mk}%
901 \numdef\@tmpa{\csuse{c@footnote#1}}%
902 \global\xright@appenditem{\@tmpa}\to\@tmp%
903 \global\cslet{footnote#1@mk}{\@tmp}%
904 }%
905 %
```

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

```
906 \expandafter\newcommand\csname footnote#1mk\endcsname{%
907 %
```

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

```
908 \letcs{\@tmp}{footnote#1@mk}%
909 \gl@p\@tmp\to\@tmpa%
910 \global\cslet{footnote#1@mk}{\@tmp}%
911 %
```



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

```

912     \letcs{\old@footnote}{c@footnote#1}%
913     \setcounter{footnote#1}{\@tmpa}%
914 %

```

Define the footnote mark and print it

```

915     \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
916     \csuse{@footnotemark#1}%
917 %

```

Restore previous footnote counter and finally add space.

```

918     \setcounter{footnote#1}{\old@footnote}%
919     \xspace%
920 }%
921 %

```

End of tools specific to familiar notes.

```

922 \fi
923 %

```

End of \newseries@par.

```

924 }%
925 %

```

### V.1.3 Get correct footnote number

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

```

926 \newcommand{\save@familiarfootnote@number}{%
927   \unless\ifnofamiliar%
928     \def\do##1{\csxdef{saved@footnote##1}{\the\csname c@footnote##1\
endcsname}}%
929     \dolistloop{\@series}%
930     \fi%
931     \xdef\saved@footnote{\the\c@footnote}%
932 }
933 \newcommand{\get@familiarfootnote@number}{%
934   \unless\ifnofamiliar%
935     \def\do##1{\setcounter{footnote##1@typeset}{\csuse{saved@footnote##1}}%
%
936     \dolistloop{\@series}%
937     \fi%
938     \setcounter{footnote@typeset}{\saved@footnote}%
939 }
940 %

```

## V.2 Create hooks

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

```

941 \unless\ifnocritical@%
942   \newhookcommand@series{Xonlyside}%
943 \fi%
944 \unless\ifnofamiliar@%
945   \newhookcommand@series{onlysideX}%
946 \fi
947
948
949 %

```

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

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

```

950 \newcommand{\init@series@par}{%
951   \def\do##1{\newseries@par{##1}}%
952   \dolistloop{\@series}%
953 }%
954 \init@series@par%
955 %

```

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

As users can use classical footnotes of  $\text{\LaTeX}$  (`\footnote`) in parallel texts, we must integrate the same tools to get correct number as for `reledmac`' footnotes (V.1.2 p. 55).

```

\footnote@reading56 \newcount\footnote@reading%
\footnote@typeset57 \newcounter{footnote@typeset}%
58 %

```

## 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 booleans `\ifnumberpstart` and `\ifnumberline`.

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}%
961 %

```

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%
964 \global\cslet{numberpstart@L\the\l@dnumpstartsL}{\ifnumberpstart}%
965 \global\cslet{numberline@L\the\l@dnumpstartsL}{\ifnumberline}%
966 \global\cslet{linenumannotationotherside@L\the\l@dnumpstartsL}{\
iflinenumannotationotherside}%
967 }%
968
969 \def\dump@pstartR@pc{%
970 \xright@appenditem{\the\c@pstartR}\to\list@pstartR@pc%
971 \global\cslet{numberpstart@R\the\l@dnumpstartsR}{\ifnumberpstart}%
972 \global\cslet{numberline@R\the\l@dnumpstartsR}{\ifnumberline}%
973 \global\cslet{linenumannotationotherside@R\the\l@dnumpstartsR}{\
iflinenumannotationotherside}%
974 }%
975
976 %

```

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

```

\restore@pstartR@pc
977 \def\restore@pstartL@pc{%
978 \ifx\list@pstartL@pc\empty\else%
979 \gl@p\list@pstartL@pc\to\@temp%
980 \global\c@pstartL=\@temp%
981 \fi%
982 }%
983 \def\restore@pstartR@pc{%
984 \ifx\list@pstartR@pc\empty\else%
985 \gl@p\list@pstartR@pc\to\@temp%
986 \global\c@pstartR=\@temp%
987 \fi%
988 }%
989 %

```

## 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.

```

990 \newenvironment{pairs}{%
991   \l@dpairingtrue
992   \l@dpagingfalse
993   \initnumbering@quote
994   \save@familiarfootnote@number%
995   \if@ledgroup%
996     \get@familiarfootnote@number%
997   \fi%
998   \save@section@number%
999   \at@begin@pairs%
1000 }{%
1001   \l@dpairingfalse
1002 }
1003
1004 %

```

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

```

1005 \newcommand{\AtBeginPairs}[1]{\gdef\at@begin@pairs{#1}}%
1006 \def\at@begin@pairs{}%
1007
1008 %

```

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.

```

1009 \newenvironment{pages}{%
1010   \l@dpairingtrue
1011   \l@dpagingtrue
1012   \initnumbering@quote
1013   \save@familiarfootnote@number%
1014   \if@ledgroup%
1015     \get@familiarfootnote@number%
1016   \fi%
1017   \save@section@number%
1018   \Lcolwidth=\textwidth%
1019   \Rcolwidth=\textwidth%
1020 }{%
1021   \l@dpairingfalse
1022   \l@dpagingfalse
1023   \global\Lcolwidth@pages=\Lcolwidth%Does not use \setlength, in case of
calc package redefiniton (cf. https://github.com/latex3/latex2e/issues/58)
1024   \global\Rcolwidth@pages=\Rcolwidth%
1025 }
1026
1027 %

```

**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.

```

1028 \newenvironment{Leftside}{%
1029   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname\else%
1030   \led@err@Leftside@PreviousNotPrinted%
1031   \fi%
1032   \ledRcolfalse
1033   \setcounter{pstartL}{1}
1034   \let\pstart\pstartL
1035   \let\thepstart\thepstartL
1036   \let\pend\pendL
1037   \let\memorydump\memorydumpL
1038   \Leftsidehook
1039   \let\old@startstanza\@startstanza%
1040   \def\@startstanza[##1][##2]{\global\instanzaLtrue\old@startstanza
1041     [##1][##2]}%
1042 }{
1043   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname%
1044   \led@error@missing@numbering{Leftside}%
1045   \fi%
1046   \Leftsidehookend}
%
```

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

```

\Leftsidehookend
\Rightsidehook
\Rightsidehookend
1047 \newcommand*\Leftsidehook{}
1048 \newcommand*\Leftsidehookend{}
1049 \newcommand*\Rightsidehook{}
1050 \newcommand*\Rightsidehookend{}
1051
1052 %
```

`\Rightpagehook` Hook at the start of Left/Right page, initially empty.

```

\Leftpagehook
1053 \newcommand*\Rightpagehook{}%
1054 \newcommand*\Leftpagehook{}%
1055 %
```

**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.

```

1056 \newenvironment{Rightside}{%
1057   \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else%
1058   \led@err@Rightside@PreviousNotPrinted%
1059   \fi%
```

```

1060 \ledRcoltrue
1061 \let\beginnumbering\beginnumberingR
1062 \let\endnumbering\endnumberingR
1063 \let\pausenumbering\pausenumberingR
1064 \let\resumenumbering\resumenumberingR
1065 \let\memorydump\memorydumpR
1066 \let\thepstart\thepstartR
1067 \let\pstart\pstartR
1068 \let\pend\pendR
1069 \let\ledpb\ledpbR
1070 \let\lednopb\lednopbR
1071 \let\lineation\lineationR
1072 \Rightsidehook
1073 \let\old@startstanza\@startstanza%
1074 \def\@startstanza[##1][##2]{\global\instanzaRtrue\old@startstanza
1075 [##1][##2]}%
1076 \ledRcolfalse
1077 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
1078 \led@error@missing@numbering{Rightside}%
1079 \fi%
1080 \Rightsidehookend
1081 }
1082
1083 %

```

## 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.

```

1084 \newcount\num@linesR

```

```

1085 \newbox\one@lineR
1086 \newcount\par@lineR
1087 %

```

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

```

1088
1089 \newcounter{pstartL}
1090 \renewcommand{\thepstartL}{\bfseries\@arabic\c@pstartL}. }
1091 \newcounter{pstartR}
1092 \renewcommand{\thepstartR}{\bfseries\@arabic\c@pstartR}. }
1093
1094 \newcommand*{\pstartL}[2][1,2,usedefault]{%
1095   \if@nobreak%
1096     \let\@oldnobreak\@nobreaktrue%
1097   \else%
1098     \let\@oldnobreak\@nobreakfalse%
1099   \fi%
1100   \@nobreaktrue%
1101   \ifluatex%
1102     \xdef\l@luatextextdir@L{\the\textdir}%
1103     \xdef\l@luatexpardir@L{\the\pardir}%
1104     \xdef\l@luatexbodydir@L{\the\bodydir}%
1105   \fi%
1106   \ifnumbering \else%
1107     \led@err@PstartNotNumbered%
1108     \beginnumbering%
1109   \fi%
1110   \ifnumberedpar@%
1111     \led@err@PstartInPstart%
1112     \pend%
1113   \fi%
1114 %

```

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

```

1115 \ifpst@rtedL\else%
1116   \list@clear{\inserts@list}%
1117   \global\let\next@insert=\empty%
1118   \global\pst@rtedLtrue%
1119 \fi%
1120 \begingroup\everypar{}%

```

```

1121 %
    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.
1122 \global\advance\l@dnumpstartsL \@ne%
1123 \global\advance\pstarts@read@L \@ne%
1124 \ifnum\l@dnumpstartsL>\l@dc@maxchunks%
1125   \led@err@TooManyPstarts%
1126   \global\l@dnumpstartsL=\l@dc@maxchunks%
1127 \fi%
1128 \global\setnamebox{l@dc@l@rawbox\the\l@dnumpstartsL}=\vbox\bgroup%
1129 %

```

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.

```

1130 \l@dzeropenalties%
1131 \ifboolexpr{%
1132   bool{autopar}%
1133   and bool{by@autopar}%
1134 }%
1135 {}%
1136 {%
1137   \ifnumberpstart%
1138   \ifsidepstartnum%
1139   \else%
1140   \thepstartL%
1141   \fi%
1142 \fi%
1143 }%m
1144 \@at@start@every@pstart%
1145 \hsize=\Lcolwidth%
1146 \numberedpar@true%
1147 \iflabelpstart%
1148   \protected@edef\@currentlabel{\p@pstartL\thepstartL}%
1149 \fi%
1150 %

```

Dump the optional arguments

```

1151 \providetoggle{before@pstartL@the\l@dnumpstartsL @par}%
1152 \ifboolexpr{%
1153   test {\ifstrempy{#1}}%
1154   and test {\ifstrempy{#2}}%
1155 }%
1156 {%
1157   \ifat@every@pstart@star@%
1158   \global\togglefalse{before@pstartL@the\l@dnumpstartsL @par}%
1159 \else%
1160   \global\toggletrue{before@pstartL@the\l@dnumpstartsL @par}%

```



```

1161 \fi%
1162 \csgdef{before@pstartL@the\l@dnumpstartsL}{\at@every@pstart}%
1163 }%
1164 {%
1165 \ifstrempy{#1}{}%
1166 \global\toggletrue{before@pstartL@the\l@dnumpstartsL @par}%
1167 \csgdef{before@pstartL@the\l@dnumpstartsL}{\noindent#1}%
1168 }%
1169 \ifstrempy{#2}{}%
1170 {%
1171 \csgdef{before@pstartL@the\l@dnumpstartsL}{#2}%
1172 \global\togglefalse{before@pstartL@the\l@dnumpstartsL @par}%
1173 }%
1174 }%
1175 \at@every@pstart@call%
1176 %

```

Let \by@autopar to FALSE.

```

1177 \global\by@autoparfalse%
1178 %

```

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

```

1179 \ignorespaces%
1180 }
1181 %

```

The same for right side.

```

1182 \newcommandx*{\pstartR}[2][1,2,usedefault]{%
1183 \if@nobraek%
1184 \let\@oldnobraek\@nobraektrue%
1185 \else%
1186 \let\@oldnobraek\@nobraekfalse%
1187 \fi%
1188 \@nobraektrue%
1189 \ifluatex%
1190 \xdef\l@luatextextdir@R{\the\textdir}%
1191 \xdef\l@luatexpardir@R{\the\pardir}%
1192 \xdef\l@luatexbodydir@R{\the\bodydir}%
1193 \fi%
1194 \ifnumberingR \else%
1195 \led@err@PstartNotNumbered%
1196 \beginnumberingR%
1197 \fi%
1198 \ifnumberedpar@%
1199 \led@err@PstartInPstart%
1200 \pendR%
1201 \fi%
1202 \ifpstart@rtedR\else%
1203 \list@clear{\inserts@listR}%

```

```

1204 \global\let\next@insertR=\empty%
1205 \global\pstrtdRtrue%
1206 \fi%
1207 \begingroup\everypar{}%
1208 \global\advance\l@dnumstartsR \@ne%
1209 \ifnum\l@dnumstartsR>\l@dc@maxchunks%
1210 \led@err@TooManyPstarts%
1211 \global\l@dnumstartsR=\l@dc@maxchunks%
1212 \fi%
1213 \global\setnamebox{\l@dc@colrawbox\the\l@dnumstartsR}=\vbox\bgroup%
1214 \l@dzeropenalties%
1215 \ifboolexpr{%
1216   bool{autopar}%
1217   and bool{by@autopar}%
1218 }%
1219 {}%
1220 {%
1221   \ifnumberpstart%
1222   \ifsidepstartnum\else%
1223     \thepstartR%
1224   \fi%
1225 \fi%
1226 }%
1227 \@at@start@every@pstart%
1228 \hsize=\Rcolwidth%
1229 \numberedpar@true%
1230 \iflabelpstart%
1231   \protected@edef\@currentlabel{\p@pstartR\thepstartR}%
1232 \fi%
1233 \providetoggle{before@pstartR@the\l@dnumstartsR @par}%
1234 \ifboolexpr{%
1235   test {\ifstrempy{#1}}%
1236   and test {\ifstrempy{#2}}%
1237 }%
1238 {%
1239   \ifat@every@pstart@star@%
1240   \global\togglefalse{before@pstartR@the\l@dnumstartsR @par}%
1241   \else%
1242     \global\toggletrue{before@pstartR@the\l@dnumstartsR @par}%
1243   \fi%
1244   \csgdef{before@pstartR@the\l@dnumstartsR}{\at@every@pstart}%
1245 }%
1246 {%
1247   \ifstrempy{#1}{}%
1248     \global\toggletrue{before@pstartR@the\l@dnumstartsR @par}%
1249     \csgdef{before@pstartR@the\l@dnumstartsR}{\noindent#1}%
1250   }%
1251   \ifstrempy{#2}{}%
1252   {%
1253     \csgdef{before@pstartR@the\l@dnumstartsR}{#2}%

```

```

1254     \global\togglefalse{before@pstartR@the\l@dnumpstartsR @par}%
1255     }%
1256   }%
1257   \at@every@pstart@call%
1258   \global\by@autoparfalse%
1259   \ignorespaces%
1260 }
1261 %

```

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

```

1262 \newcommand*{\pendL}[2][1,2,usedefault]{%
1263   \ifnumbering \else%
1264     \led@err@PendNotNumbered%
1265   \fi%
1266   \ifnumberedpar@ \else%
1267     \led@err@PendNoPstart%
1268   \fi%
1269 %

```

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.

```

1270   \at@end@every@pend%
1271   \endgraf\global\num@lines=\prevgraf\egroup%
1272   \global\par@line=0%
1273   \get@firsthalf@antilabe@wd%
1274 %

```

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

```

1275   \endgroup%
1276   \ignorespaces%
1277   \@oldnobreak%
1278   \dump@pstartL@pc%
1279   \ifnumberpstart%
1280     \addtocounter{pstartL}{1}%
1281     \ifcontinuousnumberingwithcolumns%
1282       \addtocounter{pstart}{1}%
1283     \fi%
1284   \fi
1285   \parledgroup@beforenotes@save{L}%
1286 %

```

Dump content of the optional argument.

```

1287   \providetoggle{after@pendL@the\l@dnumpstartsL @par}%
1288   \ifboolexpr{%
1289     test {\ifstrempy{#1}}%
1290     and test {\ifstrempy{#2}}%
1291   }%

```

```

1292  {%
1293    \ifat@every@pend@star@%
1294    \global\togglefalse{after@pendL@the\l@dnumpstartsL @par}%
1295    \else%
1296    \global\toggletrue{after@pendL@the\l@dnumpstartsL @par}%
1297    \fi%
1298    \csgdef{after@pendL@the\l@dnumpstartsL}{\at@every@pend}%
1299  }%
1300  {%
1301    \ifstrempy{#1}{}%
1302    \global\toggletrue{after@pendL@the\l@dnumpstartsL @par}%
1303    \csgdef{after@pendL@the\l@dnumpstartsL}{\noindent#1}%
1304  }%
1305    \ifstrempy{#2}{}%
1306    {%
1307      \csgdef{after@pendL@the\l@dnumpstartsL}{#2}%
1308      \global\togglefalse{after@pendL@the\l@dnumpstartsL @par}%
1309    }%
1310  }%
1311 }%
1312 %

```

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

```

1313 \newcommand*{\pendR}[2][1,2,usedefault]{%
1314   \ifnumberingR \else%
1315     \led@err@PendNotNumbered%
1316   \fi%
1317   \ifnumberedpar@ \else%
1318     \led@err@PendNoPstart%
1319   \fi%
1320   \@at@end@every@pend%
1321   \endgraf\global\num@linesR=\prevgraf\egroup%
1322   \global\par@lineR=0%
1323   \get@firshalf@antilabe@wd%
1324   \endgroup%
1325   \ignorespaces%
1326   \@oldnobreak%
1327   \dump@pstartR@pc%
1328   \ifnumberpstart%
1329     \addtocounter{pstartR}{1}%
1330   \fi%
1331   \parledgroup@beforenotes@save{R}%
1332   \providetoggle{after@pendR@the\l@dnumpstartsR @par}%
1333   \ifboolexpr{%
1334     test {\ifstrempy{#1}}%
1335     and test {\ifstrempy{#2}}%
1336   }%
1337   {%
1338     \ifat@every@pend@star@%

```

```

1339     \global\togglefalse{after@pendR@the\l@dumpstartsR @par}%
1340     \else%
1341     \global\toggletrue{after@pendR@the\l@dumpstartsR @par}%
1342     \fi%
1343     \csgdef{after@pendR@the\l@dumpstartsR}{\at@every@pend}}%
1344     {%
1345     \ifstrempy{#1}{}%
1346     \csgdef{after@pendR@the\l@dumpstartsR}{\noindent#1}%
1347     \global\toggletrue{after@pendR@the\l@dumpstartsR @par}%
1348     }%
1349     \ifstrempy{#2}{}%
1350     \csgdef{after@pendR@the\l@dumpstartsR}{#2}%
1351     \global\togglefalse{after@pendR@the\l@dumpstartsR @par}%
1352     }%
1353   }%
1354 }
1355 %
1356 %

```

`\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.

```

1357 \newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
1358 \gdef\at@every@pstart@call{}%
1359 %

```

`\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`

```

1360 \newif\ifprint@last@after@pendL%
1361 \newif\ifprint@last@after@pendR%
1362 %

```

## VIII.2 Processing one line

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

`\l@dleftbox` A line of left text will be put in the box `\l@dleftbox`, and analogously for a line of right text.

`\l@drightbox`

```

1363 \newbox\l@dleftbox
1364 \newbox\l@drightbox
1365
1366 %

```

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

`\countRline`

```

1367 \newcount\countLline
1368 \countLline \z@
1369 \newcount\countRline
1370 \countRline \z@
1371
1372 %

```

`\@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 synchronisation).  
`\@donereallinesR`  
`\@donetotallinesR`

```

1373 \newcount\@donereallinesL
1374 \newcount\@donetotallinesL
1375 \newcount\@donereallinesR
1376 \newcount\@donetotallinesR
1377
1378 %

```

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

```

1379 \newcommand*\do@lineL{%
1380 \letcs{\ifnumberpstart}{numberpstart@L\the\l@dpscl}%
1381 \letcs{\ifnumberline}{numberline@L\the\l@dpscl}%
1382 \letcs{\iflinenumannotationotherside}{linenumannotationotherside@L\the\l@dpscl}%
1383 \advance\countLline \@ne%
1384 \ifvbox\namebox{1@dLcolrawbox\the\l@dpscl}%
1385 {\vbadness=10000%
1386 \splittopskip=\z@%
1387 \do@lineLhook%
1388 \l@demptyd@ta%
1389 \global\setbox\one@line=\vsplit\namebox{1@dLcolrawbox\the\l@dpscl}%
1390 to\baselineskip}%
1391 \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\parledgroup@notes@startL}{}%
1392 \unvbox\one@line \global\setbox\one@line=\lastbox%
1393 \@writepageofparL%
1394 \getline@numL%
1395 \ifnum\@lock>\@ne%
1396 \inserthangingsymboltrue%
1397 \else%
1398 \inserthangingsymbolfalse%
1399 \fi%
1400 \setbox\l@dleftbox%
1401 \hb@xt@ \Lcolwidth{%
1402 \ifl@dhidenumber%
1403 \global\l@dhidenumberfalse%
1404 \f@x@l@cks%
1405 \else%

```

```

1406     \affixline@num%
1407     \fi%
1408     \xifinlist{\the\pstarts@typeset@L}{\eled@sections@}%
1409     {\unless\ifshiftedpstarts%
1410       \add@apparatusL%
1411       \fi%
1412     }%
1413     {\print@lineL}%
1414   }%
1415   \add@penaltiesL%
1416   \global\advance\@donereallinesL\@ne%
1417   \global\advance\@donetotallinesL\@ne%
1418   \else%
1419     \iflinenumberLevenifblank
1420     \setbox\l@dleftbox \hb@xt@ \Lcolwidth{%
1421       \new@lineL%
1422       \l@demptyd@ta%
1423       \getline@numL%
1424       \affixline@num%
1425       \l@dld@ta%
1426       \hspace*{\Lcolwidth}%
1427       \ledrfill\l@drd@ta%
1428     }%
1429   \else%
1430     \setbox\l@dleftbox \hb@xt@ \Lcolwidth{%
1431       \ifcontinuousnumberingwithcolumns%
1432         \new@lineL%
1433         \getline@numL%
1434         \fi%
1435         \hspace*{\Lcolwidth}%
1436       }%
1437     \fi%
1438     \global\advance\@donetotallinesL\@ne%
1439   \fi%
1440 }%
1441
1442
1443 %

```

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

```

1444 \def\print@lineL{%
1445   \affixpstart@numL%
1446   \l@dld@ta%
1447   \add@apparatusL%
1448   \l@dlsn@te%
1449   \do@insidelineLhook%
1450   \csuse{insidethis@\the\absline@num @\the\section@num}%
1451   \global\csundef{insidethis@\the\absline@num @\the\section@num}%

```

```

1452 \hb@xt@ \Lcolwidth{\ledllfill\hb@xt@ \wd\one@line{%
1453 \ifluatex%
1454 \textdir\l@luatexttextdir@L%
1455 \fi%
1456 \new@lineL%
1457 \inserthangingsymbolL%
1458 \l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta%
1459 \l@drsn@te}}%
1460
1461 %

```

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

```

1462 \def\print@eledsectionL{%
1463 \disable@edindex%
1464 \addtocounter{pstartL}{-1}%
1465 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{%
1466 \ifdefstring{\@eledsectmark}{L}{\ledsectnomark}%
1467 \numdef{\temp@}{\pstarts@typeset@L-1}%
1468 \xifinlist{\temp@}{\eled@sections@@}{\@nbreaktrue}{\@nbreakfalse}%
1469 \@eled@sectioningtrue%
1470 \bgroup%
1471 \ifluatex%
1472 \textdir\l@luatexttextdir@L%
1473 \pardir\l@luatexpardir@L%
1474 \bodydir\l@luatexbodydir@L%
1475 \ifdefstring{\l@luatexttextdir@L}{TRT}{\@RTLtrue}{%
1476 \fi%
1477 \csuse{eled@sectioning@the\pstarts@typeset@L}%
1478 \egroup%
1479 \@eled@sectioningfalse%
1480 \global\csundef{eled@sectioning@the\pstarts@typeset@L}%
1481 \ifRTL%
1482 \hspace{-3\paperwidth}%
1483 {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1484 \else%
1485 \hspace{3\paperwidth}%
1486 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1487 \fi%
1488 \vskip\eledsection@correcting@skip%
1489 \restore@edindex%
1490 }
1491
1492 %

```

`\add@apparatusL` The `\add@apparatusL` macro adds apparatus for the left lines, apparatus is both footnotes and sidenotes.

```

1493 \newcommand{\add@apparatusL}{%
1494 \if@firstlineofpage%

```



```

1495 \set@Xtxtbeforenotes%
1496 \set@txtbeforenotesX%
1497 \global\@firstlineofpagefalse%
1498 \fi%
1499 \ifdefstring{\ms@data@position}{msdata-regular}{%
1500 \insert@msdata%
1501 \add@inserts%
1502 \add@Xgroupbyline%
1503 }{%
1504 \add@inserts%
1505 \add@Xgroupbyline%
1506 \insert@msdata%
1507 }%
1508 \affixside@note%
1509 }%
1510 %

```

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

`\dolineRhook`

```

\doinlinedlineLhook
\doinlinedlineRhook
1511 \newcommand*\dolineLhook}[1]{\gdef\do@lineLhook{#1}}%
1512 \newcommand*\dolineRhook}[1]{\gdef\do@lineRhook{#1}}%
1513 \newcommand*\doinlinedlineLhook}[1]{\gdef\do@insidelineLhook{#1}}%
1514 \newcommand*\doinlinedlineRhook}[1]{\gdef\do@insidelineRhook{#1}}%
1515
1516 %

```

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

`\do@lineRhook`

```

\doinlinedlineLhook
\doinlinedlineRhook
1517 \newcommand*\do@lineLhook{}
1518 \newcommand*\do@lineRhook{}
1519 \newcommand*\do@insidelineLhook{}
1520 \newcommand*\do@insidelineRhook{}
1521
1522 %

```

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

```

1523 \newcommand*\do@lineR}{%
1524 \let\linenumrepL\linenumrep%
1525 \let\sublinenumrepL\sublinenumrep%
1526 \let\linenumrepR\linenumrepR%
1527 \let\sublinenumrepR\sublinenumrepR%
1528 \letcs{\ifnumberpstart}{numberpstart@R\the\l@dpscr}%
1529 \letcs{\ifnumberline}{numberline@R\the\l@dpscr}%
1530 \letcs{\iflinenumannotationotherside}{linenumannotationotherside@R\the\l@dpscr}%
1531 \ledRcol@true%
1532 \advance\countRline \@ne%

```

```

1533 \ifvbox\namebox{l@dRcolrawbox\the\l@dpscR}%
1534 {\vbadness=10000%
1535 \splittopskip=\z@%
1536 \do@lineRhook%
1537 \l@emptyd@ta%
1538 \global\setbox\one@lineR=\vsplit\namebox{l@dRcolrawbox\the\l@dpscR}%
1539 to\baselineskip}%
1540 \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startR}{}%
1541 \unvbox\one@lineR \global\setbox\one@lineR=\lastbox%
1542 \@writepageofparR%
1543 \getline@numR%
1544 \ifnum\@lockR>\@ne%
1545 \inserthangingsymbolRtrue%
1546 \else%
1547 \inserthangingsymbolRfalse%
1548 \fi%
1549 \setbox\l@drightbox%
1550 \hb@xt@ \Rcolwidth{%
1551 \ifl@dhiddenumber%
1552 \global\l@dhiddenumberfalse%
1553 \f@x@l@cksR%
1554 \else%
1555 \affixline@numR%
1556 \fi%
1557 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
1558 {\unless\ifshiftedpstarts%
1559 \add@apparatusR%
1560 \fi%
1561 }%
1562 {\print@lineR}%
1563 }%
1564 \add@penaltiesR%
1565 \global\advance\@donereallinesR\@ne%
1566 \global\advance\@donetotallinesR\@ne%
1567 \else%
1568 \iflinenumberRevenifblank%
1569 \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1570 \new@lineR%
1571 \l@emptyd@ta%
1572 \getline@numR%
1573 \affixline@numR%
1574 \l@dld@ta%
1575 \hspace*{\Rcolwidth}%
1576 \ledrlfill\l@drd@ta%
1577 }%
1578 \else%
1579 \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1580 \ifcontinuousnumberingwithcolumns%
1581 \new@lineR%

```

```

1582     \getline@numR%
1583     \fi%
1584     \hspace*{\Rcolwidth}%
1585   }%
1586   \fi%
1587   \global\advance\@donetotallinesR\@ne%
1588 \fi%
1589 \ledRcol@false%
1590 \let\linenumrep\linenumrepL%
1591 \let\sublinenumrep\sublinenumrepL%
1592 }
1593
1594
1595 %

```

```

\print@lineR
\print@eledsectionR
\add@apparatusR

```

The `\add@apparatusR` macro adds apparatus for the right lines, apparatus is both footnotes and sidenotes.

```

1596 \newcommand{\add@apparatusR}{%
1597 \if@firstlineofpageR%
1598   \set@Xtxtbeforenotes%
1599   \set@txtbeforenotesX%
1600   \global\@firstlineofpageRfalse%
1601 \fi%
1602 \ifdefstring{\ms@data@position}{msdata-regular}{%
1603   \insert@msdata%
1604   \add@insertsR%
1605   \add@Xgroupbyline%
1606 }{%
1607   \add@insertsR%
1608   \add@Xgroupbyline%
1609   \insert@msdata%
1610 }%
1611 \affixside@noteR%
1612 }%
1613 %

```

### 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.

```

1614 \newcommand*{\getline@numR}{%
1615 \global\advance\absline@numR \@ne
1616 \do@actionsR
1617 \do@ballastR
1618 \ifledgroupnotesR\else
1619   \ifnumberline

```

```

1620     \ifsublines@R
1621         \ifnum\sub@lockR<\tw@
1622             \global\advance\subline@numR \@ne
1623         \fi
1624     \else
1625         \ifnum\@lockR<\tw@
1626             \global\advance\line@numR \@ne
1627             \global\subline@numR \z@
1628         \fi
1629     \fi
1630 \fi
1631 \fi
1632 }
1633 \newcommand*\getline@numL}{%
1634     \global\advance\absline@num \@ne
1635     \do@actions
1636     \do@ballast
1637     \ifledgroupnotesL@\else
1638         \ifnumberline
1639             \ifsublines@
1640                 \ifnum\sub@lock<\tw@
1641                     \global\advance\subline@num \@ne
1642                 \fi
1643             \else
1644                 \ifnum\@lock<\tw@
1645                     \global\advance\line@num \@ne
1646                     \global\subline@num \z@
1647                 \fi
1648             \fi
1649         \fi
1650     \fi
1651 }
1652
1653
1654 %

```

`\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.

```

1655 \newcommand*\do@ballastR}{\global\ballast@count=\z@
1656     \begingroup
1657         \advance\absline@numR \@ne
1658         \ifnum\next@actionlineR=\absline@numR
1659             \ifnum\next@actionR>-1001
1660                 \global\advance\ballast@count by -\c@ballast
1661             \fi
1662         \fi
1663     \endgroup}
1664 %

```

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

`\do@actions@nextR`

```

1665
1666 \newif\ifl@dskipversenumberR
1667 \newcommand*{\do@actions@fixedcodeR}{%
1668   \ifcase\@l@dtmpcnta%
1669   \or%                % 1001 = starting sublineation
1670   \global\sublines@Rtrue
1671   \or%                % 1002 = ending sublineation
1672   \global\sublines@Rfalse
1673   \or%                % 1003 = starting locking number
1674   \global\@lockR=\@ne
1675   \or%                % 1004 = ending locking number
1676   \ifnum\@lockR=\tw@
1677     \global\@lockR=\thr@@
1678   \else
1679     \global\@lockR=\z@
1680   \fi
1681   \or%                % 1005 = starting locking subnumber
1682   \global\sub@lockR=\@ne
1683   \or%                % 1006 = ending locking subnumber
1684   \ifnum\sub@lockR=\tw@
1685     \global\sub@lockR=\thr@@
1686   \else
1687     \global\sub@lockR=\z@
1688   \fi
1689   \or%                % 1007 = skipping numbering
1690   \l@dskipnumbertrue
1691   \or%                % 1008 = skipping numbering in stanza
1692   \l@dskipversenumberRtrue%
1693   \or%                % 1009 = hiding number
1694   \l@dhidnumbertrue%
1695   \or%                % 1010 = inserting msdata
1696   \add@msdata%
1697   \else%
1698     \led@warn@BadAction
1699   \fi%
1700 }
1701
1702
1703 \newcommand*{\do@actionsR}{%
1704   \global\let\do@actions@nextR=\relax
1705   \@l@dtmpcntb=\absline@numR
1706   \ifnum\@l@dtmpcntb<\next@actionlineR\else
1707     \ifnum\next@actionR>-1001\relax
1708     \ifboolexpr{%
1709       bool{resumenumberingR@start}%

```

```

1710     and test {\ifdimgreater{\pagedepth}{\z@}}%
1711   }%
1712   {}%
1713   {\@firstlineofpageRtrue}%
1714   \global\page@numR=\next@actionR
1715   \ifcsdef{reset@lineR\the\absline@numR @\the\section@numR}%
1716     {%
1717       \global\line@numR=\z@ \global\subline@numR=\z@%
1718     }%
1719     {}%
1720     \global\resumenumberingR@startfalse%
1721     \add@msdata@firstlineofpage%
1722   \else
1723     \ifnum\next@actionR<-4999\relax%
1724       \@l@dttempcnta=-\next@actionR
1725       \advance\@l@dttempcnta by -5001\relax
1726       \ifsublines@R
1727         \global\subline@numR=\@l@dttempcnta
1728       \else
1729         \global\line@numR=\@l@dttempcnta
1730       \fi
1731     \else
1732       \@l@dttempcnta=-\next@actionR
1733       \advance\@l@dttempcnta by -1000\relax
1734       \do@actions@fixedcodeR
1735     \fi
1736   \fi
1737   \ifx\actionlines@listR\empty
1738     \gdef\next@actionlineR{1000000}%
1739   \else
1740     \glp\actionlines@listR\to\next@actionlineR
1741     \glp\actions@listR\to\next@actionR
1742     \global\let\do@actions@nextR=\do@actionsR
1743   \fi
1744   \fi
1745   \do@actions@nextR}
1746
1747 %

```

#### VIII.4 Line number printing

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

```

1748 \ch@cksub@l@ckR
1749 \ch@ck@l@ckR \newcommand*{\l@dcalcnum}[3]{%
1750 \f@x@l@cksR \ifnum #1 > #2\relax
1751 \affixline@numR \@l@dttempcnta = #1\relax
1752 \advance\@l@dttempcnta by -#2\relax
1753 \divide\@l@dttempcnta by #3\relax

```

```

1754 \multiply\@l@tempcnta by #3\relax
1755 \advance\@l@tempcnta by #2\relax
1756 \else
1757 \@l@tempcnta=#2\relax
1758 \fi}
1759
1760 \newcommand*\ch@cksub@l@ckR}{%
1761 \ifcase\sub@lockR
1762 \or
1763 \ifnum\sublock@disp=\@ne
1764 \@l@tempcntb \z@ \@l@tempcnta \@ne
1765 \fi
1766 \or
1767 \ifnum\sublock@disp=\tw@
1768 \else
1769 \@l@tempcntb \z@ \@l@tempcnta \@ne
1770 \fi
1771 \or
1772 \ifnum\sublock@disp=\z@
1773 \@l@tempcntb \z@ \@l@tempcnta \@ne
1774 \fi
1775 \fi}
1776
1777 \newcommand*\ch@ck@l@ckR}{%
1778 \ifcase\@lockR
1779 \or
1780 \ifnum\lock@disp=\@ne
1781 \@l@tempcntb \z@ \@l@tempcnta \@ne
1782 \fi
1783 \or
1784 \ifnum\lock@disp=\tw@
1785 \else
1786 \@l@tempcntb \z@ \@l@tempcnta \@ne
1787 \fi
1788 \or
1789 \ifnum\lock@disp=\z@
1790 \@l@tempcntb \z@ \@l@tempcnta \@ne
1791 \fi
1792 \fi}
1793
1794 \newcommand*\f@x@l@cksR}{%
1795 \ifcase\@lockR
1796 \or
1797 \global\@lockR \tw@
1798 \or \or
1799 \global\@lockR \z@
1800 \fi
1801 \ifcase\sub@lockR
1802 \or
1803 \global\sub@lockR \tw@

```

```

1804 \or \or
1805 \global\sub@lockR \z@
1806 \fi}
1807
1808
1809 \let\linenumberlistR\empty%
1810 \newcommand*{\affixline@numR}{%
1811 \ifledgroupnotesR@else\ifnumberline
1812 \ifl@dskipnumber
1813 \global\l@dskipnumberfalse
1814 \else
1815 \ifsublines@R
1816 \@l@tempcntb=\subline@numR
1817 \l@dcalcnm{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR
}
1818 \ch@cksub@l@ckR
1819 \else
1820 \@l@tempcntb=\line@numR
1821 \ifx\linenumberlistR\empty%
1822 \l@dcalcnm{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%
1823 \else
1824 \@l@tempcnta=\line@numR
1825 \edef\rem@inderR{,\linenumberlistR,\number\line@numR,}%
1826 \edef\sc@n@list{\def\noexpand\sc@n@list
1827 ###1,\number\@l@tempcnta,###2|{\def\noexpand\rem@inderR{###2}}}%
%
1828 \sc@n@list\expandafter\sc@n@list\rem@inderR|
1829 \ifx\rem@inderR\empty\advance\@l@tempcnta\@ne\fi%
1830 \fi
1831 \ch@ck@l@ckR
1832 \fi
1833 \ifnum\@l@tempcnta=\@l@tempcntb
1834 \ifl@dskipversenumberR\else
1835 \if@twocolumn
1836 \if@firstcolumn
1837 \gdef\l@dld@ta{\llap{\leftlinenumR}}%
1838 \iflinenumannotationotherside%
1839 \gdef\l@drd@ta{\rlap{\rightlinenumannotationR}}%
1840 \fi%
1841 \else
1842 \gdef\l@drd@ta{\rlap{\rightlinenumR}}%
1843 \iflinenumannotationotherside%
1844 \gdef\l@dld@ta{\llap{\leftlinenumannotationR}}%
1845 \fi%
1846 \fi
1847 \else
1848 \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columnsR}{\m@ne}}}%
1849 {\@l@tempcntb=\line@margin@columnsR}%
1850 {\@l@tempcntb=\line@marginR}%

```



```

1851 \ifnum\@l@dttempcntb>\@ne
1852 \advance\@l@dttempcntb by\page@numR
1853 \fi
1854 \ifboolexpr{%
1855   bool {l@dprintingcolumns}%
1856   and (%
1857     (test {\ifdefstring{\linenum@OnlyPages@ForColumnsR}{left}}%
1858     and test {\ifnumodd{\page@numR}}%
1859     )%
1860     or%
1861     (test {\ifdefstring{\linenum@OnlyPages@ForColumnsR}{right}}%
1862     and not test {\ifnumodd{\page@numR}}%
1863     )%
1864   )%
1865 }%
1866 {}%
1867 {%
1868   \ifodd\@l@dttempcntb%
1869   \gdef\l@drd@ta{\rlap{\rightlinenumR}}%
1870   \iflinenumannotationotherside%
1871   \gdef\l@dld@ta{\llap{\leftlinenumannotationR}}%
1872   \fi%
1873   \else%
1874   \gdef\l@dld@ta{\llap{\leftlinenumR}}%
1875   \iflinenumannotationotherside%
1876   \gdef\l@drd@ta{\rlap{\rightlinenumannotationR}}%
1877   \fi%
1878   \fi%
1879 }%
1880 \fi
1881 \fi
1882 \fi
1883 \f@x@l@cksR
1884 \fi
1885 \fi
1886 \fi}
1887 %

```

## 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@numL88
\affixpstart@numR
\leftpstartnumR
\rightpstartnumR
\leftpstartnumL
\rightpstartnumL
\ifpstartnumR

```

```

1889 \newcommand*\affixpstart@numL}{%
1890 \ifsidepstartnum
1891 \if@twocolumn
1892   \if@firstcolumn
1893     \gdef\l@dld@ta{\llap{\leftpstartnumL}}%
1894   \else
1895     \gdef\l@dld@ta{\rlap{\rightpstartnumL}}%
1896   \fi
1897 \else
1898   \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columns}{\m@ne}}}%
1899     {\@l@tempcntb=\line@margin@columns}%
1900     {\@l@tempcntb=\line@margin}%
1901   \ifnum\@l@tempcntb>\@ne
1902     \advance\@l@tempcntb \page@num
1903   \fi
1904   \ifodd\@l@tempcntb
1905     \gdef\l@dld@ta{\rlap{\rightpstartnumL}}%
1906   \else
1907     \gdef\l@dld@ta{\llap{\leftpstartnumL}}%
1908   \fi
1909 \fi
1910 \fi
1911 }
1912 \newcommand*\affixpstart@numR}{%
1913 \ifsidepstartnum
1914 \if@twocolumn
1915   \if@firstcolumn
1916     \gdef\l@dld@ta{\llap{\leftpstartnumR}}%
1917   \else
1918     \gdef\l@dld@ta{\rlap{\rightpstartnumR}}%
1919   \fi
1920 \else
1921   \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columnsR}{\m@ne}}}%
1922     {\@l@tempcntb=\line@margin@columnsR}%
1923     {\@l@tempcntb=\line@marginR}%
1924   \ifnum\@l@tempcntb>\@ne
1925     \advance\@l@tempcntb \page@numR
1926   \fi
1927   \ifodd\@l@tempcntb
1928     \gdef\l@dld@ta{\rlap{\rightpstartnumR}}%
1929   \else
1930     \gdef\l@dld@ta{\llap{\leftpstartnumR}}%
1931   \fi
1932 \fi
1933 \fi
1934 }
1935
1936 \newcommand*\leftpstartnumL}{

```

```

1937 \ifpstartnum
1938 \thepstartL
1939 \kern\linenumsep\global\pstartnumfalse\fi
1940 }
1941 \newcommand*{\rightpstartnumL}{
1942 \ifpstartnum\kern\linenumsep
1943 \thepstartL
1944 \global\pstartnumfalse\fi
1945 }
1946 \newif\ifpstartnumR
1947 \pstartnumRtrue
1948 \newcommand*{\leftpstartnumR}{
1949 \ifpstartnumR
1950 \thepstartR
1951 \kern\linenumsep\global\pstartnumRfalse\fi
1952 }
1953 \newcommand*{\rightpstartnumR}{
1954 \ifpstartnumR\kern\linenumsep
1955 \thepstartR
1956 \global\pstartnumRfalse\fi
1957 }
1958 %

```

## 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.

```

1959 \list@create{\inserts@listR}
1960 %

```

`\add@insertsR` The right text version.

```

\add@inserts@nextR
1961 \newcommand*{\add@insertsR}{%
1962 \global\let\add@inserts@nextR=\relax
1963 \ifx\inserts@listR\empty \else
1964 \ifx\next@insertR\empty
1965 \ifx\insertlines@listR\empty
1966 \global\noteschanged@true
1967 \gdef\next@insertR{100000}%
1968 \else
1969 \gl@p\insertlines@listR\to\next@insertR
1970 \fi
1971 \fi
1972 \ifnum\next@insertR=\absline@numR
1973 \gl@p\inserts@listR\to\@insertR
1974 \@insertR
1975 \global\let\@insertR=\undefined
1976 \global\let\next@insertR=\empty

```

```

1977 \global\let\add@inserts@nextR=\add@insertsR
1978 \fi
1979 \fi
1980 \add@inserts@nextR}
1981
1982 %

```

### 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@tempcnta` is used to calculate and accumulate the penalty; it is initially set to the value of `\ballast@count`, which has been worked out in `\do@ballast`. Finally, the penalty is checked to see that it does not go below  $-10000$ .

```

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

```

This is for a single chunk. However, as we are probably dealing with several chunks at a time, the above is not really relevant. Peter Wilson thinks that it is likely with parallel

text that there is no real need to add back any penalties; even if there was, they would have to match across the left and right lines. So, Peter Wilson ends up with the following.

```
1983 \newcommand*{\add@penaltiesL}{-}
1984 \newcommand*{\add@penaltiesR}{-}
1985
1986 %
```

## 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.

```
1987 \newcommand*{\flush@notesR}{%
1988   \iftoggle{notfirststrun@jobname.\extensionchars\the\section@numR R}{%
1989     \xloop%
1990       \ifx\inserts@listR\empty \else%
1991         \gl@p\inserts@listR\to\@insertR%
1992         \@insertR%
1993         \global\let\@insertR=\undefined%
1994       \repeat%
1995     }{}%
1996   }%
1997
1998
1999 %
```

## IX Footnotes

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

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

- If notes are to be printed on both sides, we just proceed the usual way: print the foot starts for the series, then the foot group.
- If notes are to be printed in the left side, we do these prints only for even pages ; if notes are to be printed in the right side, we do these prints only for odd pages.
- However, that is not enough. Because the problem does not only consists in printing notes in any particular page. It is also not to put aside room for notes in the pages where we do not want to print them. To take an example: if some note in the left side is too long by 160pt to be printed in full in the left page, we do not want to put aside 160pt a space for it in the following right page.

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

- To solve this problem, we change the magnification factor associated with notes before going to the next page. If we start a page where no notes are supposed to be printed, the magnification counter is set to 0. The dimension associated to footnote is set to `\maxdimen`, and so we can keep all the notes we want, without any break inside. We also set the note skip to 0pt. Before starting a new page where these notes are supposed to be printed, we reset these counter and skip to their default values. (About these counter, dimension and skip, read *The TeXbook* p. 122-125).
- In the output macro of the page where notes must NOT be printed, we store the content of the footnote box produced by TeX to a temporary box.
- After going to the next page, before typesetting anything in this page, we put the content of this temporary box the footnote insert box.

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

```
2000 \newcommand\print@Xnotes@forpages[1]{%
2001 %
```

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

```
2002   \ifcseempty{Xonlyside@#1}{%
2003     \csuse{#1footstart}{#1}%
2004     \csuse{#1footgroup}{#1}%
2005   }%
2006 %
```

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

```
2007   {%
2008     \ifboolexpr{%
2009       ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page
2010       }})%
2011       or%
2012       (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}})%
2013     }%
2014 %
```

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

```
2014   {%
2015     \csuse{#1footstart}{#1}%
2016     \csuse{#1footgroup}{#1}%
2017 %
```

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

```
2018     \global\count\csuse{#1footins}=\z@%
2019     \global\skip\csuse{#1footins}=\z@%
2020     \global\dimen\csuse{#1footins}=\maxdimen%
2021   }%
2022 %
```

In case we are on a page where notes must NOT be printed. We reset the settings of the notes for one side.

```

2023     {%
2024     \restore@Xnotes@settings{#1}%
2025     }%
2026 %

```

End of \print@Xnotes@forpages.

```

2027     }%
2028 }%
2029 %

```

The macro which restore the note settings. #1 = series

```

2030 \newcommand{\restore@Xnotes@settings}[1]{%
2031 \unless\ifnocritical@
2032 \global\count\csuse{#1footins}=\csuse{default@#1footins}%
2033 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
2034 \bgroup%
2035 \csuse{Xnotefontsize@#1}%
2036 \global\dimen\csuse{#1footins}=\csuse{Xmaxhnotes@#1}%
2037 \egroup%
2038 \global\setnamebox{#1footins@kept}=\box\namebox{#1footins}%
2039 \fi
2040 }%
2041 %

```

And now, the same for familiar footnotes.

```

2042 \newcommand\print@notesX@forpages[1]{%
2043 \ifcsemt{onlysideX@#1}{%
2044 \csuse{footstart#1}{#1}%
2045 \csuse{footgroup#1}{#1}%
2046 }%
2047 {%
2048 \ifboolexpr{%
2049 ((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page
2050 }})%
2051 or%
2052 (test {\ifcsstring{onlysideX@#1}{R}} and test{\ifnumodd{\c@page}})%
2053 }%
2054 {%
2055 \csuse{footstart#1}{#1}%
2056 \csuse{footgroup#1}{#1}%
2057 \global\count\csuse{footins#1}=\z@%
2058 \global\skip\csuse{footins#1}=\z@%
2059 \global\dimen\csuse{footins#1}=\maxdimen%
2060 }%
2061 {%
2062 \restore@notesX@settings{#1}%
2063 }%

```

```

2063 }%
2064 }%
2065 \newcommand{\restore@notesX@settings}[1]{%
2066   \unless\ifnofamiliar@
2067     \global\count\csuse{footins#1}=\csuse{default@footins#1}%
2068     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
2069     \bgroup%
2070     \csuse{Xnotefontsize@#1}%
2071     \global\dimen\csuse{footins#1}=\csuse{maxhnotesX@#1}%
2072     \egroup%
2073     \global\setnamebox{footins#1@kept}=\box\namebox{footins#1}%
2074   \fi
2075 }%
2076 %

```

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

```

2077 \newcommand{\insert@notes@for@onlyside}{%
2078   \def\do##1{%
2079     \unless\ifnocritical@%
2080     \ifvoid\csuse{##1footins@kept}\else%
2081       \expandafter\insert\csname ##1footins\endcsname%
2082       \bgroup%
2083       \unvnamebox{##1footins@kept}%
2084     \egroup%
2085     \fi%
2086     \restore@Xnotes@settings{##1}%
2087   \fi%
2088   \unless\ifnofamiliar@%
2089   \ifvoid\csuse{footins##1@kept}\else%
2090     \expandafter\insert\csname footins##1\endcsname%
2091     \bgroup%
2092     \unvnamebox{footins##1@kept}%
2093   \egroup%
2094   \fi%
2095   \restore@notesX@settings{##1}%
2096 \fi%
2097 }%
2098 \dolistloop{\@series}%
2099 }%
2100 %

```

## IX.2 Critical footnote printed in right side

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



```

2101 \newcount\edtext@now%
2102 \newcount\edtext@later%
2103 %

2104 \newcommand{\edtextlater}[2]{%#1 lemma, #2 critical notes
2105 %

First, increase the \edtext@later counter.

2106 \global\advance\edtext@later by \@ne%
2107 %

2108 % As we are in a pseudo-\cs{edtext}, we now need to get the \cs{sameword}
data stored in the auxiliary file for this \cs{edtextlater}.
2109 \advance\@edtext@level by \@ne%
2110 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
2111     {\global\let\sw@inthisedtext\empty}%
2112     {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
2113 %

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

2114 %
2115 \csxdef{edtext@later@the\edtext@later}{%
2116 %

\edtextnow will insert a empty \edtext.

2117     \noexpand\edtext{%
2118     }%
2119 %

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

2120     {%
2121     \unexpanded{%
2122     \lemma{\noexpands #1}%
2123     }%
2124     \noexpand\xxref{start:edtext:later:\the\edtext@later}{end:edtext:
later:\the\edtext@later}%
2125     \noexpand\linenum{||||\edfont@info}%
2126     \unexpanded{\def\sw@inthisedtext{\expandonce{\sw@inthisedtext}}}%
2127 %

As the \edtextnow is generally called on the other side than the corresponding
\edtextlater, we need to store the side for a proper formatting of the footnote. We
also need to store the pstartL / pstartR counter.

2128     \ifledRcol%
2129     \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@true}}%
2130     {%

```

```

2131     \noexpand\setcounter{pstartR}{\noexpand\xpstartref{start:edtext:
later:\the\edtext@later}}%
2132     }%
2133     \else%
2134     \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@false}}%
2135     \unexpanded{\appto\@beforeinsertofthisedtext}%
2136     {%
2137     \noexpand\setcounter{pstartL}{\noexpand\xpstartref{start:edtext:
later:\the\edtext@later}}%
2138     }%
2139     \fi%
2140 %

```

And the footnote command of this `\edtextlater`.

```

2141     \unexpanded{#2}%
2142     }%
2143     }%
2144 %

```

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

```

2145     \edlabel{start:edtext:later:\the\edtext@later}%
2146     \flag@start@later%
2147     \bgroup%
2148     \def\lemma##1{%
2149     \ifledRcol%
2150     \write\linenum@outR{\string\@lemma}%
2151     \else%
2152     \write\linenum@out{\string\@lemma}%
2153     \fi%
2154     }%
2155     \renewcommand{\do}[1]{\expandafter\renewcommandx\csname ##1footnote\
endcsname[2][1,usedefault]{}\unskip}%\unskip because of a spurious space in
\newcommandx
2156     \dolistloop{\@series}%
2157     #2%
2158     \egroup%
2159     \showlemma{#1}%
2160     \edlabel{end:edtext:later:\the\edtext@later}%
2161     \flag@end@later%
2162 %

```

We decrease the counter increased at the beginning.

```

2163     \advance\@edtext@level by -\@ne%
2164     }%
2165 %

```

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

```

2166 \newcommand{\edtextnow}[0]{%
2167   \global\advance\edtext@now by \@ne
2168   \csuse{edtext@later@\the\edtext@now}%
2169 }%
2170 %

```

## 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.

```

2171 \list@create{\labelref@listR}
2172
2173 %

```

`\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`.

```

2174 \def\l@dmake@labelsR#1|#2|#3|#4|#5|#6|#7{%
2175   \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
2176     \relax%
2177   \else%
2178     \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
2179   \fi%
2180   \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\@Rlineflag}%
2181   \global\providetoggle{label@#7@ledRcol}%False is the default value of
   this toggle, which tells us whether a label is linked to the right or left
   side
2182   \global\toggletrue{label@#7@ledRcol}%
2183   \ignorespaces}
2184 \AtBeginDocument{%
2185   \def\l@dmake@labelsR#1|#2|#3|#4|#5|#6|#7{%
2186   }
2187
2188 %

```

`\@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 Sidenotes

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.

```

\marginR Specifies which margin sidenotes can be in.
\margin*
\marginR
2189 \WithSuffix\newcommand\margin*[1]{%
2190   \l@getssidenote@margin{#1}
2191   \global\marginR=\l@tempcntb
2192   \global\margin=\l@tempcntb
2193 }
2194 \newcommand\marginR[1]{%
2195   \l@getssidenote@margin{#1}%
2196   \global\marginR=\l@tempcntb%
2197 }%
2198 \newcount\marginR
2199 \global\marginR=\one%
2200
2201 \ifmovecolumnsonrightpage%
2202   \margin{inner}%
2203   \marginR{outer}%
2204 \fi%
2205
2206 %

```

```

\morespace@rightnote@leftcolumn
\morespace@leftnote@rightcolumn
\get@sidenote@morespace@columns

```

If we are typesetting parallel columns, we may want a left sidenote called on the right column be put on the left of the page and not on the left of the column, and a right sidenote called on the left column be put on the right of the page and not on the right of the column. We store as an option in the `\if@sidenotesmarginpage` boolean.

To do it, we need to add spaces, respectively defined on `\morespace@leftnote@rightcolumn` and `\morespace@rightnote@leftcolumn`. The `\get@sidenote@morespace@columns` calculate there two lengths.

```

2207 \newdimen\morespace@leftnote@rightcolumn%
2208 \newdimen\morespace@rightnote@leftcolumn%
2209
2210
2211
2212 \newcommand\get@sidenote@morespace@columns}{%
2213 %

```

The calculations are made only if `\if@sidenotesmarginpage` is TRUE.

```

2214   \if@sidenotesmarginpage%
2215 %

```

Now, we can calculate `\morespace@leftnote@rightcolumn` and `\morespace@rightnote@leftcolumn`

```

2216   \global\morespace@leftnote@rightcolumn=\dimexpr\intercolumns@width + \
Lcolwidth\relax%
2217   \global\morespace@rightnote@leftcolumn=\dimexpr\intercolumns@width + \
Rcolwidth\relax%
2218 %

```

If `\if@sidenotesmarginpage` is False

```

2219 \else%
2220 \global\@morespace@leftnote@rightcolumn=\z%
2221 \global\@morespace@rightnote@leftcolumn=\z%
2222 %

```

End of \get@sidenote@morespace@columns.

```

2223 \fi%
2224 }%
2225 %

```

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

```

2226 \newcommand*{\affixside@noteR}{%
2227 \prepare@edindex@for@note{\the\page@numR|\the\line@numR|\the\
subline@numR|\the\page@numR|\the\line@numR|\the\subline@numR|}%
2228 \def\sidenotecontent@{%
2229 \numgdef{\itemcount@}{0}%
2230 \def\do##1{%
2231 \ifnumequal{\itemcount@}{0}%
2232 {%
2233 \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
2234 {\appto\sidenotecontent@{\@sidenotesep ##1}%
2235 }%
2236 \numgdef{\itemcount@}{\itemcount@+1}%
2237 }%
2238 \dolistloop{\l@dcsnotetext}%
2239 \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
2240 \gdef\@templ@d{%
2241 \gdef\@templ@n{\l@dcsnotetext\l@dcsnotetext@1\l@dcsnotetext@r}%
2242 \ifx\@templ@d\@templ@n \else%
2243 \if@twocolumn%
2244 \if@firstcolumn%
2245 \setl@dlp@rbox{##1}{\sidenotecontent@}%
2246 \else%
2247 \setl@drp@rbox{\sidenotecontent@}%
2248 \fi%
2249 \else%
2250 \@l@tempcntb=\sidenote@marginR%
2251 \ifnum\@l@tempcntb>\@ne%
2252 \advance\@l@tempcntb by\page@numR%
2253 \fi%
2254 \ifodd\@l@tempcntb%
2255 \setl@drp@rbox{\sidenotecontent@}%
2256 \gdef\sidenotecontent@{%
2257 \numdef{\itemcount@}{0}%
2258 \dolistloop{\l@dcsnotetext@1}%
2259 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
2260 \setl@dlp@rbox{\sidenotecontent@}%

```

```

2261     \else%
2262     \setl@dlp@rbox{\sidenotecontent@}%
2263     \gdef\sidenotecontent@{ }%
2264     \numdef\itemcount@{0}%
2265     \dolistloop{\l@dcsnotetext@r}%
2266     \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{ }%
2267     \setl@drp@rbox{\sidenotecontent@}%
2268     \fi%
2269     \fi%
2270     \fi%
2271     \advance\@edindex@fornote@\m@ne%
2272   }
2273
2274 %

```

## XII Verse

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

```

\inserthangingsymbolL75 \newif\ifinserthangingsymbolR
\inserthangingsymbolR76 \newcommand{\inserthangingsymbolL}{%
2277   \ifinserthangingsymbol%
2278     \ifinstanzaL%
2279       \@hangingsymbol%
2280     \fi%
2281     \ifinastanzaL%
2282       \@hangingsymbol%
2283     \fi%
2284   \fi%
2285 }%
2286 \newcommand{\inserthangingsymbolR}{%
2287   \ifinserthangingsymbolR%
2288     \ifinstanzaR%
2289       \@hangingsymbol%
2290     \fi%
2291     \ifinastanzaR%
2292       \@hangingsymbol%
2293     \fi%
2294   \fi%
2295 }%
2296 %

```

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.

```

2297 \chardef\next=\catcode\&
2298 \catcode\&=\active
2299
2300 %

```

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

```

2301 \newenvironmentx{astanza}[2][1,2,usedefault]{%
2302 \ifledRcol%
2303 \global\inastanzaRtrue%
2304 \else%
2305 \global\inastanzaLtrue%
2306 \fi%
2307 \catcode\&=\active
2308 \global\stanza@count\@ne\stanza@modulo\@ne
2309 \@advancestanza@number%
2310 \newdimen\parindent@beforestanza%
2311 \parindent@beforestanza=\parindent%Keep in memory the standard \parindent
2312 \ifnum\usernamecount{sza@0@}=\z@
2313 \let\stanza@hang\relax
2314 \let\endlock\relax
2315 \else
2316 \rightskip\z@ plus 1fil\relax
2317 \fi
2318 \ifnum\usernamecount{szp@0@}=\z@
2319 \let\sza@penalty\relax
2320 \fi
2321 \def&{%
2322 \endlock\mbox{}%
2323 \sza@penalty
2324 \global\advance\stanza@count\@ne
2325 \@astanza@line}%
2326 \def\&{\@stopastanza}%
2327 \ifboolexpr{%
2328 not test{\ifdefvoid{\at@every@stanza}}%
2329 and test{\ifstrempy{#1}}%
2330 and test{\ifstrempy{#2}}}%
2331 {\pstart [] [\at@every@stanza] \at@start@every@stanza}%
2332 {\pstart [#1] [#2] \at@start@every@stanza}%
2333 \@astanza@line
2334 \@insertstanza@number%
2335 \let\par\relax\ignorespaces%No paragraph in verses
2336 }{%
2337 \global\undef\parindent@beforestanza%
2338 }%
2339
2340 %

```

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

```

2341 \newcommand{\@stopastanza}[2][1,2,usedefault]{%
2342   \endlock\mbox{}%
2343   \ifboolexpr{%
2344     not test{\ifdefvoid{\@every@stop@stanza}}%
2345     and test{\ifstrempy{#1}}%
2346     and test{\ifstrempy{#2}}%
2347   }%
2348   {\before@every@stop@stanza\pend[][\@every@stop@stanza]}%
2349   {\before@every@stop@stanza\pend[#1][#2]}%
2350 }%
2351 %

```

`\@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.

```

2352 \newcommand*\@astanza@line{%
2353   \stanza@indent%
2354   \endgraf
2355   \stanza@hang%
2356   \ignorespaces}
2357
2358 %

```

Lastly reset the modified category codes.

```

2359 \catcode`\&=\next
2360
2361 %

```

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

```

\thestanzaR
2362 \newcounter{stanzaL}
2363 \newcounter{stanzaR}
2364 \renewcommand{\thestanzaL}{%
2365   \textbf{\arabic{stanzaL}}%
2366 }
2367 \renewcommand{\thestanzaR}{%
2368   \textbf{\arabic{stanzaR}}%
2369 }
2370 %
2371 %

```

## XIII Fixing babel and polyglossia

With parallel texts there is the possibility that the two sides might use different languages via `babel`. On the other hand, nor `babel` nor `polyglossia` might not be called at all (even though it might be already built into the format).



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

```
\ifl@dusedbabel A flag for checking if babel has been used as a package.
\l@dusedbabelfalse
\l@dusedbabeltrue2372 \newif\ifl@dusedbabel
2373 %
```

```
\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 \.
```

```
2374 \patchcmd{\bbl@set@language}%
2375   {\select@language{\language}}%
2376   {\edef\language{#1}\select@language{\language}}%
2377   {}%
2378   {}%
2379
2380 %
```

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
2381 \newcommand*\l@duselanguage}[1]{}
2382 \gdef\theledlanguageL{}
2383 \gdef\theledlanguageR{}
2384
2385 %
```

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

```
2386 \AtBeginDocument{%
2387   \ifundefined{xpg@main@language}{%
2388     \ifundefined{bbl@main@language}{%
2389 %
```

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

```

2390 \l@dusedbabelfalse
2391 }{%
2392 %

```

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.

```

2393 \l@dusedbabeltrue
2394 \let\l@doldselectlanguage\selectlanguage
2395 \let\l@doldbbl@set@language\bbl@set@language
2396 \renewcommand{\selectlanguage}[1]{%
2397   \l@doldselectlanguage{#1}%
2398   \ifledRcol \gdef\theledlanguageR{#1}%
2399   \else      \gdef\theledlanguageL{#1}%
2400   \fi}
2401 %

```

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

```

2402 \renewcommand*{\l@duselanguage}[1]{%
2403   \let\bbl@savelastskip\relax% Babel 3.61, which provokes spurious
vertical space
2404   \let\bbl@restorelastskip\relax%% Babel 3.61, which provokes spurious
vertical space
2405   \expandafter\l@doldselectlanguage\expandafter{#1}%
2406   }%
2407 %

```

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

```

2408 \gdef\theledlanguageL{\bbl@main@language}%
2409 \gdef\theledlanguageR{\bbl@main@language}%
2410 }%
2411 }
2412 %

```

If use polyglossia

```

2413 {%
2414   \@ifpackagelater{polyglossia}{2020/04/08}{\led@err@polyglossiaTooOld}
%
2415   \let\old@otherlanguage\otherlanguage%
2416   \renewcommand{\otherlanguage}[2] []{%
2417     \xpg@set@group@aux%
2418     \selectlanguage[#1]{#2}%
2419     \ifledRcol \gdef\theledlanguageR{#2}%
2420     \else      \gdef\theledlanguageL{#2}%
2421     \fi}%
2422   \renewcommand{\l@duselanguage}[1]{%
2423     \csuse{no\language@name @numbers}%
2424     \@select@language{#1}%

```

```

2425 }%
2426 \gdef\theledlanguageL{\xpg@main@language}%
2427 \gdef\theledlanguageR{\xpg@main@language}%
2428 %
    That is it.
2429 }}
2430 %

```

## XIV Counts and boxes for parallel texts

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

`\maxchunks` The maximum number of chunk pairs before printing has to be called for. The default is 5120 chunk pairs.

```

2431 \newcount\l@dc@maxchunks
2432 \newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}
2433 \maxchunks{5120}
2434
2435 %

```

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

```

\l@dnumpstartsR
2436 \newcount\l@dnumpstartsR
2437
2438 %

```

`\l@dpscl` A couple of scratch counts to count left and right `\pstart`, respectively.

```

\l@dpscr
2439 \newcount\l@dpscl
2440 \newcount\l@dpscr
2441
2442 %

```

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

```

2443 \newcommand*\l@dsetuprawboxes{%
2444 \l@dttempcntb=\l@dc@maxchunks
2445 \loop\ifnum\l@dttempcntb>\z@
2446 \newnamebox{\l@dLcolrawbox\the\l@dttempcntb}
2447 \newnamebox{\l@dRcolrawbox\the\l@dttempcntb}
2448 \advance\l@dttempcntb \m@ne
2449 \repeat}
2450
2451 %

```

`\l@dsetupmaxlinecounts` `\l@dzeromaxlinecounts` 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.

```

2452 \newcommand*{\l@dsetupmaxlinecounts}{%
2453   \@l@dttempcntb=\l@dc@maxchunks
2454   \loop\ifnum\@l@dttempcntb>\z@
2455     \newnamecount{l@dmaxlinesinpar\the\@l@dttempcntb}
2456     \advance\@l@dttempcntb \m@ne
2457   \repeat}
2458 \newcommand*{\l@dzeromaxlinecounts}{%
2459   \begingroup
2460   \@l@dttempcntb=\l@dc@maxchunks
2461   \loop\ifnum\@l@dttempcntb>\z@
2462     \global\usenamecount{l@dmaxlinesinpar\the\@l@dttempcntb}=\z@
2463     \advance\@l@dttempcntb \m@ne
2464   \repeat
2465   \endgroup}
2466
2467 %

```

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

```

2468 \AtBeginDocument{%
2469   \l@dsetuprawboxes
2470   \l@dsetupmaxlinecounts
2471   \l@dzeromaxlinecounts
2472   \l@dnumpsstartsL=\z@
2473   \l@dnumpsstartsR=\z@
2474   \l@dpscl=\z@
2475   \l@dpscr=\z@}
2476
2477 %

```

## XV Checking text to be processed

```

\if@pstarts \check@pstarts returns \@pstartstrue if there are any unprocessed chunks.
\@pstartstrue
\@pstartsfalse
\check@pstarts
2478 \newif\if@pstarts
2479 \newcommand*{\check@pstarts}{%
2480   \@pstartsfalse
2481   \ifnum\l@dnumpsstartsL>\l@dpscl
2482     \@pstartstrue
2483   \else
2484     \ifnum\l@dnumpsstartsR>\l@dpscr
2485       \@pstartstrue
2486     \fi

```

```

2487 \fi
2488 }
2489
2490 %

```

`\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`.

```

\araw@texttrue
\araw@textfalse
\checkraw@text
2491 \newif\ifaraw@text
2492 \newcommand*\checkraw@text}{%
2493   \araw@textfalse
2494   \ifvbox\namebox{1@dLcolrawbox\the\l@dpscL}
2495     \araw@texttrue
2496   \else
2497     \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}
2498       \araw@texttrue
2499   \fi
2500 \fi
2501 }
2502
2503 %

```

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

```

2504 \newcommand*\@writelinesinparL}{%
2505   \edef\next{%
2506     \write\linenum@out{\string\@pend[\the\@donereallinesL]}}%
2507   \next
2508   \global\@donereallinesL \z@}
2509 \newcommand*\@writelinesinparR}{%
2510   \edef\next{%
2511     \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}%
2512   \next
2513   \global\@donereallinesR \z@}
2514
2515 %

```

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

```

\@writepageofparR
2516 \newcommand*\@writepageofparL}[0]{%
2517   \ifnum\@donereallinesL=\z@%
2518     \edef\next{%
2519       \write\linenum@out{\string\@pstart[\the\l@dpscL]{\the\c@page}{\the\
2520 numpagelinesL}}%
2521     }%
2522     \next%
2523   \fi%
2524 }%

```

```

2524 \newcommand*{\@writepageofparR}[0]{%
2525   \ifnum\@donereallinesR=\z@%
2526     \edef\next{%
2527       \write\linenum@outR{\string\@pstartR{\the\l@dpscR}{\the\c@page}{\the\
numpagelinesR}}%
2528     }%
2529     \next%
2530   \fi%
2531 }%
2532 %

```

## XVI Parallel columns

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

```

\@eledsectionR
2533 \newsavebox{\@eledsectionL}%
2534 \newsavebox{\@eledsectionR}%
2535 %

```

`\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.

```

2536 \newcommand*{\Columns}{%
2537   \ifl@dpairing%
2538     \led@err@Columns@InsideEnv%
2539   \fi%
2540   \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{
l@dLcolrawbox1}}}{%
2541     \ifcsboxvoid{l@dRcolrawbox1}{%
2542       \ifcsboxvoid{l@dLcolrawbox1}%
2543       {\led@err@Columns@WithoutEnv}%
2544       {\led@err@Columns@WithoutRightside}%
2545     }%
2546     {\led@err@Columns@WithoutLeftside}%
2547   }{%
2548     \global\l@dprintingcolumnstrue%
2549     \eledsection@correcting@skip=-\baselineskip% Correction for sections'
titles
2550     \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else
2551       \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
2552     \fi
2553   %

```

Start a group and zero counters, etc.

```

2554 \begingroup
2555   \l@dzeropenalties
2556   \endgraf\global\num@lines=\prevgraf
2557   \global\num@linesR=\prevgraf
2558   \global\par@line=\z@

```

```

2559 \global\par@lineR=\z@
2560 \global\l@dpscL=\z@
2561 \global\l@dpscR=\z@
2562 \get@familiarfootnote@number%
2563 \get@intercolumns@width%
2564 \get@sidenote@morespace@columns%
2565 %

```

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

```

2566 \check@pstarts
2567 \loop@if@pstarts
2568 \global\pstartnumtrue
2569 \global\pstartnumRtrue
2570 %

```

Increase `\l@dpscL` and `\l@dpscR` which here count the numbers of left and right chunks. Increase `\pstarts@typeset@L`, which counts the number of `\pstart` typeset both in not parallel mode and in the left side of parallel mode. Also restore the value of the public `pstart` counters.

```

2571 \global\advance\l@dpscL \@ne
2572 \global\advance\l@dpscR \@ne
2573 \global\advance\pstarts@typeset@L\@ne%
2574 \restore@pstartL@pc%
2575 \restore@pstartR@pc%
2576 %

```

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

```

2577 \Columns@print@before@pstart%
2578 %

```

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

```

2579 \checkraw@text
2580 { \loop\ifaraw@text
2581 %

```

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

```

2582 \l@duselanguage{\theledlanguageL}%
2583 \do@lineL
2584 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@}@%
2585 {%
2586 \ifdefstring{\@eledsectmark}{L}%
2587 {\csuse{eled@sectmark@the\pstarts@typeset@L}%
2588 }-}%
2589 \global\csundef{eled@sectmark@the\pstarts@typeset@L}%

```

```

2590         \savebox{\@eledsectionL}{\parbox[t][t]{\Lcolwidth}{\vbox
{}\print@eledsectionL}}%\vbox{-> prevent alignment troubles with RTL
language
2591         }%
2592         {}%
2593         \l@duselanguage{\theledlanguageR}%
2594         \do@lineR
2595         \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}
2596         {%
2597         \ifdefstring{\@eledsectmark}{R}%
2598         {\csuse{eled@sectmark@the\l@dpscR R}%
2599         }}%
2600         \global\csundef{eled@sectmark@the\l@dpscR R}%
2601         \savebox{\@eledsectionR}{\parbox[t][t]{\Rcolwidth}{\vbox
{}\print@eledsectionR}}%\vbox{-> prevent alignment troubles with RTL
language
2602         }%
2603         \hb@xt@ \hsize{%
2604         \ifdefstring{\columns@position}{L}{\hfill }%
2605         \print@leftcolumn%
2606         \print@columnseparator%
2607         \print@rightcolumn%
2608         \ifdefstring{\columns@position}{R}{\hfill}%
2609         }%
2610         \checkraw@text
2611         \checkverseL
2612         \checkverseR
2613         \checkpb@columns
2614         \repeat}
2615 %

```

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`.

```

2616     \@writelinesinparL
2617     \@writelinesinparR
2618     \check@pstarts
2619     \ifbypstart@%
2620         \unless\ifinstanza%
2621         \write\linenum@out{\string\@set[1]}%
2622         \resetprevline@%
2623     \fi%
2624     \fi
2625     \ifbypstart@R
2626         \unless\ifinstanza%
2627         \write\linenum@outR{\string\@set[1]}%
2628         \resetprevline@%
2629     \fi%
2630     \fi
2631     \Columns@print@after@pend%

```



```
2632 \repeat
2633 %
```

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.

```
2634 \flush@notes
2635 \flush@notesR
2636 \endgroup
2637 %
```

```
2638 \global\l@dpscL=\z@
2639 \global\l@dpscR=\z@
2640 \global\l@dnumpestartsL=\z@
2641 \global\l@dnumpestartsR=\z@
2642 \global\l@dprintingcolumnsfalse%
2643 \ignorespaces
2644 \global\instanzaLfalse%
2645 \global\instanzaRfalse%
2646 }%
2647 }%
2648
2649 %
```

`\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).

```
2650 \def\print@columnseparator{%
2651 \ifdim\beforecolumnseparator<0pt%
2652 \hfill%
2653 \else%
2654 \hspace{\beforecolumnseparator}%
2655 \fi%
2656 \columnseparator%
2657 \ifdim\aftercolumnseparator<0pt%
2658 \hfill%
2659 \else%
2660 \hspace{\aftercolumnseparator}%
2661 \fi%
2662 }%
2663 %
```

`\get@intercolumns@width` `\intercolumns@width` The `\intercolumns@width` is calculated by `\get@intercolumns@width`. This length depends of many parameters:

- Columns width;
- columns position;
- columns separator;

- space between columns and columns separator, which can be fixed by user or automatically calculated by `reledpar`.

This length is never used directly, but it is used to calculate some other lengths.

```
2664 \newdimen\intercolumns@width%
2665 \newcommand{\get@intercolumns@width}{%
2666 %   \begin{macrocode}
2667   \global\intercolumns@width=\z@%
2668   \ifdefstring{\columns@position}{C}%
2669 %
```

First case, the columns are centered.

```
2670   {%
2671 %
```

First sub-case, the width between columns is automatically calculated.

```
2672   \ifboolexpr{%
2673     test {\ifdimless{\beforecolumnseparator}{\z@}}%
2674     and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2675   }%
2676   {%
2677     \global\intercolumns@width=\dimexpr%
2678     \columnrulewidth +%
2679     (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth)*1/2%
The total of the width before and after column separator
2680     \relax%
2681   }%
2682   {%
2683 %
```

Second sub-case, the width between column is fully determined by user setting.

```
2684   \ifboolexpr{%
2685     test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2686     and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2687   }%
2688   {%
2689     \global\intercolumns@width=\dimexpr%
2690     \columnrulewidth + \beforecolumnseparator + \aftercolumnseparator
2691 %
2692     \relax%
2693   }%
2694   {%
```

Third sub-case, the width before column separator is determined by user setting, but the width after column separator is automatically calculated.

```
2695   \ifboolexpr{%
2696     test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2697     and test {\ifdimless{\aftercolumnseparator}{\z@}}%
```

```

2698 }%
2699 {%
2700   \global\intercolumns@width=\dimexpr%
2701     \columnrulewidth + \beforecolumnseparator +%
2702     (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
beforecolumnseparator) * 1/3%
2703   \relax%
2704 }%
2705 {}%
2706 %

```

Fourth and last sub-case, the width before column separator is automatically calculated, but the width after column separator is determined by user setting.

```

2707 \ifboolexpr{%
2708   test {\ifdimless{\beforecolumnseparator}{\z@}}%
2709   and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2710 }%
2711 {%
2712   \global\intercolumns@width=\dimexpr%
2713     \columnrulewidth + \aftercolumnseparator +%
2714     (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
aftercolumnseparator) * 1/3%
2715   \relax%
2716 }%
2717 {}%
2718 %

```

Now, we have finished with the case the columns are centered aligned.

```

2719 }%
2720 %

```

Other case, the columns are left or right aligned.

```

2721 {%
2722 %

```

First sub-case, the width between columns is automatically calculated.

```

2723 \ifboolexpr{%
2724   test {\ifdimless{\beforecolumnseparator}{\z@}}%
2725   and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2726 }%
2727 {%
2728   \global\intercolumns@width=\dimexpr%
2729     \columnrulewidth +%
2730     (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth)*2/3%
The total of the width before and after column separator
2731   \relax%
2732 }%
2733 {}%
2734 %

```

Second sub-case, the width between column is fully determined by user setting.

```

2735 \ifboolexpr{%
2736   test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2737   and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2738 }%
2739 {%
2740   \global\intercolumns@width=\dimexpr%
2741   \columnrulewidth + \beforecolumnseparator + \aftercolumnseparator
2742   %
2743   \relax%
2744 }%
2745 %

```

Third sub-case, the width before column separator is determined by user setting, but the width after column separator is automatically calculated.

```

2746 \ifboolexpr{%
2747   test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2748   and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2749 }%
2750 {%
2751   \global\intercolumns@width=\dimexpr%
2752   \columnrulewidth + \beforecolumnseparator +%
2753   (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
beforecolumnseparator) * 1/2%
2754   \relax%
2755 }%
2756 }%
2757 %

```

Fourth and last sub-case, the width before column separator is automatically calculated, but the width after column separator is determined by user setting.

```

2758 \ifboolexpr{%
2759   test {\ifdimless{\beforecolumnseparator}{\z@}}%
2760   and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2761 }%
2762 {%
2763   \global\intercolumns@width=\dimexpr%
2764   \columnrulewidth + \aftercolumnseparator +%
2765   (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
aftercolumnseparator) * 1/2%
2766   \relax%
2767 }%
2768 }%
2769 %

```

We have finished with the case the columns are left or right aligned.

```

2770 }%
2771 %

```

End of \get@intercolumns@width.

```
2772 }%
2773 %
```

`\print@leftcolumn` and `\print@rightcolumn` print the line number of the left or right column respectively. Normally, it is defined by the content of the `Leftside` or `Rightside` environments. But if `\movecolumnspositiononrightpage` is set to `TRUE`, it also depends on the page number.

```
2774 \newcommand{\print@leftcolumn}{%
2775   \if\page@num>\page@numR%
2776     \l@dttempcntb=\page@num%
2777   \else%
2778     \l@dttempcntb=\page@numR%
2779   \fi%
2780   \ifboolexpr{%
2781     not bool {movecolumnspositiononrightpage}%
2782     or test {\ifnumodd{\l@dttempcntb}}%
2783   }%
2784   {%
2785     \unhbox\l@dleftbox%
2786     \ifhbox\@eledsectionL%
2787       \usebox{\@eledsectionL}%
2788     \fi%
2789   }%
2790   {%
2791     \unhbox\l@drightbox%
2792     \ifhbox\@eledsectionR%
2793       \usebox{\@eledsectionR}%
2794     \fi%
2795   }%
2796 }%
2797
2798
2799 \newcommand{\print@rightcolumn}{%
2800   \if\page@num>\page@numR%
2801     \l@dttempcntb=\page@num%
2802   \else%
2803     \l@dttempcntb=\page@numR%
2804   \fi%
2805   \ifboolexpr{%
2806     not bool {movecolumnspositiononrightpage}%
2807     or test {\ifnumodd{\l@dttempcntb}}%
2808   }%
2809   {%
2810     \unhbox\l@drightbox%
2811     \ifhbox\@eledsectionR%
2812       \usebox{\@eledsectionR}%
2813     \fi%
```

```

2814 }%
2815 {%
2816   \unhbox\l@dleftbox%
2817   \ifhbox\@eledsectionL%
2818   \usebox{\@eledsectionL}%
2819   \fi%
2820 }%
2821 }%
2822 %

```

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

```

2823
2824 \newcommand{\checkpb@columns}{%
2825   \newif\if@pb
2826   \newif\if@nopb
2827   \IfStrEq{\led@pb@setting}{before}{
2828     \numdef{\next@absline}{\the\absline@num+1}%
2829     \numdef{\next@abslineR}{\the\absline@numR+1}%
2830     \xifinlist{\next@absline}{\l@prev@pb}{\@pbtrue}{}%
2831     \xifinlist{\next@abslineR}{\l@prev@pbR}{\@pbtrue}{}%
2832     \xifinlist{\next@absline}{\l@prev@nopb}{\@nopbtrue}{}%
2833     \xifinlist{\next@abslineR}{\l@prev@nopbR}{\@nopbtrue}{%
2834   }{}
2835   \IfStrEq{\led@pb@setting}{after}{
2836     \xifinlist{\the\absline@num}{\l@prev@pb}{\@pbtrue}{}%
2837     \xifinlist{\the\absline@numR}{\l@prev@pbR}{\@pbtrue}{%
2838     \xifinlist{\the\absline@num}{\l@prev@nopb}{\@nopbtrue}{}%
2839     \xifinlist{\the\absline@numR}{\l@prev@nopbR}{\@nopbtrue}{%
2840   }{}
2841   \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
2842   \if@pb\pagebreak[4]\fi
2843 }
2844 %

```

`\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).

```

2845 \newcommand*\columnseparator{%
2846   \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}
2847   \newdimen\columnrulewidth
2848   \columnrulewidth=\z@
2849
2850 %

```

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

`\columns@position`

```

2851 \newcommand*{\columnspan}[1]{%
2852   \xdef\columns@position{#1}%
2853 }%
2854 \xdef\columns@position{R}%
2855 %

```

`\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`.

```

2856 \newlength{\beforecolumnseparator}%
2857 \setlength{\beforecolumnseparator}{-2pt}%
2858
2859 \newlength{\aftercolumnseparator}%
2860 \setlength{\aftercolumnseparator}{-2pt}%
2861
2862 %

```

`\setwidthliketwocolumns` The `\setwidthliketwocolumns` macro is called in `\beginnumbering` in a **non-parallel** typesetting context, to fix the width of the lines to be vertically aligned with parallel columns. It is 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 the notes block.

```

2863 \newcommand{\setwidthliketwocolumns}{%
2864   \get@intercolumns@width%As \columns is not necessary called before \
setwidthliketwocolumns
2865   \hsize=\dimexpr \Lcolwidth + \intercolumns@width + \Rcolwidth \relax%
2866 }%
2867
2868 \newcommand{\setpositionliketwocolumns@L}{%
2869   \renewcommand{\ledrfill}{\hfill}%
2870 }%
2871
2872 \newcommand{\setnotespositionliketwocolumns@L}{%
2873 }%
2874
2875 \newcommand{\setpositionliketwocolumns@C}{%
2876   \doinsidelinehook{\hfill}%
2877   \renewcommand{\ledrfill}{\hfill}%
2878 }%
2879
2880 \newcommand{\setnotespositionliketwocolumns@C}{%
2881   \newdimen\temp%
2882   \newdimen\tempa%
2883   \temp=\hsize%
2884   \tempa=\Lcolwidth%

```

```

2885 \advance\tempa\Rcolwidth%
2886 \advance\temp-\tempa%
2887 \divide\temp by 2%
2888 \leftskip=\temp%
2889 \rightskip=-\temp%
2890 }%
2891
2892 \newcommand{\setpositionliketwocolumns@R}{%
2893 \doinsidelinehook{\hfill}%
2894 }%
2895 %

```

`\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.

```

2896 \newcommand{\Columns@print@before@pstart}{%
2897 \ifboolexpr{%
2898 test{\ifcsstring{before@pstartL@the\l@dpscl}{\at@every@pstart}}%
2899 and test {\ifcsstring{before@pstartR@the\l@dpscr}{\at@every@pstart}}%
2900 and test {\ifdefempty{\at@every@pstart}}}%
2901 {}%
2902 {}%
2903 \ifboolexpr{not togl{before@pstartR@the\l@dpscr @par} and not togl{
before@pstartL@the\l@dpscl @par}}{%
2904 \csuse{before@pstartL@the\l@dpscl}%
2905 \csuse{before@pstartR@the\l@dpscr}%
2906 }{%
2907 \hb@xt@ \hsize{%
2908 \ifdefstring{\columns@position}{L}{\hfill }%
2909 \par\parbox[t] [] [t]{\Lcolwidth}{%
2910 \csuse{before@pstartL@the\l@dpscl}%
2911 }%
2912 \print@columnseparator%
2913 \parbox[t] [] [t]{\Rcolwidth}{%
2914 \set@sectcountR%
2915 \csuse{before@pstartR@the\l@dpscr}%
2916 }%
2917 \ifdefstring{\columns@position}{R}{\hfill}%
2918 }%
2919 }%
2920 }%
2921 \global\csundef{before@pstartL@the\l@dpscl}%
2922 \global\csundef{before@pstartR@the\l@dpscr}%
2923 }%
2924 \newcommand{\Columns@print@after@pend}{%
2925 \ifboolexpr{%
2926 test{\ifcsstring{after@pendL@the\l@dpscl}{\at@every@pend}}%
2927 and test {\ifcsstring{after@pendR@the\l@dpscr}{\at@every@pend}}%
2928 and test {\ifdefempty{\at@every@pend}}}%

```



```

2929     }%
2930     {%
2931     \ifboolexpr{not togl{after@pendR@the\l@dpscR @par} and not togl{
after@pendL@the\l@dpscL @par}}{%
2932         \csuse{after@pendL@the\l@dpscL}%
2933         \csuse{after@pendR@the\l@dpscR}%
2934     }{%
2935         \hb@xt@ \hsize{%
2936             \ifdefstring{\columns@position}{L}{-}{\hfill }%
2937             \parbox[t] [] [t]{\Lcolwidth}{%
2938                 \csuse{after@pendL@the\l@dpscL}%
2939             }%
2940             \print@columnseparator%
2941             \parbox[t] [] [t]{\Rcolwidth}{%
2942                 \set@sectcountR%
2943                 \csuse{after@pendR@the\l@dpscR}%
2944             }%
2945             \ifdefstring{\columns@position}{R}{-}{\hfill}%
2946         }%
2947     }%
2948 }%
2949 \global\csundef{after@pendL@the\l@dpscL}%
2950 \global\csundef{after@pendR@the\l@dpscR}%
2951 }%
2952 %

```

## XVII Parallel pages

This is considerably more complicated than parallel columns.

### XVII.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`

```

2953 \newcount\numpagelinesL
2954 \newcount\numpagelinesR
2955 \newcount\l@dminpagelines
2956
2957 %

```

### XVII.2 Main macro

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

```

2958 \newcommand*{\Pages}[1][1,usedefault]{%
2959   \ifl@dpairing%
2960     \led@err@Pages@InsideEnv%
2961   \fi%
2962   \ifbool{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{
l@dLcolrawbox1}}}{%
2963     \ifcsboxvoid{l@dRcolrawbox1}{%
2964       \ifcsboxvoid{l@dLcolrawbox1}%
2965       {\led@err@Pages@WithoutEnv}%
2966       {\led@err@Pages@WithoutRightside}%
2967     }%
2968     {\led@err@Pages@WithoutLeftside}%
2969   }{%
2970   \ifstrequal{#1}{mainmatter}{\Pages@mainmattertrue}{\Pages@mainmatterfalse
}%
2971   \eledsection@correcting@skip=-2\baselineskip% line correcting for section
titles.
2972   \parledgroup@notespacing@set@correction%
2973   \typeout{}%
2974   \typeout{***** PAGES *****}%
2975   \ifnum\l@dnumstartsL=\l@dnumstartsR\else%
2976     \led@err@BadLeftRightPstarts{\the\l@dnumstartsL}{\the\l@dnumstartsR}%
2977   \fi%
2978   %

```

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

```

2979   \cleartol@devenpage%
2980   \global\l@dprintingpagestrue%
2981   \begingroup%
2982   %

```

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

```

2983   \Lcolwidth=\Lcolwidth@pages%
2984   \Rcolwidth=\Rcolwidth@pages%
2985   %

2986   \l@dzeropenalties%
2987   \endgraf\global\num@lines=\prevgraf%
2988     \global\num@linesR=\prevgraf%
2989   \global\par@line=\z@%
2990   \global\par@lineR=\z@%
2991   \global\l@dpscL=\z@%
2992   \global\l@dpscR=\z@%
2993   \writtenlinesLfalse%
2994   \writtenlinesRfalse%
2995   \get@familiarfootnote@number%
2996   %

```

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

```
2997 \let\print@Xnotes\print@Xnotes@forpages%
2998 \let\print@notesX\print@notesX@forpages%
2999 %
```

Check if there are chunks to be processed.

```
3000 \check@pstarts%
3001 \loop\if@pstarts%
3002 %
```

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

```
3003 \global\advance\l@dpscL \@ne%
3004 \global\advance\l@dpscR \@ne%
3005 %
```

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

```
3006 \getlinesfromparlistL%
3007 \getlinesfromparlistR%
3008 \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}%
3009 {\usernamecount{l@dmaxlinesinpar\the\l@dpscL}}%
3010 \check@pstarts%
3011 \repeat%
3012 %
```

Zero the counts again, ready for the next bit.

```
3013 \global\l@dpscL=\z@%
3014 \global\l@dpscR=\z@%
3015 %
```

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

```
3016 \getlinesfrompagelistL%
3017 \getlinesfrompagelistR%
3018 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
3019 {\l@dminpagelines}%
3020 %
```

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.

```
3021 \check@pstarts%
3022 \if@pstarts%
3023 %
```

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

```

3024 \global\advance\l@dpscl \@ne%
3025 \global\advance\l@dpscr \@ne%
3026 \global\advance\pstarts@typeset@L\@ne%
3027 \restore@pstartL@pc%
3028 \restore@pstartR@pc%
3029 %

```

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

```

3030 \global\@donereallinesL=\z@%
3031 \global\@donetotallinesL=\z@%
3032 \global\@donereallinesR=\z@%
3033 \global\@donetotallinesR=\z@%
3034 %

```

Start a loop over the boxes (chunks).

```

3035 \checkraw@text%
3036 %
3037 % \begingroup
3038 { \loop\ifaraw@text%
3039 %

```

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

```

3040 \checkpageL%
3041 \Leftpagehook%
3042 \l@duselanguage{\theledlanguageL}%
3043 { \loop\ifl@dsamepage%
3044 %

```

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

```

3045 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}%
3046 \csuse{before@pstartL@the\l@dpscl}%
3047 \global\csundef{before@pstartL@the\l@dpscl}%
3048 \do@lineL%
3049 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@@}%
3050 {\print@eledsectionL}%
3051 {}%
3052 \advance\numpagelinesL \@ne%
3053 %

```

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. `!@dleftbox`

```

3054     \ifshiftedpstarts%
3055         \ifdim\ht\l@dleftbox>0pt%
3056             \parledgroup@correction@notespacing{L}%
3057             \hb@xt@ \hsz{\ledstrutL\unhbox\l@dleftbox}%
3058         \else%
3059             \xifinlist{\the\pstarts@typeset@L}{\eled@sections@
} %
3060             {\add@apparatusL}%
3061             {}%
3062         \unless\ifadvancedshiftedpstarts%
3063             \dimen0=\pagetotal%
3064             \advance\dimen0 by \baselineskip%
3065             \global\pagetotal=\dimen0%
3066         \else%
3067             \ifnomaxlines%
3068                 \numdef{\@tmp}{\the\l@dpscL+1}%
3069                 \ifcsdef{minpage@pstart@\@tmp}{%
3070                     \ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}%
3071                         {\dimen0=\pagetotal%
3072                         \advance\dimen0 by \baselineskip%
3073                         \global\pagetotal=\dimen0%
3074                         }%
3075                         {}%
3076                         }{}%
3077                     \fi%
3078                 \fi%
3079             \fi%
3080         \else%
3081             \parledgroup@correction@notespacing{L}%
3082             \hb@xt@ \hsz{\ledstrutL\unhbox\l@dleftbox}%
3083         \fi%
3084     %

```

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

```

3085     \get@nextboxL%
3086     \global\l@dskipversenumberfalse%
3087     \ifprint@last@after@pendL%
3088         \csuse{after@pendL@\the\l@dpscL}%
3089         \global\csundef{after@pendL@\the\l@dpscL}%
3090     \fi%
3091     \checkpageL%
3092     \checkverseL%
3093     \checkpbL%
3094     \repeat%
3095 %

```



```

3135         \ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}}%
3136         {\dimen0=\pagetotal%
3137          \advance\dimen0 by \baselineskip%
3138          \global\pagetotal=\dimen0%
3139          }%
3140         {}%
3141         }{}%
3142         \fi%
3143         \fi%
3144         \fi%
3145     \else%
3146         \parledgroup@correction@notespacing{R}%
3147         \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
3148     \fi%
3149     \get@nextboxR%
3150     \global\l@dskipversenumberRfalse%
3151     \ifprint@last@after@pendR%
3152     \csuse{after@pendR@\the\l@dpscr}%
3153     \global\csundef{after@pendR@\the\l@dpscr}%
3154     \fi%
3155     \checkpageR%
3156     \checkverseR%
3157     \checkpbR%
3158     \repeat%
3159     \ifl@dpagefull%
3160     \@writelinesonpageR{\the\numpagelinesR}%
3161     \else%
3162     \@writelinesonpageR{1000}%
3163     \fi%
3164     \numpagelinesR=\z@%
3165     \parledgroup@correction@notespacing@init%
3166 %

```

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

```

3167     \clearl@drighthpage}%
3168 %

```

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.

```

3169     \checkraw@text%
3170     \ifaraw@text%
3171     \getlinesfrompagelistL%
3172     \getlinesfrompagelistR%
3173     \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
3174     {\l@dminpagelines}%
3175     \fi%
3176     \repeat}%
3177 %

```

We have now output the text from all the chunks.

```
3178     \fi%
3179 %
```

Make sure that there are no inserts hanging around.

```
3180     \flush@notes%
3181     \flush@notesR%
3182     \endgroup%
3183 %
```

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

```
3184     \global\l@dpscL=\z0%
3185     \global\l@dpscR=\z0%
3186     \global\l@dnumstartsL=\z0%
3187     \global\l@dnumstartsR=\z0%
3188     \global\instanzaLfalse%
3189     \global\instanzaRfalse%
3190     \global\inastanzaLfalse%
3191     \global\inastanzaRfalse%
3192     \global\l@dprintingpagesfalse%
3193 %
```

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

```
3194     \ifnum\edtext@later=\edtext@now%
3195     \else%
3196         \led@error@edtext@later@now%
3197     \fi%
3198     \global\edtext@later=\z0%
3199     \global\edtext@now=\z0%
3200 %
```

Prevent final notes from overlapping the line number

```
3201     \finish@Pages@notes%
3202     \ignorespaces}}%
3203
3204
3205 %
```

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

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

```
3206 \newcommand{\finish@Pages@notes}{%
3207     \def\do##1{%
3208 %
```



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

```

3209 \ifnocritical@%
3210 \global\newnamebox{##1footins}%
3211 \fi
3212 \ifnofamiliar@%
3213 \global\newnamebox{footins##1}%
3214 \fi
3215 %

```

We must also restore the settings for the footnotes in case of using `onlyside` option.

```

3216 \restore@Xnotes@settings{##1}%
3217 \restore@notesX@settings{##1}%
3218 %

```

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

```

3219 \ifvoid\csuse{##1footins}%
3220 \ifvoid\csuse{footins##1}\else%
3221 \newpage\null%
3222 \listbreak%
3223 \fi%
3224 \else%
3225 \newpage\null%
3226 \listbreak%
3227 \fi%
3228 }%
3229 \dolistloop{\@series}%
3230 }%
3231 %

```

## XVII.4 Struts

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

```

\ledstrutR
3232 \newcommand*{\ledstrutL}{}
3233 \newcommand*{\ledstrutR}{}
3234
3235 %

```

## XVII.5 Page clearing

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

```

3236 \providecommand{\cleartoevenpage}[1][\@empty]{%
3237 \clearpage
3238 \ifodd\c@page\hbox{##1}\clearpage\fi}

```

```

3239 \newcommand*\cleartol@evenpage}{%
3240 \ifdim\pagetotal<\topskip% on an empty page
3241 \else
3242 \clearpage
3243 \Pages@mainmatter%
3244 \fi
3245 \ifodd\c@page%
3246 \ifprevpgnotnumbered%
3247 \addtocounter{par@page}{-1}%
3248 \fi%
3249 \ifdef{\prevpgstyle}{\thispagestyle{\prevpgstyle}}{%
3250 \hbox{}}\clearpage%
3251 \fi%
3252 }%
3253 %
3254 %

```

`\clearl@leftpage` and `\clearl@rightpage` 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`

```

3255 \newcommand*\clearl@leftpage}{%
3256 \ifdim\pagetotal=0pt\hbox{}}\fi%
3257 \newpage%
3258 \insert@notes@for@onlyside%
3259 \ifodd\c@page\else
3260 \led@err@LeftOnRightPage
3261 \hbox{}}%
3262 \cleardoublepage
3263 \fi}
3264
3265 \newcommand*\clearl@rightpage}{%
3266 \ifdim\pagetotal=0pt\hbox{}}\fi%
3267 \newpage%
3268 \insert@notes@for@onlyside%
3269 \ifodd\c@page
3270 \led@err@RightOnLeftPage
3271 \hbox{}}%
3272 \cleartoevenpage
3273 \fi}
3274
3275 %

```

## XVII.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 for `\getlinesfromparlistR`.

```

3276 \newcommand*\getlinesfromparlistL{%
3277   \ifx\linesinpar@listL\empty
3278     \gdef\@cs@linesinparL{0}%
3279   \else
3280     \gl@p\linesinpar@listL\to\@cs@linesinparL
3281   \fi}
3282 \newcommand*\getlinesfromparlistR{%
3283   \ifx\linesinpar@listR\empty
3284     \gdef\@cs@linesinparR{0}%
3285   \else
3286     \gl@p\linesinpar@listR\to\@cs@linesinparR
3287   \fi}
3288
3289 %

```

`\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`.

```

3290 \newcommand*\getlinesfrompagelistL{%
3291   \ifx\linesonpage@listL\empty
3292     \gdef\@cs@linesonpageL{1000}%
3293   \else
3294     \gl@p\linesonpage@listL\to\@cs@linesonpageL
3295   \fi}
3296 \newcommand*\getlinesfrompagelistR{%
3297   \ifx\linesonpage@listR\empty
3298     \gdef\@cs@linesonpageR{1000}%
3299   \else
3300     \gl@p\linesonpage@listR\to\@cs@linesonpageR
3301   \fi}
3302
3303 %

```

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

```

3304 \newcommand*\@writelinesonpageL}[1]{%
3305   \edef\next{\write\linenum@out{\string\@lopL{#1}}}%
3306   \next}
3307 \newcommand*\@writelinesonpageR}[1]{%
3308   \edef\next{\write\linenum@outR{\string\@lopR{#1}}}%
3309   \next}
3310
3311 %

```

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

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

```

3312 \newcommand*{\l@dcalc@maxoftwo}[3]{%
3313   \ifnum #2>#1\relax
3314     #3=#2\relax
3315   \else
3316     #3=#1\relax
3317   \fi}
3318 \newcommand*{\l@dcalc@minoftwo}[3]{%
3319   \ifnum #2<#1\relax
3320     #3=#2\relax
3321   \else
3322     #3=#1\relax
3323   \fi}
3324
3325 %

```

## XVII.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@dpagefull` is set FALSE and `\l@dsamepagetrue` is set TRUE. If the page is spatially full then `\ifl@dpagefull` 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@dpagefull` and `\ifl@dsamepage` are set FALSE.

```

\ifl@dsamepage \checkpageL
\l@dsamepagetrue \ifl@dsamepage
\l@dsamepagefalse \ifl@dsamepage
\ifl@dpagefull \checkpageL
\l@dpagefulltrue \ifl@dsamepage
\l@dpagefullfalse \ifl@dsamepage

\checkpageL \newif\ifl@dsamepage
\checkpageR \l@dsamepagetrue
\checkpageR \newif\ifl@dpagefull
\checkpageR

3326 \newcommand*{\checkpageL}{%
3327   \l@dsamepagetrue
3328   \newif\ifl@dpagefull
3329
3330   \l@dpagefulltrue
3331   \l@dsamepagetrue
3332   \check@goal
3333   \ifdim\pagetotal<\ledthegoal
3334     \ifnum\numpagelinesL<\l@dsminpagelines
3335       \else
3336         \ifnomaxlines%
3337         \else%
3338         \l@dsamepagefalse%
3339         \l@dpagefullfalse%
3340       \fi%
3341     \fi
3342   \else
3343     \l@dsamepagefalse
3344     \l@dpagefulltrue
3345

```

```

3346 \fi%
3347 \ifprint@last@after@pendL%
3348 \l@dpagfullfalse%
3349 \l@dsamepagefalse%
3350 \print@last@after@pendLfalse%
3351 \fi%
3352 }%
3353
3354 \newcommand*{\checkpageR}{%
3355 \l@dpagfulltrue
3356 \l@dsamepagetrue
3357 \check@goal
3358 \ifdim\pagetotal<\ledthegoal
3359 \ifnum\numpagelinesR<\l@dmminpagelines
3360 \else
3361 \ifnomaxlines%
3362 \else%
3363 \l@dsamepagefalse%
3364 \l@dpagfullfalse%
3365 \fi%
3366 \fi
3367 \else
3368 \l@dsamepagefalse
3369 \l@dpagfulltrue
3370 \fi%
3371 \ifprint@last@after@pendR%
3372 \l@dpagfullfalse%
3373 \l@dsamepagefalse%
3374 \print@last@after@pendRfalse%
3375 \fi%
3376 }%
3377
3378 %

```

`\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`.

```

3379 \newcommand{\checkpbL}{
3380 \IfStrEq{\led@pb@setting}{after}{
3381 \xifinlist{\the\absline@num}{\l@prev@pb}{\l@dpagfulltrue\
l@dsamepagefalse}{
3382 \xifinlist{\the\absline@num}{\l@prev@nopb}{\l@dpagfullfalse\
l@dsamepagetrue}{
3383 }{
3384 \IfStrEq{\led@pb@setting}{before}{
3385 \numdef{\next@absline}{\the\absline@num+1}
3386 \xifinlist{\next@absline}{\l@prev@pb}{\l@dpagfulltrue\
l@dsamepagefalse}{
3387 \xifinlist{\next@absline}{\l@prev@nopb}{\l@dpagfullfalse\
l@dsamepagetrue}{

```

```

3388 }{}
3389 }
3390
3391 \newcommand{\checkpbR}{
3392   \IfStrEq{\led@pb@setting}{after}{
3393     \xifinlist{\the\absline@numR}{\l@prev@pbR}{\l@dpagelltrue\
3394     l@dsamepagefalse}{
3395       \xifinlist{\the\absline@numR}{\l@prev@nopbR}{\l@dpagellfalse\
3396       l@dsamepagetrue}{
3397         }{}
3398     \IfStrEq{\led@pb@setting}{before}{
3399       \numdef{\next@abslineR}{\the\absline@numR+1}
3400       \xifinlist{\next@abslineR}{\l@prev@pbR}{\l@dpagelltrue\
3401       l@dsamepagefalse}{
3402         \xifinlist{\next@abslineR}{\l@prev@nopbR}{\l@dpagellfalse\
3403         l@dsamepagetrue}{
3404       }{}
3405     }
3406   }
3407 }
3408 %

```

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

```

3403 \newcommand{\checkverseL}{%
3404   \ifinstanzaL%
3405     \iflednopbinverse%
3406       \ifinserthangingsymbol%
3407         \numdef{\prev@abslineverse}{\the\absline@num-1}%
3408         \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{}%
3409         \IfStrEq{\led@pb@setting}{before}{%
3410           \ifnum\numpagelinesL<3%
3411             \ledpbnum{\prev@abslineverse}%
3412           }{}%
3413         }{}%
3414       \fi%
3415     }{}%
3416   }
3417 }
3418 \newcommand{\checkverseR}{%
3419   \ifinstanzaR%
3420     \iflednopbinverse%
3421       \ifinserthangingsymbolR%
3422         \numdef{\prev@abslineverse}{\the\absline@numR-1}%
3423         \IfStrEq{\led@pb@setting}{after}{\lednopbnumR{\prev@abslineverse}}{}%
3424         \IfStrEq{\led@pb@setting}{before}{%
3425           \ifnum\numpagelinesR<3%
3426             \ledpbnumR{\prev@abslineverse}%
3427           }{}%
3428         }{}%
3429       \fi%

```

```

3430 \fi%
3431 \fi%
3432 }
3433 %

```

`\setgoalfraction` `\ledthegoal` is the amount of space allowed to be 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
3434 \newdimen\ledthegoal
3435 \ifshiftedpstarts
3436 \newcommand*\@goalfraction{-0.95}
3437 \else
3438 \newcommand*\@goalfraction{-0.9}
3439 \fi
3440
3441 \newcommand*\check@goal{%
3442 \ledthegoal=\@goalfraction\pagegoal}
3443 \newcommand{\setgoalfraction}[1]{%
3444 \xdef\@goalfraction{#1}%
3445 }
3446 %

```

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

```

\ifwrittenlinesL
3447 \newif\ifwrittenlinesL
3448 \newif\ifwrittenlinesR
3449
3450 %

```

## XVII.8 Getting boxes content

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

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

```

3451 \newif\if@getnextbox%
3452 %

```

`\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.

```

3453 \newcommand*\get@nextboxL{%
3454 \ifvbox\namebox{1@dLcolrawbox\the\l@dpscl}% box is not empty
3455 %

```

The current box is not empty; do nothing.

```
3456 \else% box is empty
3457 %
```

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

```
3458 \@getnextboxtrue%Should be local, but be cautious
3459 %
```

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

```
3460 \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscL}>\@donetotallinesL
3461 \parledgroup@notes@endL%
3462 \unless\ifnosyncpstarts%
3463 \getnextboxfalse%
3464 %
```

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.

```
3465 \ifnomaxlines%
3466 \ifdim\pagetotal<\ledthegoal%
3467 \numdef{\@tmp}{\l@dpscL+1}%
3468 \ifcsdef{afterlines@pstart@\@tmp R}{%
3469 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp R}}
3470 %
3471 {}%
3472 {\ifcsdef{minpage@pstart@\@tmp}%
3473 {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3474 {\ifnum\numpagelinesL=\l@dminpagelines%
3475 \getnextboxtrue%
3476 \fi%
3477 }%
3478 {\@getnextboxtrue}}%
3479 }%
3480 }%
3481 {}%
3482 \fi%
3483 \fi%
3484 \fi%
3485 \else%
3486 \ifnomaxlines%
3487 \numdef{\@tmp}{\the\l@dpscL+1}%
3488 \ifcsdef{minpage@pstart@\@tmp}{%
3489 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3490 {\ifdimgreater{\pagetotal}{\ledthegoal}%
3491 {\@getnextboxtrue}%
3492 {\@getnextboxfalse}%
3493 }%
3494 }
```



```

3494         {\@getnextboxtrue}%
3495         }{}}%
3496         \fi%
3497     \fi%
3498 %

```

Sufficient lines have been output.

```

3499     \if@getnextbox%
3500         \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscL}=\@donetotallinesL
3501             \parledgroup@notes@endL
3502         \fi
3503     \ifwrittenlinesL\else
3504 %

```

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

```

3505         \@writelinesinparL
3506         \writtenlinesLtrue
3507     \fi
3508     \ifnum\l@dnumstartsL>\l@dpscL
3509 %

```

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

```

3510         \writtenlinesLfalse
3511         \ifbypstart@
3512             \global\line@num=0%
3513             \resetprevline@%
3514         \fi
3515 % Add the content of the optional argument of the previous \protect\cs{pend
3516 }.
3517 % \begin{macrocode}
3518     \csuse{after@pendL@the\l@dpscL}%
3519     \global\csundef{after@pendL@the\l@dpscL}%
3520 %

```

Check the number of lines

```

3520     \l@dcalc@maxoftwo{\the\usernamecount{1@dmaxlinesinpar\the\l@dpscL}}%
3521         {\the\@donetotallinesL}%
3522         {\usernamecount{1@dmaxlinesinpar\the\l@dpscL}}}%
3523     \global\@donetotallinesL \z@
3524 %

```

Go to the next pstart

```

3525     \global\advance\l@dpscL \@ne
3526     \global\advance\pstarts@typeset@L\@ne%
3527     \global\pstartnumtrue%
3528     \restore@pstartL@pc%
3529 %

```

Add notes of parallel ledgroup.

```

3530         \parledgroup@notes@endL
3531         \parledgroup@correction@notespacing@final{L}
3532     \else
3533 %

3534         \print@last@after@pendLtrue%
3535     \fi
3536 \fi
3537 \fi}
3538 %

3539 \newcommand*{\get@nextboxR}{%
3540 \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}% box is not empty
3541 \else% box is empty
3542 \getnextboxtrue%
3543 \ifnum\usenamecount{1@dmaxlinesinpar\the\l@dpscR}>\@donetotallinesR
3544 \parledgroup@notes@endR
3545 \unless\ifnosyncpstarts%
3546 \getnextboxfalse%
3547 \ifnomaxlines%
3548 \ifdim\pagetotal<\ledthegoal%
3549 \numdef{\@tmp}{\l@dpscR+1}%
3550 \ifcsdef{afterlines@pstart@\@tmp L}{%
3551 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp L}}
3552 %
3553     }%
3554     {\ifcsdef{minpage@pstart@\@tmp}%
3555     {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3556     {\ifnum\numpagelinesR=\l@dminpagelines%
3557     \getnextboxtrue%
3558     \fi%
3559     }%
3560     {\@getnextboxtrue}}%
3561     {\@getnextboxtrue}%
3562     }%
3563     }%
3564     \fi%
3565 \fi%
3566 \fi%
3567 \else%
3568 \ifnomaxlines%
3569 \numdef{\@tmp}{\the\l@dpscR+1}%
3570 \ifcsdef{minpage@pstart@\@tmp}{%
3571 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3572 {\ifdimgreater{\pagetotal}{\ledthegoal}%
3573 {\@getnextboxtrue}%
3574 {\@getnextboxfalse}%

```

```

3575     }%
3576     {\@getnextboxtrue}%
3577   }{}
3578   \fi%
3579 \fi%
3580 \if@getnextbox%
3581   \ifnum\usernamecount{l@dmxlinesinpar\the\l@dpscR}=\@donetotallinesR
3582     \parledgroup@notes@endR
3583   \fi
3584   \ifwrittenlinesR\else
3585     \@writelinesinparR
3586     \writtenlinesRtrue
3587   \fi
3588   \ifnum\l@dnumpstartsR>\l@dpscR
3589     \writtenlinesRfalse
3590     \ifbypstart@R
3591       \unless\ifinstanzaR%
3592         \global\line@numR=0%
3593         \resetprevline@%
3594       \fi%
3595     \fi
3596     \csuse{after@pendR@\the\l@dpscR}%
3597     \global\csundef{after@pendR@\the\l@dpscR}%
3598     \l@dcalc@maxoftwo{\the\usernamecount{l@dmxlinesinpar\the\l@dpscR}}%
3599       {\the\@donetotallinesR}%
3600       {\usernamecount{l@dmxlinesinpar\the\l@dpscR}}%
3601     \global\@donetotallinesR \z@
3602     \global\advance\l@dpscR \@ne
3603     \global\pstartnumRtrue%
3604     \restore@pstartR@pc%
3605     \parledgroup@notes@endR
3606     \parledgroup@correction@notespacing@final{R}
3607   \else
3608     \print@last@after@pendRtrue%
3609   \fi
3610 \fi
3611 \fi}
3612
3613 %

```

## XVIII The 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@mainmatter14 \newif\ifPages@mainmatter
\Pages@mainmatter15 \newcommand{\Pages@mainmatter}{%

```

```

3616 \ifPages@mainmatter%
3617 \pagenumbering{arabic}%
3618 \addtocounter{page}{1}%
3619 \addtocounter{par@page}{-1}%
3620 \patchcmd{\thepage}{page}{par@page}{}{}%
3621 \fi%
3622 }
3623 %

```

## XIX Sections' titles' commands

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

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

```

\chapter 3624 \pretocmd{\chapter}{%
3625 \ifl@printingpages%
3626 \vbox{}%
3627 \fi%
3628 }%
3629 {}%
3630 {}%
3631 %

```

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

```

3632 \newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}}
3633 \eledsectnotoc{R}
3634 %

```

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

```

3635 \newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}}
3636 \eledsectmark{L}
3637 %

```

`\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`.

```

3638 \newskip\eledsection@correcting@skip
3639 %

```

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

```

3640 \newwrite\eled@sectioningR@out
3641 %

```

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

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

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

```
3642 \def\l@prev@pbR{}
3643 \def\l@prev@nopbR{}
3644 %
```

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

```
3645 \newcommand{\ledpbR}{\write\linenum@outR{\string\led@pbR}}
3646 \newcommand{\ledpbnumR}[1]{\write\linenum@outR{\string\led@pbnumR{#1}}}
3647 \newcommand{\lednopbR}{\write\linenum@outR{\string\led@nopbR}}
3648 \newcommand{\lednopbnumR}[1]{\write\linenum@outR{\string\led@nopbnumR{#1}}}
3649 %
```

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

```
3650 \newcommand{\led@pbR}{\listxadd{\l@prev@pbR}{\the\absline@numR}}
3651 \newcommand{\led@pbnumR}[1]{\listxadd{\l@prev@pbR}{#1}}
3652 \newcommand{\led@nopbR}{\listxadd{\l@prev@nopbR}{\the\absline@numR}}
3653 \newcommand{\led@nopbnumR}[1]{\listxadd{\l@prev@nopbR}{#1}}
3654 %
```

## XXI Parallel ledgroup

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

```
3655 \newmarks\parledgroup@
3656 \newmarks\parledgroup@series
3657 \newmarks\parledgroup@type
3658 %
```

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

```

3659 \newcommand{\parledgroup@notes@startL}{%
3660   \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscL}>0%
3661     \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
3662       bhooknoteX@\splitfirstmarks\parledgroup@series}}{X}%
3663     \IfStrEq{\splitfirstmarks\parledgroup@type}{footnote}{\csuse{
3664       bhookXnote@\splitfirstmarks\parledgroup@series}}{X}%
3665     \fi%
3666   \global\ledgroupnotesL@true%
3667   \insert@noterule@ledgroup{L}%
3668 }
3669 \newcommand{\parledgroup@notes@startR}{%
3670   \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscR}>0%
3671     \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
3672       bhooknoteX@\splitfirstmarks\parledgroup@series}}{X}%
3673     \IfStrEq{\splitfirstmarks\parledgroup@type}{footnote}{\csuse{
3674       bhookXnote@\splitfirstmarks\parledgroup@series}}{X}%
3675     \fi%
3676   \global\ledgroupnotesR@true%
3677   \insert@noterule@ledgroup{R}%
3678 }
3679 %

```

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

```

3676 \newcommand{\parledgroup@notes@endL}{%
3677   \global\ledgroupnotesL@false%
3678 }
3679 \newcommand{\parledgroup@notes@endR}{%
3680   \global\ledgroupnotesR@false%
3681 }
3682 %

```

`\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`

```

3683 \newcommand{\insert@noterule@ledgroup}[1]{
3684   \IfStrEq{\splitbotmarks\parledgroup@}{begin}{%
3685     \IfStrEq{\splitbotmarks\parledgroup@type}{footnote}{
3686       \csuse{ifledgroupnotes#1@}
3687       \vskip\skip\csuse{mp\splitbotmarks\parledgroup@series footins}
3688       \csuse{\splitbotmarks\parledgroup@series footnoterule}
3689     }
3690   }
3691   {}
3692   \IfStrEq{\splitbotmarks\parledgroup@type}{footnoteX}{
3693     \csuse{ifledgroupnotes#1@}

```

```

3694         \vskip\skip\csuse{mpfootins\splitbotmarks\parledgroup@series}
3695         \csuse{footnoterule\splitbotmarks\parledgroup@series}
3696         \fi
3697     }{}
3698 }
3699 {}
3700 }
3701 %

```

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

```

3702 \newcommand{\setparledgroupnotespacing}[1]{\gdef\@parledgroupnotespacing
3703   {#1}}
3704 \newcommand{\@parledgroupnotespacing}{}
3705 %

```

`\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`.

```

3705 \dimdef{\parledgroup@notespacing@correction}{0pt}
3706 \newcommand{\parledgroup@notespacing@set@correction}{%
3707   {\@getfirstseries\csuse{Xnotefontsize@\@firstseries}%We suppose all the
3708   series has the same footnote size setup
3709   \@parledgroupnotespacing\dimgdef{\temp@spacing}{\baselineskip}}%
3710   \dimgdef{\parledgroup@notespacing@correction}{\baselineskip-\temp@spacing}
3711 }%
3712 }
3713 %

```

`\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.

```

3712 \newcommand{\parledgroup@correction@notespacing@init}{
3713   \dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}
3714   \dimdef{\parledgroup@notespacing@correction@modulo}{0pt}
3715 }
3716 \parledgroup@correction@notespacing@init
3717 %

```

`\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`.

```

3718 \newcommand{\parledgroup@correction@notespacing@final}[1]{
3719   \ifparledgroup

```

```

3720 \vspace{\parledgroup@notespacing@correction@accumulated}
3721 \parledgroup@correction@notespacing@init%
3722 \ifstrequal{#1}{L}{
3723   \numdef{\@checking}{\the\l@dpscL-1}
3724 }{
3725   \numdef{\@checking}{\the\l@dpscR-1}
3726 }
3727 \dimdef{\@beforenotes@current@diff}{\csuse{@parledgroup@beforenotes@
@checking L}-\csuse{@parledgroup@beforenotes@\@checking R}}%
3728 \ifstrequal{#1}{L}%
3729   {% Left
3730   \ifdimgreater{@beforenotes@current@diff}{0pt}{-}\vspace{-\
@beforenotes@current@diff}}%
3731   }%
3732   {% Right
3733   \ifdimgreater{@beforenotes@current@diff}{0pt}{\vspace{\
@beforenotes@current@diff}}{-}
3734   }%
3735   \fi
3736 }
3737 %

```

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

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

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

```

3738 \newcommand{\parledgroup@correction@notespacing}[1]{%
3739   \csuse{ifledgroupnotes#1@}%
3740   \vspace{-\parledgroup@notespacing@correction}%
3741   \dimdef{\parledgroup@notespacing@correction@accumulated}{\
parledgroup@notespacing@correction@accumulated+\
parledgroup@notespacing@correction}%
3742   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo+\
parledgroup@notespacing@correction}%
3743   \ifdimless{\parledgroup@notespacing@correction@modulo}{\baselineskip
}{-}\advance\numpagelinesL -\@ne%
3744   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo-\baselineskip}%
3745   }% mean greater than equal

```



```

3746   \fi%
3747 }
3748 %

```

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

```

3749 \dimdef\parledgroup@beforenotesL{0pt}
3750 \dimdef\parledgroup@beforenotesR{0pt}
3751 %

```

`\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.

```

3752 \newcommand{\parledgroup@beforenotes@save}[1]{
3753   \ifparledgroup
3754     \csdimgdef{@parledgroup@beforenotes@\the\csuse{1@dnumstarts#1}#1}{\
csuse{parledgroup@beforenotes#1}}
3755     \csdimgdef{parledgroup@beforenotes#1}{0pt}
3756   \fi
3757 }
3758 %

```

## XXII Compatibility with eledmac

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

```

3759 \ifeledmaccompat%
3760
3761
3762   \unless\ifnocritical@
3763   \let\onlyXside\Xonlyside
3764   \fi
3765 \fi
3766 %

```

## XXIII The End

</code>

## Appendix A Some things to do when changing version

### A.1 Migration to eledpar 1.4.3

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

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

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

See the following two examples:

With standard `\hangingsymbol`:

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

With the modification of the `hangingsymbol`:

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

### 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.

#### A.2.1 Deprecated options

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

#### 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>

### 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>

## 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.

## A.4 Migration to `reledpar` 2.3.0

The line number style (alphabetic, numeric, etc.) for notes on the right-hand 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.

## 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.

## A.6 Migration to `reledpar` 2.5.0

If you use either `\stanza` or `astanza` environment, please read A.12 p. 415.

## A.7 Migration to `reledpar` 2.6.0

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

## A.8 Migration to `reledpar` 2.6.1

If you use `perpage` package to control footnote numbering, please read the handbook on 6.3.3 p. 16.

### A.9 Migration to reledpar 2.14.0

The `\linenumberlist` command is reserved for the left side. For the equivalent feature on the right side, define `\linenumberlistR`.

### A.10 Migration to reledpar 2.14.1

The execution of the code sent to `\doinsidelineLhook` and `\doinsidelineRhook` has been moved to a place where `\markboth` can work.

Normally, it should not interfere with your prior use of `\doinsidelineLhook` and `\doinsidelineRhook`.

Please contact us if that causes problems.

### A.11 Migration to reledpar v. 2.17.2

When changing the `\Lcolwidth` and `\Rcolwidth` length within an `pages` environment, the contents were aligned to the right. Now, they are aligned to the left. Please contact us if you need to retain the older behaviour.

### A.12 Migration to reledpar v. 2.24.3

There was an inconsistency with hanging verses. If you defined indentation of hanging verses as 2, with `setstanzaindents{2,...}`, the indent was, in reality, typeset as equal to 4. This happened:

- only in parallel typesetting;
- only with the `\stanza` macro, not with the `astanza` environment.

This inconsistency has been corrected. If you want to keep old indentation after upgrading the package, just multiply by two the first argument of `\setstanzaindents`.

## 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>\@morespace@leftnote@rightcolumn</code> .....	1
<code>\@morespace@rightnote@leftcolumn</code> .....	1
<code>\@nl</code> .....	1
<code>\@nl@regR</code> .....	1
<code>\@par@sync@option</code> .....	1
<code>\@par@this@sync@option</code> .....	1
<code>\@parledgroupnotespacing</code> .....	1
<code>\@pend</code> .....	1
<code>\@pendR</code> .....	1
<code>\@pstart</code> .....	1
<code>\@pstartR</code> .....	1
<code>\@pstartsfalse</code> .....	1
<code>\@pstartstrue</code> .....	1
<code>\@ref</code> .....	1
<code>\@ref@reg@parseR</code> .....	1
<code>\@ref@regR</code> .....	1
<code>\@set</code> .....	1
<code>\@stopastanza</code> .....	1
<code>\@writelinesinparL</code> .....	1
<code>\@writelinesinparR</code> .....	1
<code>\@writelinesonpageL</code> .....	1
<code>\@writelinesonpageR</code> .....	1
<code>\@writepageofparL</code> .....	1
<code>\@writepageofparR</code> .....	1
COMMAND\+ .....	57
COMMAND\@Rlineflag .....	91, 166
COMMAND\@adv .....	47, 165
COMMAND\@cs@linesinparL .....	123
COMMAND\@cs@linesonpageL .....	123
COMMAND\@eledsectionL .....	102
COMMAND\@eledsectionR .....	102
COMMAND\@eledsectmark .....	132

COMMAND\@eledsectnotoc	132
COMMAND\@goalfraction	15, 127
COMMAND\@l@dtempcnta	84
COMMAND\@lab	46, 91, 165
COMMAND\@lopL	52, 116, 123
COMMAND\@lopR	52, 123
COMMAND\@morespace@leftnote@rightcolumn	92
COMMAND\@morespace@rightnote@leftcolumn	92
COMMAND\@namedef	33
COMMAND\@namuse	33
COMMAND\@nl	46, 53, 54, 91, 165
COMMAND\@nl@regR	46
COMMAND\@page	91
COMMAND\@par@sync@option	45
COMMAND\@parledgroupnotespacing	135
COMMAND\@pend	51
COMMAND\@pendR	51
COMMAND\@pstart	52
COMMAND\@pstartstrue	100
COMMAND\@ref	49–51, 54, 165
COMMAND\@ref@later	50
COMMAND\@ref@reg@parseargR	50
COMMAND\@ref@regR	49
COMMAND\@set	47, 165
COMMAND\@sw	50
COMMAND\AtBeginPairs	10, 60, 163, 167
COMMAND\AtEveryPend	12, 163–165
COMMAND\AtEveryPend*	12
COMMAND\AtEveryPstart	3, 12, 22, 69, 103, 163–165
COMMAND\AtEveryPstart*	12
COMMAND\AtEveryPstartCall	3, 22, 69, 164
COMMAND\AtEveryStanza	167
COMMAND\AtEveryStopStanza	167
COMMAND\Clear the right lines for \read@linelist	45
COMMAND\Columns	9, 10, 21, 22, 58, 102, 110, 160–162, 164, 166
COMMAND\Columns@print@after@pend	112
COMMAND\Columns@print@before@pstart	112
COMMAND\Lcolwidth	10, 13, 114, 140
COMMAND\Lcolwidthth	168
COMMAND\Leftpagehook	22, 170
COMMAND\Leftsidehook	160
COMMAND\Leftsidehookend	160
COMMAND\MakePerPage	16
COMMAND\Pages	5, 9, 12, 13, 15, 21, 22, 29, 57, 58, 81, 85, 86, 88, 113, 114, 120, 131, 135, 160, 162, 164–166, 168
COMMAND\Pages@mainmatter	131
COMMAND\Pairs	57
COMMAND\Rcolwidth	10, 13, 114, 140
COMMAND\Rcolwidthth	168
COMMAND\Rightpagehook	22, 170

COMMAND\Rightsidehook	160
COMMAND\Rightsidehookend	160
COMMAND\Rlineflag	138
COMMAND\Xendlineflag	18, 166
COMMAND\Xlineflag	18, 139, 166
COMMAND\Xmaxhnotes	16
COMMAND\Xnoteswidthliketwocolumns	11, 163
COMMAND\Xonlyside	17, 85, 139, 167, 170
COMMAND\Xparindent	170
COMMAND\Xpstart	169
COMMAND\Xtxtbeforenotes	167
COMMAND\&	23
COMMAND\absline@numR	44
COMMAND\add@apparatusL	72
COMMAND\add@apparatusR	75
COMMAND\add@penalties	84
COMMAND\add@penaltiesL	84
COMMAND\advanceline	47, 54, 165
COMMAND\affixline@num	78
COMMAND\affixline@numR	78, 160, 161
COMMAND\affixpstart@num	81
COMMAND\affixpstart@numR	81
COMMAND\affixside@note	93
COMMAND\aftercolumnseparator	10, 111, 162, 168, 169
COMMAND\araw@textfalse	101
COMMAND\araw@texttrue	101
COMMAND\at@begin@pairs	60
COMMAND\autopar	22, 169
COMMAND\ballast@count	84
COMMAND\baselineskip	110, 136
COMMAND\bbl@set@language	97, 98, 165
COMMAND\beforecolumnseparator	10, 111, 162, 169
COMMAND\begin	23
COMMAND\beginnumbering	11, 18, 19, 22, 34, 46, 111, 161, 162, 164, 168, 169
COMMAND\beginnumberingR	53
COMMAND\bf	161
COMMAND\bfseries	161
COMMAND\brokenpenalty	84
COMMAND\by@autopar	65
COMMAND\chapter	132, 160
COMMAND\check@goal	127
COMMAND\check@pstarts	100
COMMAND\checkpageL	116, 124
COMMAND\checkpb@columns	110
COMMAND\checkpbL	125
COMMAND\checkpbR	125
COMMAND\checkraw@text	101
COMMAND\checkverseL	126
COMMAND\checkverseR	126
COMMAND\clear(double)page	121

COMMAND\clearl@dleftpage	122
COMMAND\clearl@drightpage	122
COMMAND\clearpage	122, 164
COMMAND\cleartoevenpage	121
COMMAND\cleartol@devenpage	121, 131
COMMAND\columnrulewidth	10, 110
COMMAND\columns	168
COMMAND\columns@position	110
COMMAND\columnseparator	10
COMMAND\columnspan	10, 162
COMMAND\continuousnumberingwithcolumns	169
COMMAND\correct@Xfootins@box	164
COMMAND\correct@footinsX@box	164
COMMAND\critext	163
COMMAND\csname	59
COMMAND\displaywidowpenalty	84
COMMAND\do@actions	76
COMMAND\do@actions@fixedcode	160
COMMAND\do@actions@nextR	77
COMMAND\do@actionsR	77, 160
COMMAND\do@ballast	84
COMMAND\do@ballastR	76
COMMAND\do@insidelineLhook	162
COMMAND\do@insidelineRhook	162
COMMAND\do@line	69
COMMAND\do@line(L/R)	73
COMMAND\do@lineL	70, 84, 160, 161
COMMAND\do@lineLhook	160
COMMAND\do@lineR	73, 160–162
COMMAND\do@lineRhook	160
COMMAND\do@lockoff	165
COMMAND\do@lockoffR	48
COMMAND\do@lockon	165
COMMAND\do@lockonR	48
COMMAND\doinsidelineLhook	22, 140, 163, 167
COMMAND\doinsidelineRhook	22, 140, 163, 167
COMMAND\doinsidelinehook	22
COMMAND\doinsidethislinehook	170
COMMAND\dolineLhook	22, 163
COMMAND\dolineRhook	22, 163
COMMAND\dolinehook	22
COMMAND\edindex	163
COMMAND\edlabel	161, 163
COMMAND\edlineref	168
COMMAND\edtext	17, 49, 54, 89, 163, 167, 168
COMMAND\edtext@later	89, 90, 120
COMMAND\edtext@now	120
COMMAND\edtextlater	17, 18, 88–90, 167–169
COMMAND\edtextnow	17, 18, 88–90, 167, 168
COMMAND\eled@sectioningR@out	132



COMMAND\eledchapter	164
COMMAND\eledsection	163–165, 169
COMMAND\eledsection@correcting@skip	132
COMMAND\eledsectmark	25, 132
COMMAND\eledsectnotoc	25, 132
COMMAND\eledxxx	163
COMMAND\end	23
COMMAND\endgraf	67
COMMAND\endlock	54, 165
COMMAND\endnumbering	11, 18, 22, 35, 164, 168, 169
COMMAND\endnumberingR	53
COMMAND\endsub	54, 165
COMMAND\endumbering	18
COMMAND\expandafter	56
COMMAND\extensionchars	34
COMMAND\firstlinenum	19, 162, 165, 166
COMMAND\firstsublinenum	162, 165, 166
COMMAND\fix@page	47, 165
COMMAND\flag@end	54, 160, 163
COMMAND\flag@start	54, 163
COMMAND\flush@notesR	85
COMMAND\footnote	58
COMMAND\footnoteX	55, 56
COMMAND\footnoteX@reading	166
COMMAND\footnoteXmk	17
COMMAND\footnoteXnomk	17, 56
COMMAND\frontmatter	15, 25
COMMAND\get@intercolumns@width	105, 109
COMMAND\get@nextboxL	161
COMMAND\get@nextboxR	161
COMMAND\get@sidenote@morespace@columns	92, 93
COMMAND\getline@numL	75
COMMAND\getline@numR	75
COMMAND\getlinesfrompagelistL	123
COMMAND\getlinesfrompagelistR	123
COMMAND\getlinesfromparlistL	123
COMMAND\getlinesfromparlistR	123
COMMAND\gl@p	56
COMMAND\goalfraction	138
COMMAND\hangingsymbol	138, 160
COMMAND\hfill	105, 111
COMMAND\hidenumbering	19, 165
COMMAND\if@getnextbox	127
COMMAND\if@sidenotesmarginpage	92
COMMAND\ifPages@mainmatter	131
COMMAND\ifbypage@	165
COMMAND\ifbypstart@R	165
COMMAND\ifcsboxvoid	33
COMMAND\ifdim	105
COMMAND\ifinserthangingsymbol	94

COMMAND\ifinserthangingsymbolR	94
COMMAND\ifl@dpagefull	124
COMMAND\ifl@dpaging	29, 163
COMMAND\ifl@dpairing	29, 160
COMMAND\ifl@dsamelang	163
COMMAND\ifl@dsamepage	124
COMMAND\ifl@pagefull	116
COMMAND\ifledRcol	29
COMMAND\iflinenumberLevenifblank	43
COMMAND\iflinenumberRevenifblank	43
COMMAND\iflledRcol	161
COMMAND\ifmovecolumnspostiononrightpage	28
COMMAND\ifnumberedpar@	62
COMMAND\ifnumberingR	161
COMMAND\ifnumberline	58
COMMAND\ifnumberpstart	58
COMMAND\ifpst@rtedL	34, 35, 63, 160
COMMAND\ifpst@rtedR	34
COMMAND\ifsublines@	48
COMMAND\ifsublines@R	170
COMMAND\insert@countR	49
COMMAND\insert@noterule@ledgroup	134
COMMAND\insert@notes@for@onlyside	88
COMMAND\insertlines@list	49
COMMAND\insertlines@listR	49
COMMAND\inserts@list	63
COMMAND\inserts@listR	83
COMMAND\intercolumns@width	105
COMMAND\l@d@nums	54
COMMAND\l@d@set	47, 54, 165
COMMAND\l@dLcolrawbox	62
COMMAND\l@dLcolrawbox1	99
COMMAND\l@dLcolrawbox2	99
COMMAND\l@dRcolrawbox	62
COMMAND\l@dcalc@maxoftwo	124
COMMAND\l@dcalc@minoftwo	124
COMMAND\l@dchecklang	160, 162
COMMAND\l@dcsnote	162
COMMAND\l@dleftbox	69, 116, 164
COMMAND\l@dlinenumR	43, 160
COMMAND\l@dlsnote	162
COMMAND\l@dmake@labels	91
COMMAND\l@dmaxlinesinpar	115
COMMAND\l@dmaxlinesinpar1	100
COMMAND\l@dminpagelines	115, 160
COMMAND\l@dnumpstartsL	99, 160
COMMAND\l@dprintingcolumnstrue	163
COMMAND\l@dprintingpagestrue	163
COMMAND\l@dpscL	103, 115, 129
COMMAND\l@dpscR	103, 115

COMMAND\@drsnote	162
COMMAND\@dsetupmaxlinecounts	100
COMMAND\@duselanguage	97, 98, 160
COMMAND\@dzeromaxlinecounts	100
COMMAND\@prev@nopbR	133
COMMAND\@prev@pbR	133
COMMAND\labelstarttrue	161
COMMAND\labelref@list	91
COMMAND\labelref@listR	91
COMMAND\lang	97
COMMAND\last@page@numR	47
COMMAND\led	161
COMMAND\led@nopbR	133
COMMAND\led@nopbnumR	133
COMMAND\led@pbR	133
COMMAND\led@pnumR	133
COMMAND\ledinnernote	23
COMMAND\ledleftnote	23
COMMAND\lednopb	23, 110, 125
COMMAND\lednopbR	133
COMMAND\lednopbnumR	133
COMMAND\ledouternote	23
COMMAND\ledpb	110, 125
COMMAND\ledpbR	133
COMMAND\ledpbnumR	133
COMMAND\ledrightnote	23
COMMAND\ledsidenote	23
COMMAND\ledstrutL	160
COMMAND\ledstrutR	160, 166
COMMAND\ledthegoal	127
COMMAND\ledtrutL	160, 166
COMMAND\leftlinenumR	43, 160
COMMAND\leftlinenumannotationR	44
COMMAND\lemma	89, 90
COMMAND\let	56
COMMAND\line@list@R	50
COMMAND\line@list@stuff	46, 53
COMMAND\line@margin	39
COMMAND\line@marginR	39, 160
COMMAND\line@numR	44
COMMAND\lineation	20, 164, 169
COMMAND\lineation*	20, 39, 163
COMMAND\lineationR	20, 38, 164, 167
COMMAND\linenum	89
COMMAND\linenum@out	91
COMMAND\linenum@outR	52
COMMAND\linenumOnlyPagesForColumns	21, 40, 169
COMMAND\linenumOnlyPagesForColumnsR	21, 40, 169
COMMAND\linenumannotationothersidettrue	170
COMMAND\linenumberLevenifblanktrue	21, 167

COMMAND\linenumberRevenifblank	167
COMMAND\linenumberRevenifblanktrue	21
COMMAND\linenumberannotation	169
COMMAND\linenumberlist	20, 140, 167
COMMAND\linenumberlistR	20, 140, 167
COMMAND\linenumberstyle	20, 139, 168
COMMAND\linenumberstyle*	139
COMMAND\linenumberstyleR	20, 139
COMMAND\linenumincrement	19, 162, 165, 166
COMMAND\linenummargin	20, 21, 39, 160, 165–167
COMMAND\linenummargin*	20, 39, 166
COMMAND\linenummarginColumns	20, 40
COMMAND\linenummarginColumns*	20
COMMAND\linenummarginColumnsR	20
COMMAND\linenummarginR	20, 21, 39, 166
COMMAND\linenumrepR	42, 160
COMMAND\linesinpar@listL	51, 123
COMMAND\linesonpage@listL	52, 123
COMMAND\lock@off	48
COMMAND\lock@on	48
COMMAND\mainmatter	3, 15, 25, 166
COMMAND\makeatletter	73
COMMAND\markboth	140, 167
COMMAND\maxchunks	9, 23, 99, 100
COMMAND\maxdimen	86
COMMAND\maxhnotesX	16
COMMAND\memorydump	18, 37
COMMAND\movecolumnsonrightpage	109
COMMAND\msdata	168
COMMAND\n@num	164
COMMAND\new@lineL	53
COMMAND\new@lineR	54
COMMAND\newhookcommand@series	58
COMMAND\newif	164
COMMAND\newpage	121, 122, 164
COMMAND\newseries	58
COMMAND\newseries@	55
COMMAND\newseries@par	55, 57, 58
COMMAND\next@line@list@stuffR	53, 169
COMMAND\noledxxx	163
COMMAND\nomark@	56
COMMAND\nomaxlines	52
COMMAND\notesXwidthliketwocolumns	11, 163
COMMAND\num@lines	84
COMMAND\num@lines(R)	62
COMMAND\numberingR	36
COMMAND\numberlinefalse	8, 169
COMMAND\numberlinetrue	169
COMMAND\numberonlyfirstinline	161
COMMAND\numberpstartfalse	19, 169

COMMAND\numberpstarttrue	19, 161, 165, 169
COMMAND\numberstanztrue	170
COMMAND\one@line	62, 94
COMMAND\one@lineR	62
COMMAND\onlyXside	139
COMMAND\onlysideX	17, 85, 88, 164, 167, 170
COMMAND\page@action	48, 165
COMMAND\pagenumbering	165
COMMAND\pages	15
COMMAND\pagetotal	116, 164
COMMAND\par@line	84
COMMAND\par@line(R)	62
COMMAND\par@sync@option	27
COMMAND\parindentX	170
COMMAND\parledgroup@	133
COMMAND\parledgroup@beforenotes@save	137
COMMAND\parledgroup@beforenotesL	137
COMMAND\parledgroup@beforenotesR	137
COMMAND\parledgroup@correction@notespacing	136
COMMAND\parledgroup@correction@notespacing@final	135
COMMAND\parledgroup@correction@notespacing@init	135
COMMAND\parledgroup@notes@endL	134
COMMAND\parledgroup@notes@endR	134
COMMAND\parledgroup@notes@startL	134
COMMAND\parledgroup@notes@startR	134
COMMAND\parledgroup@notespacing@correction	135, 136
COMMAND\parledgroup@notespacing@correction@accumulated	136
COMMAND\parledgroup@notespacing@correction@modulo	136
COMMAND\parledgroup@notespacing@set@correction	135
COMMAND\parledgroup@series	133
COMMAND\parledgroup@type	133
COMMAND\parledgroupnotespacing	138
COMMAND\parledgrouptrue	24
COMMAND\patchcmd	165
COMMAND\pausenumbering	11, 37, 167
COMMAND\pend	5, 8, 9, 12, 13, 21–24, 54, 58, 62, 63, 67–69, 99, 112, 162, 163, 165, 166
COMMAND\pendL	162
COMMAND\pendR	162
COMMAND\pends	22
COMMAND\perpage	16
COMMAND\prev@nopbR	133
COMMAND\prev@pbR	133
COMMAND\prevpgstyle	29
COMMAND\print@Xnotes	85
COMMAND\print@Xnotes@forpages	85, 87, 164
COMMAND\print@columnseparator	105, 162
COMMAND\print@eledsectionL	72
COMMAND\print@leftcolumn	109
COMMAND\print@line	71
COMMAND\print@lineL	71

COMMAND\print@notesX@forpages	164
COMMAND\print@rightcolumn	109
COMMAND\printlinenumR	139
COMMAND\printlinesR	166
COMMAND\pstart	5, 8, 9, 12, 13, 19, 21–24, 38, 47, 54, 58, 62, 63, 67, 69, 99, 103, 112, 161, 162, 164–166, 168–170
COMMAND\pstartL	69, 162
COMMAND\pstartR	69, 161, 162
COMMAND\pstartinfootnote	164
COMMAND\pstarts@typeset@L	103
COMMAND\raw@text	99
COMMAND\read@linelist	45, 46, 142, 165
COMMAND\ref@reg	49
COMMAND\ref@regR	49, 165
COMMAND\relax	139
COMMAND\reledmac	165
COMMAND\renewcommand	22, 138
COMMAND\resumenumbering	11, 37, 162, 167
COMMAND\resumenumberingR	163
COMMAND\rightlinenumR	43, 160
COMMAND\rightlinenumannotationR	44
COMMAND\sameword	89
COMMAND\section	160
COMMAND\section@num	34
COMMAND\selectlanguage	22, 97, 98
COMMAND\set@line	54, 165
COMMAND\set@line@action	48, 165
COMMAND\setRlineflag	21, 138
COMMAND\setgoalfraction	15, 138
COMMAND\sethangingsymbol	23
COMMAND\setline	47, 54, 165
COMMAND\setlinenum	48, 54, 165
COMMAND\setmsdataposition	168
COMMAND\setnoteposition...	111
COMMAND\setparledgroupnotespacing	138, 165
COMMAND\setposition...	111
COMMAND\setstanzaindents	10, 23, 138, 140
COMMAND\setwidthliketwocolumns	111
COMMAND\sidenotemargin	24, 162
COMMAND\sidenotemargin*	24, 162
COMMAND\sidenotemarginR	24, 168
COMMAND\skipnumbering	19, 164
COMMAND\sloppy	10
COMMAND\stanza	9, 10, 19, 23, 95, 139, 140, 161
COMMAND\stanzanumtrue	23
COMMAND\startlock	54, 165
COMMAND\startsub	54, 165
COMMAND\sub@action	48, 165
COMMAND\sub@off	91
COMMAND\sub@on	91

COMMAND\subline@numR	44
COMMAND\sublinenumberstyle	20, 139
COMMAND\sublinenumberstyle*	139
COMMAND\sublinenumberstyleR	20, 139
COMMAND\sublinenumincrement	162, 165, 166
COMMAND\sublinenumrepR	42, 160
COMMAND\syntaxonly	166
COMMAND\sza@0@	23
COMMAND\textheight	16
COMMAND\textwidth	60
COMMAND\the@labelX	166
COMMAND\theledlanguageL	97, 98
COMMAND\theledlanguageR	97, 98
COMMAND\thepage	25
COMMAND\thepstartL	19, 161
COMMAND\thepstartR	19, 161
COMMAND\thestanzaL	23
COMMAND\thestanzaR	23
COMMAND\this@section@last@page@numR	47
COMMAND\vbox	64
COMMAND\vskip	134
COMMAND\vsplit	84
COMMAND\widthliketwocolumns	11
COMMAND\widthliketwocolumnsfalse	11
COMMAND\widthliketwocolumnstrue	11
COMMAND\xflagref	166
COMMAND\xright@appenditem	56
COMMAND\xspace	26
COMMAND\xxxfootstart	111
ENVIRONMENTLeftside	12, 61, 109
ENVIRONMENTRightside	12, 61, 109
ENVIRONMENTastanza	23, 95, 139, 140, 166, 168, 170
ENVIRONMENTcolumns	36, 166
ENVIRONMENTledgroup	8, 166
ENVIRONMENTleft	19
ENVIRONMENTpages	29, 36, 59, 140, 164, 166, 168
ENVIRONMENTpairs	59, 164, 166
PACKAGEEDMAC	140
PACKAGEEDSTANZA	140
PACKAGEEledmac	164
PACKAGEEledpar	164
PACKAGETABMAC	140
PACKAGEbabel	22, 96–98, 166, 170
PACKAGEcalc	170
PACKAGEeledmac	6, 99, 137, 138, 140, 162, 165
PACKAGEeledpar	6, 7, 16, 41, 138, 139, 162, 163
PACKAGEetoolbox	33, 105, 133
PACKAGEledmac	7
PACKAGEledpar	2, 7
PACKAGEMemoir	140

PACKAGEMusixtex	162
PACKAGEperpage	3, 16, 139, 166
PACKAGEpolyglossia	22, 96–98, 167, 170
PACKAGEreledmac	2, 4, 6–9, 11, 12, 15, 16, 20, 22–26, 28, 29, 34, 35, 39, 41, 44–49, 51–56, 58, 71, 81, 91, 94, 115, 121, 138, 139, 165–170
PACKAGEreledpar	2, 4, 6–15, 22–28, 38, 44, 45, 51–53, 55, 58, 91, 106, 138–140, 165
PACKAGEsetspace	4, 25
PACKAGESyntonly	166
PACKAGEXkeyval	26
PACKAGEXR	167

## A

<code>\absline@numR</code>	1
<code>\actionlines@listR</code>	1
<code>\actions@listR</code>	1
<code>\add@apparatusL</code>	1
<code>\add@apparatusR</code>	1
<code>\add@inserts@nextR</code>	1
<code>\add@insertsR</code>	1
<code>\add@penaltiesL</code>	1
<code>\add@penaltiesR</code>	1
<code>\advanceline</code>	1
<code>\affixline@numR</code>	1
<code>\affixpstart@numL</code>	1
<code>\affixpstart@numR</code>	1
<code>\affixside@noteR</code>	1
<code>\aftercolumnseparator</code>	1, 10
<code>\annot@listR</code>	1
<code>\araw@textfalse</code>	1
<code>\araw@texttrue</code>	1
astanza (environment)	23
<code>\AtBeginPairs</code>	1, 10
<code>\AtEveryPstartCall</code>	1
<code>\autopar</code>	22

## B

<code>\bbl@set@language</code>	1
<code>\beforecolumnseparator</code>	1, 10
<code>\beginnumbering</code>	18
<code>\beginnumberingR</code>	1

## C

<code>\c@firstlinenumR</code>	1
<code>\c@firstsublinenumR</code>	1
<code>\c@linenumincrementR</code>	1
<code>\c@sublinenumincrementR</code>	1
<code>\ch@ck@l@ckR</code>	1
<code>\ch@cksub@l@ckR</code>	1
<code>\chapter</code>	1
<code>\chapterinpages</code>	1



<code>\check@goal</code> .....	1
<code>\check@pstarts</code> .....	1
<code>\checkpageL</code> .....	1
<code>\checkpageR</code> .....	1
<code>\checkpb@columns</code> .....	1
<code>\checkpbL</code> .....	1
<code>\checkpbR</code> .....	1
<code>\checkraw@text</code> .....	1
<code>\checkverseL</code> .....	1
<code>\checkverseR</code> .....	1
<code>\clearl@dleftpage</code> .....	1
<code>\clearl@drightpage</code> .....	1
<code>\cleartoevenpage</code> .....	1
<code>\cleartol@devenpage</code> .....	1
<code>\columnrulewidth</code> .....	1, 10
<code>\Columns</code> .....	1, 9
<code>\columns@position</code> .....	1
<code>\Columns@print@after@pend</code> .....	1
<code>\Columns@print@before@pstart</code> .....	1
<code>\columnseparator</code> .....	1, 10
<code>\columnsposition</code> .....	1, 10
<code>\correct@footinsX@box</code> .....	1
<code>\correct@Xfootins@box</code> .....	1
<code>\countLline</code> .....	1
<code>\countRline</code> .....	1
<code>\critext</code> .....	1

## D

<code>\do@actions@fixedcodeR</code> .....	1
<code>\do@actions@nextR</code> .....	1
<code>\do@actionsR</code> .....	1
<code>\do@ballastR</code> .....	1
<code>\do@insidelineLhook</code> .....	1
<code>\do@insidelineRhook</code> .....	1
<code>\do@lineL</code> .....	1
<code>\do@lineLhook</code> .....	1
<code>\do@lineR</code> .....	1
<code>\do@lineRhook</code> .....	1
<code>\do@lockoff</code> .....	1
<code>\do@lockoffR</code> .....	1
<code>\do@lockon</code> .....	1
<code>\do@lockonR</code> .....	1
<code>\doinsidelineLhook</code> .....	1, 22
<code>\doinsidelineRhook</code> .....	1, 22
<code>\dolineLhook</code> .....	1, 22
<code>\dolineRhook</code> .....	1, 22
<code>\dump@pstartL@pc</code> .....	1
<code>\dump@pstartR@pc</code> .....	1

## E

<code>\edlabel</code>	1
<code>\edtext</code>	1
<code>\edtext@later</code>	1
<code>\edtext@now</code>	1
<code>\edtextlater</code>	1, 17
<code>\edtextnow</code>	1, 17
<code>\eled@sectioningR@out</code>	1
<code>\eledsection@correcting@skip</code>	1
<code>\eledsectmark</code>	1, 25
<code>\eledsectnotoc</code>	1, 25
<code>\endlock</code>	1
<code>\endnumbering</code>	1, 18
<code>\endnumberingR</code>	1
<code>\endsub</code>	1
environments:	
<code>astanza</code>	23
<code>Leftside</code>	18
<code>pages</code>	12
<code>pairs</code>	9
<code>Rightside</code>	18

## F

<code>\f@x@l@cksR</code>	1
<code>\finish@Pages@notes</code>	1
<code>\first@linenum@out@Rfalse</code>	1
<code>\first@linenum@out@Rtrue</code>	1
<code>\firstlinenum</code>	1, 19
<code>\firstlinenum*</code>	1, 20
<code>\firstlinenumR</code>	1, 20
<code>\firstsublinenum</code>	1, 19
<code>\firstsublinenum*</code>	1, 20
<code>\firstsublinenumR</code>	1, 20
<code>\fix@page</code>	1
<code>\flag@end</code>	1
<code>\flag@start</code>	1
<code>\flush@notesR</code>	1
<code>\footnote@reading</code>	1
<code>\footnote@typeset</code>	1
<code>\footnoteXmk</code>	17
<code>\footnoteXnomk</code>	17

## G

<code>\get@familiarfootnote@number</code>	1
<code>\get@intercolumns@width</code>	1
<code>\get@nextboxL</code>	1
<code>\get@nextboxR</code>	1
<code>\get@sidenote@morespace@columns</code>	1
<code>\getline@numR</code>	1
<code>\getlinesfrompagelistL</code>	1

<code>\getlinesfrompagelistR</code> .....	1
<code>\getlinesfromparlistL</code> .....	1
<code>\getlinesfromparlistR</code> .....	1
<code>\goalfraction</code> .....	1

## H

<code>\hidenumbering</code> .....	19
-----------------------------------	----

## I

<code>\if@getnextbox</code> .....	1
<code>\if@pstarts</code> .....	1
<code>\if@sidenotesmarginpage</code> .....	1
<code>\ifaraw@text</code> .....	1
<code>\ifcsboxvoid</code> .....	1
<code>\iffirst@linenum@out@R</code> .....	1
<code>\ifl@dpagfull</code> .....	1
<code>\ifl@dpaging</code> .....	1
<code>\ifl@dpairing</code> .....	1
<code>\ifl@dsamepage</code> .....	1
<code>\ifl@dusedbabel</code> .....	1
<code>\ifledRcol</code> .....	1
<code>\iflinenumberLevenifblank</code> .....	1
<code>\iflinenumberRevenifblank</code> .....	1
<code>\ifmovecolumnsonrightpage</code> .....	1
<code>\ifnomaxlines</code> .....	1
<code>\ifnosyncpstarts</code> .....	1
<code>\ifPages@mainmatter</code> .....	1
<code>\ifprint@last@after@pendL</code> .....	1
<code>\ifprint@last@after@pendR</code> .....	1
<code>\ifpst@rtedL</code> .....	1
<code>\ifpst@rtedR</code> .....	1
<code>\ifpstartnumR</code> .....	1
<code>\ifresumenumberingR@start</code> .....	1
<code>\ifshiftedpstarts</code> .....	1
<code>\ifwidthliketwocolumns</code> .....	1
<code>\ifwrittenlinesL</code> .....	1
<code>\init@series@par</code> .....	1
<code>\initnumbering@sectcountR</code> .....	1
<code>\insert@countR</code> .....	1
<code>\insert@noterule@ledgroup</code> .....	1
<code>\insert@notes@for@onlyside</code> .....	1
<code>\inserthangingsymbolL</code> .....	1
<code>\inserthangingsymbolR</code> .....	1
<code>\insertlines@listR</code> .....	1
<code>\inserts@listR</code> .....	1
<code>\intercolumns@width</code> .....	1

## L

<code>\l@d@set</code> .....	1
<code>\l@dc@maxchunks</code> .....	1

<code>\ldcalc@maxoftwo</code>	1
<code>\ldcalc@minoftwo</code>	1
<code>\ldcalcnun</code>	1
<code>\ldchecklang</code>	1
<code>\ldleftbox</code>	1
<code>\ldlinenumR</code>	1
<code>\ldmake@labelsR</code>	1
<code>\ldminpagelines</code>	1
<code>\ldnumpstartsL</code>	1
<code>\ldnumpstartsR</code>	1
<code>\ldpagefullfalse</code>	1
<code>\ldpagefulltrue</code>	1
<code>\ldpscL</code>	1
<code>\ldpscR</code>	1
<code>\ldrightrightbox</code>	1
<code>\ldsamepagefalse</code>	1
<code>\ldsamepagetrue</code>	1
<code>\ldsetupmaxlinecounts</code>	1
<code>\ldsetuprawboxes</code>	1
<code>\ldskipversenumberR</code>	1
<code>\ldusedbabelfalse</code>	1
<code>\ldusedbabeltrue</code>	1
<code>\lduselanguage</code>	1
<code>\ldzeromaxlinecounts</code>	1
<code>\labelref@listR</code>	1
<code>\last@page@numR</code>	1
<code>\Lcolwidth</code>	1, 10, 13
<code>\led@err@BadLeftRightPstarts</code>	1
<code>\led@err@Columns@InsideEnv</code>	1
<code>\led@err@Columns@WithoutEnv</code>	1
<code>\led@err@Columns@WithoutLeftside</code>	1
<code>\led@err@Columns@WithoutRightside</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@Pages@WithoutLeftside</code>	1
<code>\led@err@Pages@WithoutRightside</code>	1
<code>\led@err@polyglossiaTooOld</code>	1
<code>\led@err@RightOnLeftPage</code>	1
<code>\led@err@Rightside@PreviousNotPrinted</code>	1
<code>\led@err@TooManyPstarts</code>	1
<code>\led@error@edtext@later@now</code>	1
<code>\led@error@fail@patch@mempnum</code>	1
<code>\led@error@fail@patch@outputpage</code>	1
<code>\led@error@fail@patch@pagenumbering</code>	1
<code>\led@error@fail@patch@thepage</code>	1
<code>\led@error@missing@numbering</code>	1
<code>\led@error@note@called@onleftside</code>	1
<code>\led@error@note@called@onrightside</code>	1

<code>\led@nopbnumR</code> .....	1
<code>\led@nopbR</code> .....	1
<code>\led@pbnumR</code> .....	1
<code>\led@pbR</code> .....	1
<code>\led@warn@ChangeSyncOption</code> .....	1
<code>\led@warn@setting@in@rightside</code> .....	1
<code>\lednopbnum</code> .....	1
<code>\lednopbnumR</code> .....	1
<code>\ledpbnumR</code> .....	1
<code>\ledpbR</code> .....	1
<code>\ledstrutL</code> .....	1
<code>\ledstrutR</code> .....	1
<code>\ledthegoal</code> .....	1
<code>\leftlinenumannotationR</code> .....	1
<code>\leftlinenumR</code> .....	1
<code>\Leftpagehook</code> .....	1, 22
<code>\leftpstartnumL</code> .....	1
<code>\leftpstartnumR</code> .....	1
<code>Leftside (environment)</code> .....	18
<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, 20
<code>\lineationR</code> .....	1, 20
<code>\linenum@outR</code> .....	1
<code>\linenumberLevenifblanktrue</code> .....	21
<code>\linenumberRevenifblanktrue</code> .....	21
<code>\linenumberstyle*</code> .....	1, 20
<code>\linenumberstyleR</code> .....	1, 20
<code>\linenumincrement</code> .....	1, 19
<code>\linenumincrement*</code> .....	1, 20
<code>\linenumincrementR</code> .....	1, 20
<code>\linenummargin</code> .....	1
<code>\linenummargin*</code> .....	1, 20
<code>\linenummarginColumns</code> .....	1, 20
<code>\linenummarginColumns*</code> .....	1, 20
<code>\linenummarginColumnsR</code> .....	1, 20
<code>\linenummarginR</code> .....	1, 20
<code>\linenumOnlyPagesForColumns</code> .....	1, 20
<code>\linenumOnlyPagesForColumnsR</code> .....	1, 20
<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, 9
<code>\maxlinesinpar@list</code> .....	1
<code>\memorydump</code> .....	18
<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>\next@line@list@stuffR</code> .....	1
<code>\notesXwidthliketwocolumns</code> .....	11
<code>\num@linesR</code> .....	1
<code>\numberpstartfalse</code> .....	19
<code>\numberpstarttrue</code> .....	19
<code>\numpagelinesL</code> .....	1
<code>\numpagelinesR</code> .....	1

## O

<code>\one@lineR</code> .....	1
<code>\onlysideX</code> .....	17
<code>optionadvancedshiftedpstarts</code> .....	13, 14, 27
<code>optionauxdir</code> .....	167
<code>optioncontinuousnumberingwithcolumns</code> .....	11, 167–169
<code>optionmainmatter</code> .....	131
<code>optionmovecolumnsonrightpage</code> .....	12, 168
<code>optionnomaxlines</code> .....	13, 14, 27
<code>optionnopbinverse</code> .....	168
<code>optionnosyncpstarts</code> .....	14, 27, 127
<code>optionprevpgnotnumbered</code> .....	168
<code>optionprevpgstyle</code> .....	168
<code>optionsameparallelpagenumber</code> .....	165, 169
<code>optionshiftedpstarts</code> .....	7, 14, 27, 161
<code>optionshiftedverses</code> .....	160, 161
<code>optionsidenotesmarginpage</code> .....	24, 28, 169
<code>optionwidthliketwocolumns</code> .....	11, 169

## P

<code>\page@action</code> .....	1
<code>\page@numR</code> .....	1
<code>\Pages</code> .....	1, 12
<code>pages (environment)</code> .....	12
<code>\Pages@mainmatter</code> .....	1
<code>pairs (environment)</code> .....	9
<code>\par@lineR</code> .....	1

<code>\parledgroup@</code> .....	1
<code>\parledgroup@beforenotes@save</code> .....	1
<code>\parledgroup@beforenotesL</code> .....	1
<code>\parledgroup@beforenotesR</code> .....	1
<code>\parledgroup@correction@notespacing</code> .....	1
<code>\parledgroup@correction@notespacing@final</code> .....	1
<code>\parledgroup@correction@notespacing@init</code> .....	1
<code>\parledgroup@notes@startL</code> .....	1
<code>\parledgroup@notes@startR</code> .....	1
<code>\parledgroup@notespacing@correction</code> .....	1
<code>\parledgroup@notespacing@set@correction</code> .....	1
<code>\parledgroupseries@</code> .....	1
<code>\parledgroupstype@</code> .....	1
<code>\pausenumbering@page@numR</code> .....	1
<code>\pausenumberingR</code> .....	1
<code>\pend</code> .....	21
<code>\pendL</code> .....	1
<code>\pendR</code> .....	1
<code>\prev@nopbR</code> .....	1
<code>\prev@pbR</code> .....	1
<code>\prevpgstyle</code> .....	1
<code>\print@columnseparator</code> .....	1
<code>\print@eledsectionL</code> .....	1
<code>\print@eledsectionR</code> .....	1
<code>\print@leftcolumn</code> .....	1
<code>\print@lineL</code> .....	1
<code>\print@lineR</code> .....	1
<code>\print@notesX@forpages</code> .....	1
<code>\print@rightcolumn</code> .....	1
<code>\print@Xnotes@forpages</code> .....	1
<code>\pstart</code> .....	21
<code>\pstartL</code> .....	1
<code>\pstartR</code> .....	1

## R

<code>\Rcolwidth</code> .....	1, 10, 13
<code>\read@linelist</code> .....	1
<code>\reledpar@error</code> .....	1
<code>\reledpar@warning</code> .....	1
<code>\restore@notesX@settings</code> .....	1
<code>\restore@pstartL@pc</code> .....	1
<code>\restore@pstartR@pc</code> .....	1
<code>\restore@Xnotes@settings</code> .....	1
<code>\resumenumberingR</code> .....	1
<code>\rightlinenumannotationR</code> .....	1
<code>\rightlinenumR</code> .....	1
<code>\Rightpagehook</code> .....	1, 22
<code>\rightpstartnumL</code> .....	1
<code>\rightpstartnumR</code> .....	1
Rightside (environment) .....	18

<code>\Rightsidehook</code> .....	1
<code>\Rightsidehookend</code> .....	1
<code>\Rlineflag</code> .....	1

## S

<code>\save@familiarfootnote@number</code> .....	1
<code>\save@section@number</code> .....	1
<code>\section@numR</code> .....	1
<code>\selectlanguage</code> .....	1
<code>\set@line</code> .....	1
<code>\set@line@action</code> .....	1
<code>\set@sectcountR</code> .....	1
<code>\setgoalfraction</code> .....	1, 15
<code>\sethangingsymbol</code> .....	23
<code>\setline</code> .....	1
<code>\setlinenum</code> .....	1
<code>\setnamebox</code> .....	1
<code>\setnotepositionliketwocolumns@C</code> .....	1
<code>\setnotepositionliketwocolumns@L</code> .....	1
<code>\setnotepositionliketwocolumns@R</code> .....	1
<code>\setpositionliketwocolumns@C</code> .....	1
<code>\setpositionliketwocolumns@L</code> .....	1
<code>\setpositionliketwocolumns@R</code> .....	1
<code>\setRlineflag</code> .....	21
<code>\setwidthliketwocolumns</code> .....	1
<code>\sidenote@marginR</code> .....	1
<code>\sidenotemargin*</code> .....	1
<code>\sidenotemarginR</code> .....	1
<code>\skip@lockoff</code> .....	1
<code>\skipnumbering</code> .....	1, 19
<code>\startlock</code> .....	1
<code>\startsub</code> .....	1
<code>\sub@action</code> .....	1
<code>\subline@numR</code> .....	1
<code>\sublinenumberstyle*</code> .....	1, 20
<code>\sublinenumberstyleR</code> .....	1, 20
<code>\sublinenumincrement</code> .....	1, 19
<code>\sublinenumincrement*</code> .....	1, 20
<code>\sublinenumincrementR</code> .....	1, 20
<code>\sublinenumrepr</code> .....	1

## T

<code>\theledlanguageL</code> .....	1
<code>\theledlanguageR</code> .....	1
<code>\thepstartL</code> .....	19
<code>\thepstartR</code> .....	19
<code>\thestanzaL</code> .....	1, 23
<code>\thestanzaR</code> .....	1, 23
<code>\this@section@last@page@numR</code> .....	1
<code>\this@section@page@numR</code> .....	1



U

`\unhnamebox` ..... 1  
`\unvnamebox` ..... 1  
`\usernamecount` ..... 1

W

`\widthliketwocolumns` ..... 11

X

`\Xendlineflag` ..... 18  
`\Xlineflag` ..... 18  
`\Xnoteswidthliketwocolumns` ..... 11  
`\Xonlyside` ..... 17

## Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
\Columns: Added \l@dchecklang and \l@duselanguage to \Columns	103
\Pages: Added \l@duselanguage to \Pages	116
General: Added section of babel related code	96
Fix babel problems	1
v0.3.0.	
\Pages: Added \ledstrutL to \Pages	116
Added \ledstrutR to \Pages	118
\Rightsidehookend: Added \Leftsidehook, \Leftsidehookend, \Rightsidehook and \Rightsidehookend	61
\affixline@numR: Changed \affixline@numR to match new eledmac	78
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR	77
\do@lineL: Added \do@lineLhook to \do@lineL	70
Simplified \do@lineL by using macros for some common code	70
\do@lineR: Changed \do@lineR similarly to \do@lineL	73
\flag@end: Removed extraneous spaces from \flag@end	54
\ifledRcol: Moved \ifl@dpairing to eledmac	29
\ifpst@rtedR: Moved \ifpst@rtedL to eledmac	34
\l@dlinenumR: Simplified \leftlinenumR and \rightlinenumR by introducing \l@dlinenumR	43
\l@dnumpstartsR: Moved \l@dnumpstartsL to eledmac	99
\ledstrutR: Added \ledstrutL and \ledstrutR	121
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR	42
General: Added \do@lineLhook and \do@lineRhook	73
Added hooks into Leftside environment	61
Reorganize for ledarab	1
v0.3.a.	
\line@marginR: Do not just set \line@marginR in \linenummargin	39
General: Minor \linenummargin fix	1
v0.3.b.	
\Pages: Added \l@dminpagelines calculation for succeeding page pairs	119
General: Improved parallel page balancing	1
v0.3.c.	
General: Compatilby with Polyglossia	1
v0.4.0.	
General: No more ledparpatch. All patches are now in the main file.	1
v0.5.0.	
General: Corrections about \section and other titles in numbered sections	1
v0.6.0.	
General: Be able to us \chapter in parallel pages.	1
v0.7.0.	
General: Add shiftedverses option which makes 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 \hangingsymbol to define the character.	1

v0.9.0.	
\ifledRcol: Moved \iflledRcol and \ifnumberingR to eledmac . . . . .	29
General: Possibility to number \pstart. . . . .	19
Possibility to number the pstart with the commands \numberpstarttrue. . . . .	1
v0.9.1.	
General: The numbering of the pstarts restarts on each \beginnumbering. . . . .	1
v0.9.2.	
General: Debug: with \Columns, the hanging indentation now runs on the left columns and the hanging symbol is shown only when \stanza is used. . . . .	1
v0.9.3.	
General: \thepstartL and \thepstartR use now \bfseries and not \bf, which is deprecated and makes conflicts with memoir class. . . . .	1
v0.10.0.	
General: \edlabel commands on the right side are now correctly indicated. . . . .	1
\edlabel commands which start a paragraph are now put in the right place. . . . .	1
v0.11.0.	
\Columns: Line numbering by pstart. . . . .	104
\affixline@numR: Changed \affixline@numR to allow to disable line numbering (like in eledmac 0.15). . . . .	78
\get@nextboxR: Change \get@nextboxL and \get@nextboxR to allow to disable line numbering (like in eledmac 0.15). . . . .	127
Pstart number can be printed in side . . . . .	129
\inserthangingsymbolR: Prevent the column separator for hanging verse from shifting	94
General: Change \do@lineL and \do@lineR to allow line numbering by pstart (like in eledmac 0.15). . . . .	70
Lineation can be by pstart (like in eledmac 0.15). . . . .	38
New management of hangingsymbol insertion, preventing undesirable insertions. . .	94
v0.12.0.	
General: New management of hangingsymbol insertion, preventing undesirable insertions. . . . .	94
v1.0.0.	
General: Compatibility with eledmac. Change name to eledpar. . . . .	1
Debug in lineation by pstart . . . . .	38
v1.0.1.	
General: Correction on \numberonlyfirstinline with lineation by pstart or by page. .	1
v1.1.0.	
\pstartR: Add \labelpstarttrue (from eledmac). . . . .	63
General: The shiftedverses options becomes shiftedpstarts. . . . .	1
v1.1.1.	
\pstartR: Correct \pstartR bug introduced by 1.1. . . . .	63
v1.1.2.	
\affixside@noteR: Remove spurious space between line number and line content . .	93
v1.2.0.	
General: Support for \led<section> commands in parallel texts. . . . .	1
v1.2.1.	
\set@sectcountR: For the right section, the counter is defined only once. . . . .	36
v1.3.0.	
\edtext: Manage RTL language. . . . .	54
v1.3.2.	
General: Debug with some classes. . . . .	1

v1.3.3.	General: Debugging the left notes of the right column. . . . .	93
v1.3.4.	General: Allow use of commands in sidenotes, as introduced by eledmac 1.0. . . . .	93
v1.4.0.	General: Added <code>\do@insidelineLhook</code> and <code>\do@insidelineRhook</code> . . . . .	73
v1.4.1.	General: Enable the use of <code>stanzaindentsrepetition</code> within <code>astanza</code> environment. . . . .	95
v1.4.3.	<code>\inserthangingsymbolR</code> : Hanging verse is no longer automatically flush right. . . . .	94
	<code>\pendL</code> : Spurious spaces in <code>\pendL</code> . . . . .	67
	<code>\pendR</code> : Spurious spaces in <code>\pstartR</code> . . . . .	68
	<code>\pstartR</code> : Spurious spaces in <code>\pstartL</code> and <code>\pstartR</code> . . . . .	63
	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. . . . .	41
	General: Add, as in eledmac, features to manage page breaks. . . . .	1
v1.6.0.	General: Add tool and documentation for parallel ledgroups . . . . .	24
v1.7.0.	General: Add, as in eledmac, features to make crossrefs with <code>pstart</code> numbers. . . . .	1
v1.8.0.	<code>\Columns</code> : Modify <code>\Columns</code> to enable to add section's title. . . . .	102
	Suppress <code>\l@dchecklang</code> from <code>\Columns</code> . . . . .	103
	<code>\Pages</code> : Modify <code>\Pages</code> to enable to add section's title. . . . .	113
	<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. . . . .	97
	<code>\pendL</code> : As in eledmac, <code>\pendL</code> can have an optional argument. . . . .	67
	<code>\pendR</code> : As in eledmac, <code>\pendR</code> can have an optional argument. . . . .	68
	<code>\print@columnseparator</code> : Move some code of <code>\Columns</code> to <code>\print@columnseparator</code> . . . . .	105
	<code>\pstartR</code> : As in eledmac, <code>\pendL</code> and <code>\pendR</code> can have an optional argument. . . . .	63
	<code>\sidenotemarginR</code> : <code>\sidenotemargin</code> is now directly defined in eledmac to be able to manage eledpar. . . . .	92
	Add <code>\sidenotemargin*</code> . . . . .	92
	<code>\theledlanguageR</code> : Correct left/right language setting with polyglossia. . . . .	98
	General: <code>\beginnumbering</code> is defined only on eledmac, not on eledpar. . . . .	34
	<code>\l@dlsnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in eledmac. . . . .	93
	Add <code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code> . . . . .	10
	Add <code>\columnspostion</code> . . . . .	10
	Add, as in eledmac, new system of sectioning commands. . . . .	1
	Add, as in eledmac, option to insert something after <code>\pends</code> / verses. . . . .	1
	Add, as in eledmac, option to insert something between <code>\pstarts</code> / verse. . . . .	1
	Change <code>\do@lineR</code> and <code>\do@lineR</code> to allow new sectioning commands. . . . .	70
	Compatibility with <code>musixtex</code> . . . . .	1
	Debug eledmac sectioning command after using <code>\resumenumbering</code> . . . . .	1
	New sectioning commands, as in eledmac. . . . .	25

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.	97
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 (?).	70
<code>\do@lineR</code> : Fix a bug with critical notes at the beginning of a page, added by v1.8.0 (?).	73
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>	54
<code>\lineation*</code> : Add <code>\lineation*</code>	39
<code>\reledpar@error</code> : Errors specific to <code>eledpar</code> send to <code>eledpar</code> handbook	29
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> .	113
<code>\doinsidelineRhook</code> : Added <code>\dolineLhook</code> , <code>\dolineRhook</code> , <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code>	73
<code>\pausenumbering@page@numR</code> : Debug <code>\resumenumberingR</code>	37
General: Add <code>\noeledxxx</code> , as in <code>eledmac</code>	1
v1.9.0.	
<code>\ifwidthliketwocolumns</code> : Added <code>widthliketwocolumns</code> option	28
<code>\theledlanguageR</code> : Debug left/right language switching with <code>polyglossia</code> . Do not write in <code>.aux</code> file when setting left/right lines.	98
General: Add <code>\AtBeginPairs</code> macro.	10
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>	29
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).	113
Debug wrong parallel pages synchronization when an <code>\edtext</code> falls across two pages.	113
General: Compatibility with <code>\AtEveryPstart</code> and <code>\AtEveryPend</code>	1
Restore critical notes in <code>\eledsection</code> in parallel columns (this bug was added in 1.8.2).	1
v1.10.1.	
<code>\next@line@list@stuffR</code> : Revert modification of 1.4.2, which makes bugs with numbering. Leave vertical mode to solve spurious space before <code>minipage</code> .	53
v1.11.0.	
<code>\edtext</code> : <code>\critext</code> and <code>\edtext</code> are now defined only in <code>eledmac</code> .	54
General: Compatibility of standard footnotes with some <code>biblatex</code> styles.	1
v1.12.0.	
<code>\Columns</code> : Add <code>\l@dprintingcolumnstrue</code>	102
<code>\Pages</code> : Add <code>\l@dprintingpagestrue</code>	113
<code>\edlabel</code> : <code>\edlabel</code> and <code>\edindex</code> works now with <code>hyperref</code> when using <code>eledpar</code> .	91
<code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code>	91
<code>\print@eledsectionL</code> : Compatibility with <code>Lua<math>\TeX</math></code> RTL languages.	72
<code>\print@eledsectionR</code> : Compatibility with <code>Lua<math>\TeX</math></code> RTL languages.	75
<code>\print@lineL</code> : Compatibility with <code>Lua<math>\TeX</math></code> RTL languages.	71

General: Compatibility with Lua $\TeX$ RTL languages. . . . .	1
v1.12.1.	
\print@eledsectionL: Fix bug with Lua $\TeX$ RTL \eledsection. . . . .	72
v1.13.0.	
\Pages: Prevent false overfull hboxes when using \Pages outside of pages environment. . . . .	114
When using shiftedpstarts option, a \l@leftbox with a null height will advance the \pagetotal in any case. . . . .	113
\clearl@drighthouse: Use \newpage instead of \clearpage. . . . .	122
\ifledRcol: Remove false boolean settings which are not needed. . . . .	29
General: Enable the use of optional argument of & in astanza environment. . . . .	95
Fix a 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) . . . . .	113
\correct@footinsX@box: Call \correct@footinsX@box and \correct@Xfootins@box directly in \print@notesX@forpages and \print@Xnotes@forpages. . . . .	85
Correct \correct@footinsX@box and \correct@Xfootins@box . . . . .	85
v1.14.0.	
General: Fix a 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 . . . . .	77
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting when a verse is hanging . . . . .	94
\lineationR: As \lineation, \lineationR automatically set the \pstartinfootnote. . . . .	38
\n@num: \n@num defined only one time for both Eledmac and Eledpar. . . . .	49
\skipnumbering: \skipnumbering defined only one time for both Eledmac and Eledpar . . . . .	54
General: Add \AtEveryPstartCall. . . . .	1
Add sameparallepagenumber option. . . . .	15
Fix vertical spurious space before right \eledchapter (bug added in v1.13.0). . . . .	1
Prevent vertical space when using \AtEveryPstart or \AtEveryPend with a command which prints nothing . . . . .	1
v1.16.0.	
\newseries@par: Fix a bug with \onlysideX. . . . .	55
General: Error message when calling \Pages inside pages environment and \Columns inside pairs environment. . . . .	1
Error message when starting a Leftside/a Rightside while the previous one has not been yet typeset. . . . .	1
Error message when using \beginnumbering...\endnumbering without \pstart. . . . .	1
Fix a bug with nofamiliar / nocritical option of eledmac. . . . .	1
New package option sameparallepagenumber 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 a 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>\hidenumbering</code> . . . . .	19
v2.0.0.	
<code>\@adv: \@adv</code> defined only in <code>reledmac</code> . . . . .	47
<code>\@lab: \@lab</code> defined only in <code>eledmac</code> . . . . .	91
<code>\@ref@regR: \@ref</code> defined only in <code>reledmac</code> , code specific to right side moved in <code>\ref@regR</code> . . . . .	49
<code>\@set: \@set</code> defined only in <code>reledmac</code> . . . . .	47
<code>\advanceline: \advanceline</code> defined only in <code>reledmac</code> . . . . .	54
<code>\bbl@set@language: Patch \bbl@set@language</code> instead of redefining it . . . . .	97
<code>\do@lockonR: \do@lockon</code> defined only in <code>reledmac</code> . . . . .	48
<code>\endlock: \startlock</code> and <code>\endlock</code> defined only in <code>reledmac</code> . . . . .	54
<code>\endsub: \startsub</code> and <code>\endsub</code> defined only in <code>reledmac</code> . . . . .	54
<code>\fix@page: \fix@page</code> is defined only once in <code>reledmac</code> . . . . .	47
<code>\l@d@set: \l@d@set</code> defined only in <code>reledmac</code> . . . . .	48
<code>\line@marginR: \linenummargin</code> now defined only once time in <code>reledmac</code> . . . . .	39
<code>\page@action: \page@action</code> defined only in <code>reledmac</code> . . . . .	48
<code>\read@linelist: \read@linelist</code> is defined only once time in <code>\reledmac</code> . . . . .	46
<code>\set@line: \set@line</code> defined only in <code>reledmac</code> . . . . .	54
<code>\set@line@action: \set@line@action</code> defined only in <code>reledmac</code> . . . . .	48
<code>\setline: \setline</code> defined only in <code>reledmac</code> . . . . .	54
<code>\setlinenum: \setlinenum</code> defined only in <code>reledmac</code> . . . . .	54
<code>\skip@lockoff: \do@lockoff</code> defined only in <code>reledmac</code> . . . . .	48
<code>\sub@action: \sub@action</code> defined only in <code>reledmac</code> . . . . .	48
<code>\sublinenumincrement*: \firstlinenum, \linenumincrement,</code> <code>\firstsublinenum, \sublinenumincrement</code> are now defined only in <code>reledmac</code> . . . . .	41
General: <code>\@nl</code> is now defined only in <code>reledmac</code> . . . . .	46
<code>\ifbypage@</code> and <code>\ifbypstart@R</code> defined in <code>eledmac</code> . . . . .	38
Fix some bugs with <code>sameparallelpagenunder</code> option. . . . .	1
Many code refactored and moved to <code>reledmac</code> . . . . .	1
Package's name becomes <code>reledpar</code> . . . . .	1
Totally new implementation of <code>sameparallelpagenunder</code> option. . . . .	1
<code>chapterinpages: Deleting the old system of managing parallel chapter, keep only the         new one with \patchcmd</code> . . . . .	60
v2.1.0.	
General: Fix a bug when using <code>\eledsection</code> and related on right pages when page width is short. . . . .	1
Fix a bug when using <code>\pagenumbering</code> with <code>memoir</code> (bug added in v2.0.0). . . . .	1
Fix a bug with <code>\setparledgroupnotespacing</code> with the <code>shiftedpstarts</code> option. . . . .	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code> . . . . .	1
Options to custom empty right page before <code>\Pages</code> . . . . .	1

v2.2.0.	
General: <code>astanza</code> environment can take an optional argument, which will be the optional argument of <code>\pstart</code> started by this environment. . . . .	1
New tools to number stanza . . . . .	1
v2.2.1.	
General: Fix a bug with optional argument of last left <code>\pend</code> . . . . .	1
v2.3.0.	
<code>\Pages</code> : Fix a bug when calling <code>\Columns</code> after a <code>\Pages</code> (bug added in v1.13.0). . . . .	114
General: Change some internal codes in order to provide compatibility with $\LaTeX$ release of october 2015 . . . . .	1
Fix a bug with title number in parallel columns . . . . .	1
New line setting command suffixed by R to set only the right side. . . . .	1
v2.4.0.	
<code>\ledstrutR</code> : Deleted <code>\ledtrutL</code> and <code>\ledstrutR</code> . . . . .	121
Fix a bug with dotted letter . . . . .	121
General: New way of (not) synchronizing the parallel pages. . . . .	1
Option to switch to <code>\mainmatter</code> when calling <code>\Pages</code> . . . . .	1
v2.5.0.	
General: Disable empty lines as paragraph in <code>astanza</code> . . . . .	1
Fix a bug introduced in v1.15.0 which made hanging indentation in verse not work anymore. . . . .	1
New commands <code>\linenummarginR</code> and <code>\linenummargin*</code> . . . . .	1
v2.5.1.	
General: Fix spurious space when using optional argument of <code>astanza</code> environment (introduced in v2.5.0). . . . .	1
v2.5.2.	
General: Fix a bug introduced in v2.5.0 with <code>\linenummargin</code> , <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> . . . . .	1
v2.6.0.	
<code>\l@dmake@labelsR</code> : <code>\@Rlineflag</code> is not stored directly after the line number, but as a fifth argument of <code>\the@labelX</code> . Can be retrieved by <code>\xflagref</code> . . . . .	91
General: <code>\Xlineflag</code> and <code>\Xendlineflag</code> added . . . . .	1
<code>\printlinesR</code> deleted . . . . .	1
Error message when calling <code>\Pages</code> or <code>\Columns</code> without previous <code>pages</code> or <code>pairs</code> environment. . . . .	1
Fix a bug with footnote numbering when using the same series of familiar footnotes on both sides. . . . .	1
Fix a 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 a 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> . . . . .	55
General: Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.6.0). . . . .	1



Fix a 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 a 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 a 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	77
General: Add new tools to make apparatuses of manuscripts	1
v2.11.0.	
<code>\correct@footinsX@box</code> : Clarification in the handbook about the use of <code>\Xonlyside</code> and <code>onlysideX</code> and error message if misuse.	85
New implementation of <code>\Xonlyside</code> and <code>\onlysideX</code> hooks, prevent trouble with vertical spacing.	85
General: Compatibility with <code>reledmac</code> 's <code>auxdir</code> option.	1
v2.12.0.	
General: Add <code>\edtextlater</code> and <code>\edtextnow</code> .	1, 88
Fix a bug with <code>\AtBeginPairs</code> .	1
Fix a bug with <code>\lineationR</code> .	1
v2.13.0.	
General: Compatibility with the new features of <code>reledmac</code> 2.15.0	1
v2.13.1.	
General: Simplification of the code concerning the sectioning command	1
v2.14.0.	
General: Add <code>\linenumberlistR</code> , equivalent to <code>\linenumberlist</code> for the right side.	1
Error message when left / right side are not defined.	1
More explicit error message when the stanza indentation is not defined.	1
New commands to have specific line margins for text running in parallel columns	1
When indexing texts in sidenotes with <code>\edtext</code> , referring to the line number where the sidenote is called.	1
v2.14.1.	
General: Fix a bug when using <code>\markboth</code> on <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code> .	1

Fix a bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.13.1).	1
v2.14.2.	
General: Fix spurious space with <code>\edtextlater</code> .	1
v2.14.3.	
General: Fix a bug in <code>\edtextnow</code> .	1
v2.14.4.	
General: Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code> .	1
v2.15.0.	
General: Add <code>movecolumnsonrightpage</code> option.	1
Fix a bug about space before sectioning commands after a pairs environment (bug added in v. 2.14.4).	1
Fix a bug with <code>\msdata</code> when using multiple <code>\beginnumbering... \endnumbering</code> .	1
Fix a bug with page break when using <code>\columns</code> (bug added in v. 2.14.1).	1
New macro <code>\sidenotemarginR</code> , fix default value to right.	1
v2.16.0.	
General: Compatibility with <code>\setmsdataposition</code> .	1
v2.16.1.	
General: Restore sidenotes on left side, deleted by mistake in v2.16.0.	1
v2.16.2.	
General: Change log message when numbered files still don't exist, in order to improve compatibility with <code>latexmk</code> .	1
v2.16.3.	
General: Fix a bug when setting <code>\aftercolumnseparator</code> .	1
v2.17.0.	
General: Change code for compatibility with <code>reledmac</code> 2.18.0.	1
v2.17.1.	
General: Fix a bug with <code>prevpgstyle</code> option when not using <code>prevpgnotnumbered</code> .	1
v2.17.2.	
<code>\Pages</code> : Fix a bug when changing <code>\Lcolwidth</code> and <code>\Rcolwidth</code> in <code>pages</code> environment.	114
v2.17.3.	
General: Fix a bug when using multiple sidenotes on the same line, on right side.	1
v2.17.4.	
General: Fix bugs with sublines in parallel typesetting.	1
v2.18.0.	
General: Compatibility with v2.20.0 of <code>reledmac</code> .	1
v2.19.0.	
General: Compatibility with v2.22.0 of <code>reledmac</code> .	1
v2.19.1.	
General: Fix a bug with <code>\edtext</code> in sectioning command with <code>shiftedpstarts</code> mode.	1
v2.19.2.	
General: <code>continuousnumberingwithcolumns</code> also works for the <code>pstart</code> number.	1
v2.19.3.	
General: Do not print footnotes at the first run.	1
Fix a bug with <code>nopbinverse</code> when using consecutive <code>\Pages</code> .	1
v2.19.4.	
General: Optimisation of the code in order to not print footnotes at the first run.	1
v2.19.5.	
General: Fix a bug added in <code>astanza</code> in v. 2.18.0.	1

v2.20.0.	
\next@line@list@stuffR: Add \next@line@list@stuffR internal hook. . . . .	53
General: Add \linenumOnlyPagesForColumns and	
\linenumOnlyPagesForColumnsR to tell to print the line numbers associated with	
a column only on some specific pages (for example, only on right pages). . . . .	1
Add sidenotesmarginpage option to make the sidenotes refer to page margins and	
not column margins. . . . .	1
Fix a bug with \continuousnumberingwithcolumns when using multiple series of	
continued numbering text. Now, the line number are correctly reset at each	
\beginnumbering. . . . .	1
Fix a bug with continuousnumberingwithcolumns and inner / outer notes and line	
numbering. . . . .	1
Fix a bug with continuousnumberingwithcolumns when a \beginnumbering is	
called after another \beginnumbering... \endnumbering. . . . .	1
Fix a bug with widthliketwocolumns when manually setting	
\beforecolumnseparator and \aftercolumnseparator. . . . .	1
Fix compatibility between continuousnumberingwithcolumns and	
\numberpstartfalse. . . . .	1
Fix incompatibility between \lineation{page} and	
continuousnumberingwithcolumns option . . . . .	1
v2.20.1.	
General: Fix bug with continuousnumberingwithcolumns when left column and	
right column have not the same size. . . . .	1
v2.20.2.	
General: Fix bugs with continuousnumberingwithcolumns when going to a new page	1
v2.21.0.	
General: Add compatibility with new features of reledmac 2.26.0 . . . . .	1
v2.21.1.	
General: Fix bug with \linenumberannotation . . . . .	1
v2.21.2.	
General: Fix bug with \Xpstart and \edtextlater . . . . .	1
v2.21.3.	
General: Fix bug when using \numberpstarttrue and alternating between manual	
\pstart and automatic \pstart produced by \autopar . . . . .	1
v2.21.4.	
General: Fix bug with using \eledsection and related when alternate normal	
typesetting and parallel typesetting . . . . .	1
v2.21.5.	
General: Compatibility with reledmac v. 2.26.6 . . . . .	1
v2.21.6.	
General: Fix page number in footnote and endnotes when using	
sameparallelpagenumber option . . . . .	1
v2.21.7.	
General: Compatibility with reledmac 2.26.7 . . . . .	1
v2.21.8.	
General: Fix bug with critical footnotes line number when alternating between	
\numberlinefalse and \numberlinetrue. . . . .	1
Fix bug with marginal line number in parallel typesetting when alternating between	
\numberlinefalse and \numberlinetrue. . . . .	1

Workaround with a bug of $\LaTeX$ 2018/04/01 patch level 3 to generate the .sty file from a .ins file	1
v2.21.9.	
General: Fix (little) bug when using calc package.	1
v2.22.0.	
General: Compatibility with \doinsidethislinehook macro.	1
v2.22.1.	
General: Really reset line number annotation at each line	1
v2.22.2.	
General: Fix bug with \Xonlyside and \onlysideX	1
v2.22.3.	
General: When using \parindentX or \Xparindent and stanza, use the normal indentation in the footnote, not the verse indentation.	1
v2.22.4.	
\theledlanguageR: Provide compatibility with the latest version of polyglossia (v1.45).	98
v2.23.0.	
\theledlanguageR: Restore compatibility with versions of polyglossia prior to v1.45.	98
General: Add \Leftpagehook and \Rightpagehook.	1
Add hanging symbol feature to astanza.	1
v2.23.1.	
General: Fix bug when using polyglossia v1.45 or later.	1
v2.24.0.	
General: Add antilabe feature.	1
Add startstanzaindentsrepetition feature.	1
v2.24.1.	
General: Fix incompatibility between lineation by \pstart and (a)stanza.	1
Makes \numberstantzatrue working with astanza.	1
v2.24.2.	
General: Fix some bugs with lineation by page, when a numbered section starts at the very beginning of page.	1
v2.24.3.	
General: Fix bug with hanging verse in parallel typesetting.	1
v2.25.0.	
General: Add compatibility with \linenumannotationothersidetrue of reledmac.	1
v2.25.1.	
General: \ifsublines@R migrated to reledmac	1
v2.25.2.	
General: Fix incompatibility with babel v.3.61	1