

PW added following as the definitions are at some unknown elsewhere

# 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	6
<b>2 Options</b>	<b>7</b>
2.1 Synchronization’s options	7
2.2 Other options	7
<b>3 General</b>	<b>8</b>
<b>4 Parallel columns</b>	<b>9</b>
4.1 Basic use	9
4.2 Setting	9
4.2.1 Column’s width	9
4.2.2 Column’s separator	9
4.2.3 Column’s positions	10

---

\*This file (reledpar.dtx) has version number v2.25.10, last revised 2025/05/13.

<sup>†</sup>maieul at maieul dot net

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

<i>Contents</i>	3
4.2.4 Mixing two columns and one column texts . . . . .	10
4.2.5 Schemas of column typesetting . . . . .	10
4.2.6 Continuous numbering between single and double column texts . . . . .	10
4.3 Note about <code>\AtEveryPstart*</code> and <code>\AtEveryPend*</code> . . . . .	11
<b>5 Keeping translation in the outside/inside column</b>	<b>11</b>
<b>6 Facing pages</b>	<b>12</b>
6.1 Basic usage . . . . .	12
6.2 Setting . . . . .	12
6.2.1 Text width . . . . .	12
6.2.2 Way of synchronizing . . . . .	12
6.2.3 Page number . . . . .	14
6.2.4 Page breaking . . . . .	14
6.2.5 Right page before <code>\Pages</code> . . . . .	14
6.2.6 Notes about <code>\mainmatter</code> . . . . .	14
6.3 Critical and familiar footnotes . . . . .	14
6.3.1 Notes height setting . . . . .	14
6.3.2 About the numbering of familiar footnotes . . . . .	15
6.3.3 Using <code>perpage</code> package . . . . .	15
6.3.4 Notes for one side only . . . . .	15
6.3.5 Familiar notes called on the right side, but to be printed on the left side . . . . .	16
6.4 Critical notes called on leftside but printed on right side . . . . .	16
6.5 Using line flag . . . . .	16
<b>7 Left and right texts</b>	<b>17</b>
7.1 Environments . . . . .	17
7.2 Numbering text lines and paragraphs . . . . .	17
7.3 First line number and line number increment . . . . .	18
7.4 Lineation system . . . . .	18
7.5 Line number style . . . . .	18
7.6 Line number margin . . . . .	19
7.7 Line flag . . . . .	19
7.8 Chunks . . . . .	20
7.9 <code>\AtEveryPstart</code> and <code>\AtEveryPstartCall</code> . . . . .	20
7.10 Language setting . . . . .	20
7.11 Executing code at each line . . . . .	21
7.12 Executing code at each page . . . . .	21
<b>8 Verse</b>	<b>21</b>
<b>9 Side notes</b>	<b>22</b>
<b>10 Parallel ledgroups</b>	<b>22</b>
10.1 General . . . . .	22
10.2 Parallel ledgroups and <code>setspace</code> package . . . . .	23
<b>11 Sectioning commands</b>	<b>23</b>
<b>12 Notes about page number</b>	<b>23</b>

<b>I Implementation overview</b>	<b>24</b>
<b>II Preliminaries</b>	<b>24</b>
II.1 Package's meta-data	24
II.2 Package's requirement	24
II.3 Package's options	24
II.4 Package's options	25
II.4.1 Synchronization's options	25
II.4.2 Position options	26
II.4.3 Other options	26
II.5 Determining side and category of parallel processing	27
II.6 Text's width	27
II.7 Messages	27
II.8 Naming macros	30
<b>III Sectioning commands</b>	<b>31</b>
<b>IV Line counting</b>	<b>35</b>
IV.1 Setting lineation reset	35
IV.2 Setting line number margin	36
IV.3 Setting lineation start and step	38
IV.4 Setting line flag	39
IV.5 Setting line number style	39
IV.6 Print marginal line number	40
IV.7 Line-number counters and lists	41
IV.7.1 Correspond to those in <code>reledmac</code> for regular or left text	41
IV.7.2 Specific to <code>reledpar</code>	42
IV.8 Reading the line-list file	42
IV.9 Commands within the line-list file	42
IV.10 Writing to the line-list file	48
<b>V Marking text for notes</b>	<b>50</b>
V.1 Specific hooks and commands for notes	51
V.1.1 Notes to be printed on one side only	51
V.1.2 Tools specific to familiar footnotes	51
V.1.3 Get correct footnote number	53
V.2 Create hooks	53
V.3 Init standards series (A,B,C,D,E,Z)	54
V.4 Tools specific to $\LaTeX$ 's classical footnotes	54
<b>VI Pstart numbers dumping and restoration</b>	<b>54</b>
<b>VII Parallel environments</b>	<b>55</b>
<b>VIII Paragraph decomposition and reassembly</b>	<b>58</b>
VIII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	58
VIII.2 Processing one line	64
VIII.3 Line and page number computation	70
VIII.4 Line number printing	73
VIII.5 Pstart number printing in side	75
VIII.6 Add insertions to the vertical list	77

<i>Contents</i>	5
VIII.7 Penalties . . . . .	78
VIII.8 Printing leftover notes . . . . .	78
<b>IX Footnotes</b>	<b>79</b>
IX.1 Footnotes output specific to <code>\Pages</code> . . . . .	79
IX.2 Critical footnote printed in right side . . . . .	82
<b>X Cross referencing</b>	<b>84</b>
<b>XI Sidenotes</b>	<b>85</b>
<b>XII Verse</b>	<b>87</b>
<b>XIII Fixing babel and polyglossia</b>	<b>89</b>
<b>XIV Counts and boxes for parallel texts</b>	<b>92</b>
<b>XV Checking text to be processed</b>	<b>93</b>
<b>XVI Parallel columns</b>	<b>94</b>
<b>XVII Parallel pages</b>	<b>105</b>
XVII.1 Specific counters . . . . .	105
XVII.2 Main macro . . . . .	105
XVII.3 Ensure all notes are printed at the end of parallel pages . . . . .	112
XVII.4 Struts . . . . .	112
XVII.5 Page clearing . . . . .	112
XVII.6 Lines managing . . . . .	113
XVII.7 Page break managing . . . . .	115
XVII.8 Getting boxes content . . . . .	118
<b>XVIII The mainmatter option of <code>\Pages</code></b>	<b>122</b>
<b>XIX Sections' titles' commands</b>	<b>122</b>
<b>XX Page break/no page break, depending on the specific line</b>	<b>123</b>
<b>XXI Parallel ledgroup</b>	<b>124</b>
<b>XXII Compatibility with <code>eledmac</code></b>	<b>127</b>
<b>XXIII The End</b>	<b>127</b>
<b>Appendix A Some things to do when changing version</b>	<b>128</b>
A.1 Migration to <code>eledpar 1.4.3</code> . . . . .	128
A.2 Migration from <code>eledpar</code> to <code>reledpar</code> . . . . .	128
A.2.1 Deprecated options . . . . .	128
A.2.2 <code>\renewcommand</code> replaced with <code>command</code> . . . . .	128
A.2.3 Commands the names of which have changed . . . . .	129
A.3 Migration to <code>reledpar 2.2.0</code> . . . . .	129
A.4 Migration to <code>reledpar 2.3.0</code> . . . . .	129
A.5 Migration to <code>reledpar 2.4.0</code> . . . . .	129

A.6 Migration to reledpar 2.5.0 . . . . .	129
A.7 Migration to reledpar 2.6.0 . . . . .	129
A.8 Migration to reledpar 2.6.1 . . . . .	129
A.9 Migration to reledpar 2.14.0 . . . . .	129
A.10 Migration to reledpar 2.14.1 . . . . .	130
A.11 Migration to reledpar v. 2.17.2 . . . . .	130
A.12 Migration to reledpar v. 2.24.3 . . . . .	130
<b>References</b>	<b>130</b>
<b>Index</b>	<b>130</b>
<b>Change History</b>	<b>148</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. 12 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. 10).

**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.

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

**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.

---

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

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

### 3 General

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

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

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

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

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

```
\maxchunks{5120}
```

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

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

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

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

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



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

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

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

#### 4.2.2 Column's separator

`\columnrulewidth` The macro `\columnseparator` is called between each left/right pair of lines. By default 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 `\setstanzaindent` outside the `Leftside` or `Rightside` environment.

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

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

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

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

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

### 4.2.4 Mixing two columns and one column texts

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

`\Xnoteswidthliketwocolumns` In most cases, you should use `\widthliketwocolumns` in combination with `\Xnoteswidthliketwo`  
`\notesXwidthliketwocolumns` 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. 20).

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.

`\Rcolwidth`

#### 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  $\text{\LaTeX}$  to be run several times. Therefore, it is advised to use in conjunction utilities such as `latexmk` to ensure that synchronization is complete.

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

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

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

The aim of the default setting is twofold:

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

- To ensure that left pages contain what is to be on left 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:

- 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 `sameparallelpagenumber` to have the same page number for both left and right side. In this case, this setting will apply only for pages typeset by `\Pages`, not for “normal” pages.

Please also read advising in [12 p. 23](#).

### 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. 23](#).

`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  $\text{\LaTeX}$ 's feature.

So, you MUST change the default setting via `\Xmaxhnotes` (for critical notes) and `\maxhnotesX` (for familiar notes). Both commands are explained in `reledmac` handbook (7.12.7 p. 53). 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}
\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[⟨s⟩]{⟨p⟩}` to set critical notes, and `\onlysideX[⟨s⟩]{⟨p⟩}` to set familiar notes. `⟨p⟩` must be set to

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

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  $\LaTeX$  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{<notecontent>}` in the left side. X is to be replaced by the series letter. We do this call in the left side after the word which matches up to the one in the right side after which we want to insert the actual footnote mark.

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

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

## 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{<lemma>}{<footnote command>}` command must be called on the same side as the lemma. It is similar to the standard `\edtext` command, but does not add the footnote immediately on the page, keeping it for later. `\edtextnow` must be called on the side on which you want to print the lemma, approximately at the point corresponding to the equivalent `\edtextlater` command. It will add the notes stored by the equivalent `\edtextlater` command.

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

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

## 6.5 Using line flag

`\Xlineflag` Use `\Xlineflag[<s>]` to add right line flag (7.7 p. 19) 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 T<sub>E</sub>X’s memory of previous texts and any associated notes, allowing longer apparent streams of parallel texts. The command should be applied to both left and right texts, and after making sure that all previous notes have been output. For example, along the lines of:

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

```
...
\end{pages}
```

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

You can 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` 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.

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

`\firstsublinenum` The lineation commands which finish by a R apply for right text. The lineation commands which are starred apply for both left and right texts. The lineation command which does not finish by a R and who are not starred apply for the left side. **However, these commands apply to right side when they are called inside a left environment. However, such features should not be used any more. The recommended practice is to add all setting commands to the preamble.**

`\firstlinenum*` The starred versions change both left and right numbering schemes.

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

```
\firstlinenumR
```

```
\linenumincrementR
```

```
\firstsublinenumR
```

```
\sublinenumincrementR
```

```
\lineationR
```

```
\lineation*
```

### 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. 21) 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` `\linenumberstyleR` is the equivalent of `\reledmac \linenumberstyle` for right text.

`\sublinenumberstyleR` `\sublinenumberstyleR` is the equivalent of `\reledmac \sublinenumberstyle` right

text. The starred version are for both side.

```
\sublinenumberstyle*
```

## 7.6 Line number margin

`\linenummarginR` `\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`.

`\linenummarginColumns` 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` 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).

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

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

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

`\autopar` The `\autopar` macro can be used, instead of manually inserting `\pstart... \pends`. Please read `reledmac`'s handbook (5.2.2 p. 18).

## 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. 19) 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`,  
`\dolineRhook` respectively for left and right side.  
`\doinsidelineLhook` `\doinsidelineLhook` and `\doinsidelineRhook` are the equivalent to the `reledmac`  
`\doinsidelineRhook` `\doinsidelinehook`, for the left and the right sides respectively.  
 About these two hooks, read `reledmac`'s handbook (5.6 p. 27).

## 7.12 Executing code at each page

`\Rightpagehook` When typesetting parallel pages, the `\Leftpagehook` and `\Rightpagehook` macros are  
`\Leftpagehook` executed each time `reledpar` switches to Left / Right page.  
 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` When using `\stanzanumtrue` (9.10 p. 58) 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. 59). To do it you have:

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

See the following example:

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

```

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

```

## 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 `sameparallepage` option (6.2.3 p. 14) or `prevpnotnumbered` option (6.2.5 p. 14), 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}[2025/05/13 v2.25.10 reledmac extension for
  parallel texts]%
4
5 %

```

### II.2 Package’s requirement

Few commands use `\xspace` command.

```

6 \RequirePackage{xspace}%
7 %

```

### II.3 Package’s options

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

```

8 \RequirePackage{xkeyval}
9 %

```



## II.4 Package's options

### II.4.1 Synchronization's options

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

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

With the `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.

```
\ifshiftedpstarts12 \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.

```
\ifadvancedshiftedpstarts18 \newif\ifadvancedshiftedpstarts
19 \DeclareOptionX{advancedshiftedpstarts}{%
20 \advancedshiftedpstartstrue%
21 \shiftedpstartstrue%
22 \apptocmd{\@par@this@sync@option}{advancedshifted}{-}{-}%
23 }%
24 %
```

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

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

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

```

\ifnosyncpstarts31 \newif\ifnosyncpstarts%
32 \DeclareOptionX{nosyncpstarts}{%
33   \shiftedpstartstrue%
34   \nomaxlinestru%
35   \nosyncpstartstrue%
36   \apptocmd{\@par@this@sync@option}{nosync}{-}{-}%
37 }%
38 %

```

#### II.4.2 Position options

`\ifmovecolumnpositiononrightpage` The `\ifmovecolumnpositiononrightpage` 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\ifmovecolumnpositiononrightpage%
40 \DeclareOptionX{movecolumnpositiononrightpage}{%
41   \movecolumnpositiononrightpagetrue%
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}%
53 %

```

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

```

54 \DeclareOptionX{sameparallelpagenumber}{\sameparallelpagenumbertrue}
55 \DeclareOptionX{prevpgnotnumbered}{\prevpgnotnumberedtrue}
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@pairing \ifl@pairing is set TRUE if we are processing parallel texts and \ifl@dpaging is
\ifl@dpaging also set TRUE if we are doing parallel pages. \ifledRcol is set TRUE if we are doing
\ifledRcol the right hand text. They are defined in reledmac.

```

## II.6 Text’s width

```

\lcolwidth The widths of the left and right parallel columns (or pages). The suffixed versions are not
\Rcolwidth 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@warning 70 \newcommand{\reledpar@warning}[1]{\PackageWarning{reledpar}{#1}}%
71 %

```

```

\led@err@TooManyPstarts 72 \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 2022/04/10 v1.55 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@InsideEnv103 \newcommand*{\led@err@Pages@InsideEnv}{%
\led@err@Columns@InsideEnv104   \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@WithoutEnv108 \newcommand*{\led@err@Pages@WithoutEnv}{%
\led@err@Columns@WithoutEnv109   \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 %

```

```

\newcommand*\led@err@Pages@WithoutRightside}{%
  \reledpar@error{\string\Pages\space called without any preceding `
  Rightside` environment (or environment without content)}{\@ehc}}%
\newcommand*\led@err@Pages@WithoutLeftside}{%
  \reledpar@error{\string\Pages\space called without any preceding `
  Leftside` environment (or environment without content)}{\@ehc}}%
\newcommand*\led@err@Columns@WithoutRightside}{%
  \reledpar@error{\string\Columns\space called without any preceding `
  Rightside` environment (or environment without content)}{\@ehc}}%
\newcommand*\led@err@Columns@WithoutLeftside}{%
  \reledpar@error{\string\Columns\space called without any preceding `
  Leftside` environment (or environment without content)}{\@ehc}}%
%

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

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

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

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

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

```

```

\led@error@edtext@later@now 145 \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@ChangeSyncOption 149 \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@rightside 153 \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@numbering 159 \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  $\TeX$  kernel provides `\@namedef` and `\@namuse` for defining and using macros that may have non-letters in their names. We need something similar here as we are going to need and use some numbered boxes and counters.

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

```

\setnamebox
\unhnamebox
\unvnamebox
\namebox
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.

```

\useusernamecount
189 \providecommand*{\newnamecount}[1]{%
190   \expandafter\newcount\csname #1\endcsname}
191 \providecommand*{\useusernamecount}[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   \begingroup
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` `\line@marginR` 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` `\linenummargin*` `\linenummarginR` applies directly for right side, while `\linenummargin*` applies for both side.

```

413 \newcommand{\linenummarginR}[1]{%
414     \l@dgetline@margin{#1}%

```

```

415 \ifnum\@l@tempcntb>\m@ne%
416 \global\line@marginR=\@l@tempcntb%
417 \fi%
418 }
419 \WithSuffix\newcommand\linenummargin*[1]{%
420 \l@dgetline@margin{#1}%
421 \ifnum\@l@tempcntb>\m@ne%
422 \global\line@marginR=\@l@tempcntb%
423 \global\line@margin=\@l@tempcntb%
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@tempcntb>\m@ne%
438 \global\line@margin@columns=\@l@tempcntb%
439 \fi%
440 }%
441
442 \WithSuffix\newcommand\linenummarginColumns*[1]{%
443 \l@dgetline@margin{#1}%
444 \ifnum\@l@tempcntb>\m@ne%
445 \global\line@margin@columns=\@l@tempcntb%
446 \global\line@margin@columnsR=\@l@tempcntb%
447 \fi%
448 }%
449
450 \newcommand{\linenummarginColumnsR}[1]{%
451 \l@dgetline@margin{#1}%
452 \ifnum\@l@tempcntb>\m@ne%
453 \global\line@margin@columnsR=\@l@tempcntb%
454 \fi%
455 }%
456
457 %

```

`\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
463 \def\linenum@OnlyPages@ForColumnsR{%
464 \newcommand{\linenumOnlyPagesForColumnsR}[1]{%
465 \gdef\linenum@OnlyPages@ForColumnsR{#1}%
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`.

```

479 \firstsublinenum
480 \sublinenumincrement
481 \firstlinenum*
482 \linenumincrement*
483 \firstsublinenum*
484 \sublinenumincrement*
485
486
487
488
489
490
491
492
493
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{annotR@the\absline@numR @the\section@numR}%
566       }%
567     }{}%
568   \fi%
569 }%
570
571 %
```



`\leftlinenumannotationR` and `\rightlinenumannotationR` 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` Here are the commands to create these lists:

```

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

```
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 }%
625 %
```

`\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
been marked by a \setlinenum
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` `\last@page@numR` holds the page number of the last right page. `\this@section@last@page@numR`  
`\this@section@last@page@numR` holds the page number of the last right page of the current section. Its value is modified  
`\fix@page` 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  
`\do@lockonR` 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@lockoff` `\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` marks the start of a passage, for creation of a footnote reference. It takes two arguments:  
`\@ref@regR`  
`\insert@countR`

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

```

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   \ifsublines@R \the\subline@numR \else 0\fi|%
757   \the\endpage@num|\the\endline@num|%
758   \ifsublines@R \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{\csname sw@list@edtext@tmp@\the\
@edtext@level\endcsname}%
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@parseR 59 \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 `\@pendR` for `\@pendR`. If needed, it resets line number. Both are defined in `reledmac`, but they are empty. They are really defined only in `reledpar`.

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

```

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   }%
824   {\csnumgdef{minpage@pstart@#1}{\@tmp}}
825 \csnumgdef{afterlines@pstart@#1R}{#3}%
826 }%
827 %

```

`\@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{notfirststrun@#1}%
842   \IfFileExists{\l@auxdir#1}%
843     {\global\toggletrue{notfirststrun@#1}}%
844     {\global\togglefalse{notfirststrun@#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@cc@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@page] [\thepage]}%
874   \fi%
875 }%
876 %
```

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

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

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

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

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

`\startlock` `\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 temporary 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 \global\protected@csxdef{hr@thefnmark#1}{\noexpand\@tmpa @\the\
csname footnote#1@series\endcsname}%
905 }%
906 %
```

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

```
907 \expandafter\newcommand\csname footnote#1mk\endcsname{%
908 %
```

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

```
909 \letcs{\@tmp}{footnote#1@mk}%
910 \gl@p\@tmp\to\@tmpa%
911 \global\cslet{footnote#1@mk}{\@tmp}%
912 %
```

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

```
913 \letcs{\old@footnote}{c@footnote#1}%
914 \setcounter{footnote#1}{\@tmpa}%
915 %
```

Define the footnote mark and print it

```

916     \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
917     \csuse{@footnotemark#1}%
918 %

```

Restore previous footnote counter and finally add space.

```

919     \setcounter{footnote#1}{\old@footnote}%
920     \xspace%
921 }%
922 %

```

End of tools specific to familiar notes.

```

923 \fi
924 %

```

End of `\newseries@par`.

```

925 }%
926 %

```

### 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.

```

927 \newcommand{\save@familiarfootnote@number}{%
928   \unless\ifnofamiliar@%
929     \def\do##1{\csxdef{saved@footnote##1}{\the\c@footnote##1\
endcsname}}%
930     \dolistloop{\@series}%
931     \fi%
932     \xdef\saved@footnote{\the\c@footnote}%
933 }
934 \newcommand{\get@familiarfootnote@number}{%
935   \unless\ifnofamiliar@%
936     \def\do##1{\setcounter{footnote##1@typeset}{\csuse{saved@footnote##1}}}
937   %
938   \dolistloop{\@series}%
939   \fi%
940   \setcounter{footnote@typeset}{\saved@footnote}%
941 }
942 %

```

## V.2 Create hooks

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

```

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

```

```
948
949
950 %
```

### 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`.

```
951 \newcommand{\init@series@par}{%
952   \def\do##1{\newseries@par{##1}}%
953   \dolistloop{\@series}%
954 }%
955 \init@series@par%
956 %
```

### 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. 51).

```
\footnote@reading57 \newcount\footnote@reading%
\footnote@typeset58 \newcounter{footnote@typeset}%
59 %
```

## 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@pc60 \list@create{\list@pstartL@pc}%
\list@pstartR@pc61 \list@create{\list@pstartR@pc}%
62 %
```

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 `\curname` construction).

```
\dump@pstartL@pc63 \def\dump@pstartL@pc{%
\dump@pstartR@pc64   \xright@appenditem{\the\c@pstartL}\to\list@pstartL@pc%
65   \global\cset{numberpstart@L\the\l@dnumpstartsL}{\ifnumberpstart}%
66   \global\cset{numberline@L\the\l@dnumpstartsL}{\ifnumberline}%
67   \global\cset{linenumannotationotherside@L\the\l@dnumpstartsL}{\
iflinenumannotationotherside}%
```

```

968 }%
969
970 \def\dump@pstartR@pc{%
971   \xright@appenditem{\the\c@pstartR}\to\list@pstartR@pc%
972   \global\cslet{numberpstart@R\the\l@dnumpstartsR}{\ifnumberpstart}%
973   \global\cslet{numberline@R\the\l@dnumpstartsR}{\ifnumberline}%
974   \global\cslet{linenumannotationotherside@R\the\l@dnumpstartsR}{\
iflinenumannotationotherside}%
975 }%
976
977 %

```

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

`\restore@pstartR@pc`

```

978 \def\restore@pstartL@pc{%
979   \ifx\list@pstartL@pc\empty\else%
980     \gl@p\list@pstartL@pc\to\@temp%
981     \global\c@pstartL=\@temp%
982   \fi%
983 }%
984 \def\restore@pstartR@pc{%
985   \ifx\list@pstartR@pc\empty\else%
986     \gl@p\list@pstartR@pc\to\@temp%
987     \global\c@pstartR=\@temp%
988   \fi%
989 }%
990 %

```

## 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.

```

991 \newenvironment{pairs}{%
992   \reledpar@init@language%
993   \l@dpairingtrue
994   \l@dpagingfalse
995   \initnumbering@quote
996   \save@familiarfootnote@number%
997   \if@ledgroup%
998     \get@familiarfootnote@number%
999   \fi%
1000   \save@section@number%
1001   \at@begin@pairs%
1002 }{%
1003   \l@dpairingfalse
1004 }
1005
1006 %

```

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

```

1007 \newcommand{\AtBeginPairs}[1]{\gdef\at@begin@pairs{#1}}%
1008 \def\at@begin@pairs{}%
1009
1010 %

```

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.

```

1011 \newenvironment{pages}{%
1012   \reledpar@init@language%
1013   \l@dpairingtrue
1014   \l@dpageingtrue
1015   \initnumbering@quote
1016   \save@familiarfootnote@number%
1017   \if@ledgroup%
1018     \get@familiarfootnote@number%
1019   \fi%
1020   \save@section@number%
1021   \Lcolwidth=\textwidth%
1022   \Rcolwidth=\textwidth%
1023 }{%
1024   \l@dpairingfalse
1025   \l@dpageingfalse
1026   \global\Lcolwidth@pages=\Lcolwidth%Does not use \setlength, in case of
calc package redefiniton (cf. https://github.com/latex3/latex2e/issues/58)
1027   \global\Rcolwidth@pages=\Rcolwidth%
1028 }
1029
1030 %

```

**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.

```

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

```



```

1045 \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname%
1046 \led@error@missing@numbering{Leftside}%
1047 \fi%
1048 \Leftsidehookend}
1049 %

```

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

```

\Leftsidehookend
\Rightsidehook
1050 \newcommand*\Leftsidehook{}
\Rightsidehookend 1051 \newcommand*\Leftsidehookend{}
1052 \newcommand*\Rightsidehook{}
1053 \newcommand*\Rightsidehookend{}
1054 %
1055 %

```

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

```

\Leftpagehook
1056 \newcommand*\Rightpagehook{}%
1057 \newcommand*\Leftpagehook{}%
1058 %

```

**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.

```

1059 \newenvironment{Rightside}{%
1060 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else%
1061 \led@err@Rightside@PreviousNotPrinted%
1062 \fi%
1063 \ledRcoltrue
1064 \let\beginnumbering\beginnumberingR
1065 \let\endnumbering\endnumberingR
1066 \let\pausenumbering\pausenumberingR
1067 \let\resumenumbering\resumenumberingR
1068 \let\memorydump\memorydumpR
1069 \let\thepstart\thepstartR
1070 \let\pstart\pstartR
1071 \let\pend\pendR
1072 \let\ledpb\ledpbR
1073 \let\lednopb\lednopbR
1074 \let\lineation\lineationR
1075 \Rightsidehook
1076 \let\old@startstanza\@startstanza%
1077 \def\@startstanza[##1][##2]{\global\instanzaRtrue\old@startstanza
[##1][##2]}%
1078 }{%
1079 \ledRcolfalse
1080 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
1081 \led@error@missing@numbering{Rightside}%
1082 \fi%
1083 \Rightsidehookend
1084 }
1085 %
1086 %

```

## 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.

```
1087 \newcount\num@linesR
1088 \newbox\one@lineR
1089 \newcount\par@lineR
1090 %
```

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

Beware: everything that occurs between `\pstart` and `\pend` is happening within a group; definitions must be global if you want them to survive past the end of the paragraph.

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

```
1091
1092 \newcounter{pstartL}
1093 \renewcommand{\thepstartL}{\bfseries\@arabic\c@pstartL}. }
1094 \newcounter{pstartR}
1095 \renewcommand{\thepstartR}{\bfseries\@arabic\c@pstartR}. }
1096
1097 \newcommandx*{\pstartL}[2][1,2,usedefault]{%
1098   \if@nbreak%
1099     \let\@oldnbreak\@nbreaktrue%
1100   \else%
1101     \let\@oldnbreak\@nbreakfalse%
1102   \fi%
1103   \@nbreaktrue%
1104   \ifluatex%
1105     \xdef\l@luatextextdir@L{\the\textdir}%
1106     \xdef\l@luatexpardir@L{\the\pardir}%
1107     \xdef\l@luatexbodydir@L{\the\bodydir}%
1108   \fi%
1109   \ifnumbering \else%
```

```

1110 \led@err@PstartNotNumbered%
1111 \beginnumbering%
1112 \fi%
1113 \ifnumberedpar@%
1114 \led@err@PstartInPstart%
1115 \pend%
1116 \fi%
1117 %

```

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

```

1118 \ifpstart@rtedL\else%
1119 \list@clear{\inserts@list}%
1120 \global\let\next@insert=\empty%
1121 \global\pstart@rtedLtrue%
1122 \fi%
1123 \begingroup\everypar{}%
1124 %

```

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.

```

1125 \global\advance\l@dnumpstartsL \@ne%
1126 \global\advance\pstarts@read@L \@ne%
1127 \ifnum\l@dnumpstartsL>\l@dc@maxchunks%
1128 \led@err@TooManyPstarts%
1129 \global\l@dnumpstartsL=\l@dc@maxchunks%
1130 \fi%
1131 \global\setnamebox{\l@dLcolrawbox\the\l@dnumpstartsL}=\vbox\bgroup%
1132 %

```

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.

```

1133 \l@dzeropenalties%
1134 \ifboolexpr{%
1135   bool{autopar}%
1136   and bool{by@autopar}%
1137 }%
1138 {}%
1139 {}%
1140 \ifnumberpstart%
1141   \ifsidepstartnum%
1142   \else%
1143     \thepstartL%
1144   \fi%
1145 \fi%
1146 }%m
1147 \@at@start@every@pstart%
1148 \hsize=\lcolwidth%
1149 \numberedpar@true%
1150 \iflabelpstart%
1151   \protected@edef\@currentlabel{\p@pstartL\thepstartL}%
1152 \fi%
1153 %

```

Dump the optional arguments

```

1154 \providetoggle{before@pstartL@the\l@dnumpstartsL @par}%
1155 \ifboolexpr{%
1156   test {\ifstrempy{#1}}%
1157   and test {\ifstrempy{#2}}%
1158 }%
1159 {%
1160   \ifat@every@pstart@star@%
1161   \global\togglefalse{before@pstartL@the\l@dnumpstartsL @par}%
1162   \else%
1163   \global\toggletrue{before@pstartL@the\l@dnumpstartsL @par}%
1164   \fi%
1165   \csgdef{before@pstartL@the\l@dnumpstartsL}{\at@every@pstart}%
1166 }%
1167 {%
1168   \ifstrempy{#1}{}%
1169   \global\toggletrue{before@pstartL@the\l@dnumpstartsL @par}%
1170   \csgdef{before@pstartL@the\l@dnumpstartsL}{\noindent#1}%
1171   }%
1172   \ifstrempy{#2}{}%
1173   {%
1174     \csgdef{before@pstartL@the\l@dnumpstartsL}{#2}%
1175     \global\togglefalse{before@pstartL@the\l@dnumpstartsL @par}%
1176     }%
1177   }%
1178   \at@every@pstart@call%
1179 %

```

Let \by@autopar to FALSE.

```

1180 \global\by@autoparfalse%
1181 %

```

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

```

1182 \ignorespaces%
1183 }
1184 %

```

The same for right side.

```

1185 \newcommandx*{\pstartR}[2][1,2,usedefault]{%
1186   \if@nbreak%
1187   \let\@oldnbreak\@nbreaktrue%
1188   \else%
1189   \let\@oldnbreak\@nbreakfalse%
1190   \fi%
1191   \@nbreaktrue%
1192   \ifluatex%
1193   \xdef\l@luatextextdir@R{\the\textdir}%
1194   \xdef\l@luatexpardir@R{\the\pardir}%
1195   \xdef\l@luatexbodydir@R{\the\bodydir}%
1196   \fi%
1197   \ifnumberingR \else%
1198   \led@err@PstartNotNumbered%
1199   \beginnumberingR%

```

```

1200 \fi%
1201 \ifnumberedpar@%
1202   \led@err@PstartInPstart%
1203   \pendR%
1204 \fi%
1205 \ifpstart@rtedR\else%
1206   \list@clear{\inserts@listR}%
1207   \global\let\next@insertR=\empty%
1208   \global\pstart@rtedRtrue%
1209 \fi%
1210 \begin@group\every@par{}%
1211 \global\advance\l@dnumpstartsR \@ne%
1212 \ifnum\l@dnumpstartsR>\l@dc@maxchunks%
1213   \led@err@TooManyPstarts%
1214   \global\l@dnumpstartsR=\l@dc@maxchunks%
1215 \fi%
1216 \global\setnamebox{\l@dc@Rcolrawbox\the\l@dnumpstartsR}=\vbox\bgroup%
1217   \l@dc@zeropenalties%
1218   \ifboolexpr{%
1219     bool{autopar}%
1220     and bool{by@autopar}%
1221   }%
1222   {}%
1223   {%
1224     \ifnumberpstart%
1225     \ifsidepstartnum\else%
1226       \thepstartR%
1227     \fi%
1228   \fi%
1229   }%
1230 \at@start@every@pstart%
1231 \hsize=\Rcolwidth%
1232 \numberedpar@true%
1233 \iflabelpstart%
1234   \protected@edef\@currentlabel{\p@pstartR\thepstartR}%
1235 \fi%
1236 \providetoggle{before@pstartR@the\l@dnumpstartsR @par}%
1237 \ifboolexpr{%
1238   test {\ifstrempy{#1}}%
1239   and test {\ifstrempy{#2}}%
1240 }%
1241 {%
1242   \ifat@every@pstart@star@%
1243   \global\togglefalse{before@pstartR@the\l@dnumpstartsR @par}%
1244   \else%
1245   \global\toggletrue{before@pstartR@the\l@dnumpstartsR @par}%
1246   \fi%
1247   \csgdef{before@pstartR@the\l@dnumpstartsR}{\at@every@pstart}%
1248 }%
1249 {%
1250   \ifstrempy{#1}{}%
1251   \global\toggletrue{before@pstartR@the\l@dnumpstartsR @par}%
1252   \csgdef{before@pstartR@the\l@dnumpstartsR}{\noindent#1}%
1253   }%

```

```

1254     \ifstrempy{#2}{}%
1255     {%
1256         \csgdef{before@pstartR@the\l@dnumpstartsR}{#2}%
1257         \global\togglefalse{before@pstartR@the\l@dnumpstartsR @par}%
1258     }%
1259     }%
1260     \at@every@pstart@call%
1261     \global\by@autoparfalse%
1262     \ignorespaces%
1263 }
1264 %

```

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

```

1265 \newcommandx*{\pendL}[2][1,2,usedefault]{%
1266     \ifnumbering \else%
1267         \led@err@PendNotNumbered%
1268     \fi%
1269     \ifnumberedpar@ \else%
1270         \led@err@PendNoPstart%
1271     \fi%
1272 %

```

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.

```

1273     \at@end@every@pend%
1274     \endgraf\global\num@lines=\prevgraf\egroup%
1275     \global\par@line=0%
1276     \get@firsthalf@antilabe@wd%
1277 %

```

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

```

1278     \endgroup%
1279     \ignorespaces%
1280     \@oldnbreak%
1281     \dump@pstartL@pc%
1282     \ifnumberpstart%
1283         \addtocounter{pstartL}{1}%
1284         \ifcontinuousnumberingwithcolumns%
1285             \addtocounter{pstart}{1}%
1286         \fi%
1287     \fi
1288     \parledgroup@beforenotes@save{L}%
1289 %

```

Dump content of the optional argument.

```

1290     \providetoggle{after@pendL@the\l@dnumpstartsL @par}%
1291     \ifboolexpr{%
1292         test {\ifstrempy{#1}}%
1293         and test {\ifstrempy{#2}}%
1294     }%
1295     {%
1296         \ifat@every@pend@star%

```

```

1297     \global\togglefalse{after@pendL@the\l@dnumpstartsL @par}%
1298     \else%
1299     \global\toggletrue{after@pendL@the\l@dnumpstartsL @par}%
1300     \fi%
1301     \csgdef{after@pendL@the\l@dnumpstartsL}{\at@every@pend}%
1302   }%
1303   {%
1304     \ifstrempy{#1}{}%
1305     \global\toggletrue{after@pendL@the\l@dnumpstartsL @par}%
1306     \csgdef{after@pendL@the\l@dnumpstartsL}{\noindent#1}%
1307   }%
1308   \ifstrempy{#2}{}%
1309   {%
1310     \csgdef{after@pendL@the\l@dnumpstartsL}{#2}%
1311     \global\togglefalse{after@pendL@the\l@dnumpstartsL @par}%
1312   }%
1313 }%
1314 }%
1315 %

```

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

```

1316 \newcommandx*{\pendR}[2][1,2,usedefault]{%
1317   \ifnumberingR \else%
1318     \led@err@PendNotNumbered%
1319   \fi%
1320   \ifnumberedpar@ \else%
1321     \led@err@PendNoPstart%
1322   \fi%
1323   \@at@end@every@pend%
1324   \endgraf\global\num@linesR=\prevgraf\egroup%
1325   \global\par@lineR=0%
1326   \get@firsthalf@antilabe@wd%
1327   \endgroup%
1328   \ignorespaces%
1329   \@oldnobreak%
1330   \dump@pstartR@pc%
1331   \ifnumberpstart%
1332     \addtocounter{pstartR}{1}%
1333   \fi%
1334   \parledgroup@beforenotes@save{R}%
1335   \providetoggle{after@pendR@the\l@dnumpstartsR @par}%
1336   \ifboolexpr{%
1337     test {\ifstrempy{#1}}%
1338     and test {\ifstrempy{#2}}%
1339   }%
1340   {%
1341     \ifat@every@pend@star%
1342     \global\togglefalse{after@pendR@the\l@dnumpstartsR @par}%
1343     \else%
1344     \global\toggletrue{after@pendR@the\l@dnumpstartsR @par}%
1345     \fi%
1346     \csgdef{after@pendR@the\l@dnumpstartsR}{\at@every@pend}}%
1347   {%

```

```

1348 \ifstrempy{#1}{-}{%
1349 \csgdef{after@pendR@the\l@dnumpstartsR}{\noindent#1}%
1350 \global\toggletrue{after@pendR@the\l@dnumpstartsR @par}%
1351 }%
1352 \ifstrempy{#2}{-}{%
1353 \csgdef{after@pendR@the\l@dnumpstartsR}{#2}%
1354 \global\togglefalse{after@pendR@the\l@dnumpstartsR @par}%
1355 }%
1356 }%
1357 }
1358
1359 %

```

**\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.

```

1360 \newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
1361 \gdef\at@every@pstart@call{}%
1362 %

```

**\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**

```

1363 \newif\ifprint@last@after@pendL%
1364 \newif\ifprint@last@after@pendR%
1365 %

```

## 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**

```

1366 \newbox\l@dleftbox
1367 \newbox\l@drightbox
1368
1369 %

```

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

**\countRline**

```

1370 \newcount\countLline
1371 \countLline \z@
1372 \newcount\countRline
1373 \countRline \z@
1374
1375 %

```

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

**\@donetotallinesL**

**\@donereallinesR**

**\@donetotallinesR**



```

1376 \newcount\@donereallinesL
1377 \newcount\@donetotallinesL
1378 \newcount\@donereallinesR
1379 \newcount\@donetotallinesR
1380
1381 %

```

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

```

1382 \newcommand*\do@lineL{%
1383   \letcs{\ifnumberpstart}{numberpstart@L\the\l@dpscl}%
1384   \letcs{\ifnumberline}{numberline@L\the\l@dpscl}%
1385   \letcs{\iflinenumannotationotherside}{linenumannotationotherside@L\the\
l@dpscl}%
1386   \advance\countLline \@ne%
1387   \ifvbox\namebox{1@dLcolrawbox\the\l@dpscl}%
1388     {\vbadness=10000%
1389     \splittopskip=\z@%
1390     \do@lineLhook%
1391     \l@emptyd@ta%
1392     \global\setbox\one@line=\vsplit\namebox{1@dLcolrawbox\the\l@dpscl}%
1393       to\baselineskip}%
1394   \ifStrEq{\splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startL}{}%
1395   \unvbox\one@line \global\setbox\one@line=\lastbox%
1396   \@writepageofparL%
1397   \getline@numL%
1398   \ifnum\@lock>\@ne%
1399     \inserthangingsymboltrue%
1400   \else%
1401     \inserthangingsymbolfalse%
1402   \fi%
1403   \setbox\l@dleftbox%
1404   \hb@xt@ \lcolwidth{%
1405     \ifl@dhidenum%
1406       \global\l@dhidenumfalse%
1407       \f@x@l@cks%
1408     \else%
1409       \affixline@num%
1410     \fi%
1411     \xifinlist{\the\pstarts@typeset@L}{\eled@sections@}%
1412     {\unless\ifshiftedpstarts%
1413       \add@apparatusL%
1414     \fi%
1415     }%
1416     {\print@lineL}%
1417   }%
1418   \add@penaltiesL%
1419   \global\advance\@donereallinesL\@ne%
1420   \global\advance\@donetotallinesL\@ne%
1421 \else%
1422   \iflinenumLevenifblank
1423   \setbox\l@dleftbox \hb@xt@ \lcolwidth{%

```

```

1424 \new@lineL%
1425 \l@demptyd@ta%
1426 \getline@numL%
1427 \affixline@num%
1428 \l@dld@ta%
1429 \hspace*{\Lcolwidth}%
1430 \ledrlfill\l@drd@ta%
1431 }%
1432 \else%
1433 \setbox\l@dleftbox \hb@xt@ \Lcolwidth{%
1434 \ifcontinuousnumberingwithcolumns%
1435 \new@lineL%
1436 \getline@numL%
1437 \fi%
1438 \hspace*{\Lcolwidth}%
1439 }%
1440 \fi%
1441 \global\advance\@donetotallinesL\@ne%
1442 \fi%
1443 }%
1444
1445
1446 %

```

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

```

1447 \def\print@lineL{%
1448 \affixpstart@numL%
1449 \l@dld@ta%
1450 \add@apparatusL%
1451 \l@dlsn@te%
1452 \do@insidelineLhook%
1453 \csuse{insidethis@the\absline@num @the\section@num}%
1454 \global\csundef{insidethis@the\absline@num @the\section@num}%
1455 \hb@xt@ \Lcolwidth{\ledllfill\hb@xt@ \wd\one@line{%
1456 \ifluatex%
1457 \textdir\l@luatextextdir@L%
1458 \fi%
1459 \new@lineL%
1460 \inserthangingsymbolL%
1461 \l@dunhbox@line{\one@line}\ledrlfill\l@drd@ta%
1462 \l@drsn@te}}%
1463
1464 %

```

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

```

1465 \def\print@eledsectionL{%
1466 \disable@edindex%
1467 \addtocounter{pstartL}{-1}%
1468 \ifdefstring{@eledsectnotoc}{L}{\ledsectnotoc}{ }
1469 \ifdefstring{@eledsectmark}{L}{ }\ledsectnomark
1470 \numdef{\temp@}{\pstarts@typeset@L-1}%

```

```

1471 \xifinlist{\temp@}{\eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
1472 \@eled@sectioningtrue%
1473 \bgroup%
1474 \ifluatex%
1475 \textdir\l@luatextextdir@L%
1476 \pardir\l@luatexpardir@L%
1477 \bodydir\l@luatexbodydir@L%
1478 \ifdefstring{\l@luatextextdir@L}{TRT}{\@RTLtrue}{}%
1479 \fi%
1480 \csuse{eled@sectioning@the\pstarts@typeset@L}%
1481 \egroup%
1482 \@eled@sectioningfalse%
1483 \global\csundef{eled@sectioning@the\pstarts@typeset@L}%
1484 \ifRTL%
1485 \hspace{-3\paperwidth}%
1486 {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1487 \else%
1488 \hspace{3\paperwidth}%
1489 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1490 \fi%
1491 \vskip\eledsection@correcting@skip%
1492 \restore@edindex%
1493 }
1494
1495 %

```

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

```

1496 \newcommand{\add@apparatusL}{%
1497 \if@firstlineofpage%
1498 \set@Xtxtbeforenotes%
1499 \set@txtbeforenotesX%
1500 \global\@firstlineofpagefalse%
1501 \fi%
1502 \ifdefstring{\ms@data@position}{msdata-regular}{%
1503 \insert@msdata%
1504 \add@inserts%
1505 \add@Xgroupbyline%
1506 }{%
1507 \add@inserts%
1508 \add@Xgroupbyline%
1509 \insert@msdata%
1510 }%
1511 \affixside@note%
1512 }%
1513 %

```

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

**\dolineRhook**

**\doinsidelineLhook**

**\doinsidelineRhook**

```

1514 \newcommand*\dolineLhook}[1]{\gdef\do@lineLhook{#1}}%
1515 \newcommand*\dolineRhook}[1]{\gdef\do@lineRhook{#1}}%
1516 \newcommand*\doinsidelineLhook}[1]{\gdef\do@insidelineLhook{#1}}%

```

```

1517 \newcommand*{\doinsidelineRhook}[1]{\gdef\do@insidelineRhook{#1}}%
1518
1519 %

```

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

```

\do@lineRhook
\do@insidelineLhook
\do@insidelineRhook

```

```

1520 \newcommand*{\do@lineLhook}{}
1521 \newcommand*{\do@lineRhook}{}
1522 \newcommand*{\do@insidelineLhook}{}
1523 \newcommand*{\do@insidelineRhook}{}
1524
1525 %

```

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

```

1526 \newcommand*{\do@lineR}{%
1527   \let\linenumrepL\linenumrep%
1528   \let\sublinenumrepL\sublinenumrep%
1529   \let\linenumrepR\linenumrepR%
1530   \let\sublinenumrepR\sublinenumrepR%
1531   \letcs{ifnumberpstart}{numberpstart@R\the\l@dpscr}%
1532   \letcs{ifnumberline}{numberline@R\the\l@dpscr}%
1533   \letcs{iflinenumannotationotherside}{linenumannotationotherside@R\the\l@dpscr}%
1534   \ledRcol@true%
1535   \advance\countRline \@ne%
1536   \ifvbox\namebox{l@dRcolrawbox\the\l@dpscr}%
1537     {\vbadness=10000%
1538      \splittopskip=\z@%
1539      \do@lineRhook%
1540      \l@demptyd@ta%
1541      \global\setbox\one@lineR=\vsplit\namebox{l@dRcolrawbox\the\l@dpscr}%
1542        to\baselineskip}%
1543     \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startR}{}%
1544     \unvbox\one@lineR \global\setbox\one@lineR=\lastbox%
1545     \@writepageofparR%
1546     \getline@numR%
1547     \ifnum\@lockR>\@ne%
1548       \inserthangingsymbolRtrue%
1549     \else%
1550       \inserthangingsymbolRfalse%
1551     \fi%
1552     \setbox\l@drightbox%
1553     \hb@xt@ \Rcolwidth{%
1554       \ifl@dhidenumber%
1555         \global\l@dhidenumberfalse%
1556         \f@x@l@cksR%
1557       \else%
1558         \affixline@numR%
1559       \fi%
1560     \xifinlist{\the\l@dpscr}{\eled@sectionsR@@}%
1561     {\unless\ifshiftedpstarts%

```

```

1562     \add@apparatusR%
1563     \fi%
1564   }%
1565   {\print@lineR}%
1566 }%
1567 \add@penaltiesR%
1568 \global\advance\@donereallinesR\@ne%
1569 \global\advance\@donetotallinesR\@ne%
1570 \else%
1571   \iflinenumberRevenifblank%
1572     \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1573       \new@lineR%
1574       \l@emptyd@ta%
1575       \getline@numR%
1576       \affixline@numR%
1577       \l@dld@ta%
1578       \hspace*{\Rcolwidth}%
1579       \ledrlfill\l@drd@ta%
1580     }%
1581   \else%
1582     \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1583       \ifcontinuousnumberingwithcolumns%
1584         \new@lineR%
1585         \getline@numR%
1586         \fi%
1587         \hspace*{\Rcolwidth}%
1588       }%
1589   \fi%
1590   \global\advance\@donetotallinesR\@ne%
1591 \fi%
1592 \ledRcol@false%
1593 \let\linenumrep\linenumrepL%
1594 \let\sublinenumrep\sublinenumrepL%
1595 }
1596
1597
1598 %

```

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

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

```

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

```

```

1610 \add@insertsR%
1611 \add@Xgroupbyline%
1612 \insert@msdata%
1613 }%
1614 \affixside@noteR%
1615 }%
1616 %

```

### 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.

```

1617 \newcommand*{\getline@numR}{%
1618 \global\advance\absline@numR \@ne
1619 \do@actionsR
1620 \do@ballastR
1621 \ifledgroupnotesR\else
1622 \ifnumberline
1623 \ifsublines@R
1624 \ifnum\sub@lockR<\tw@
1625 \global\advance\subline@numR \@ne
1626 \fi
1627 \else
1628 \ifnum\@lockR<\tw@
1629 \global\advance\line@numR \@ne
1630 \global\subline@numR \z@
1631 \fi
1632 \fi
1633 \fi
1634 \fi
1635 }
1636 \newcommand*{\getline@numL}{%
1637 \global\advance\absline@num \@ne
1638 \do@actions
1639 \do@ballast
1640 \ifledgroupnotesL\else
1641 \ifnumberline
1642 \ifsublines@
1643 \ifnum\sub@lock<\tw@
1644 \global\advance\subline@num \@ne
1645 \fi
1646 \else
1647 \ifnum\@lock<\tw@
1648 \global\advance\line@num \@ne
1649 \global\subline@num \z@
1650 \fi
1651 \fi
1652 \fi
1653 \fi
1654 }
1655 %
1656 %
1657 %

```

`\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.

```

1658 \newcommand*{\do@ballastR}{\global\ballast@count=\z@
1659   \begingroup
1660     \advance\absline@numR \@ne
1661     \ifnum\next@actionlineR=\absline@numR
1662       \ifnum\next@actionR>-1001
1663         \global\advance\ballast@count by -\c@ballast
1664       \fi
1665     \fi
1666   \endgroup}
1667 %

```

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

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

```

1668
1669 \newif\ifl@dskipversenumberR
1670 \newcommand*{\do@actions@fixedcodeR}{%
1671   \ifcase\@l@dttempcnta%
1672     \or%           % 1001 = starting sublineation
1673     \global\sublines@Rtrue
1674     \or%           % 1002 = ending sublineation
1675     \global\sublines@Rfalse
1676     \or%           % 1003 = starting locking number
1677     \ifnum\@lockR=\thr@@% If there was a \endlock (implicit or explicit)
just before
1678     \global\@lockR=\tw@%
1679     \else%
1680     \global\@lockR=\@ne%
1681     \fi%
1682     \or%           % 1004 = ending locking number
1683     \ifnum\@lockR=\tw@
1684     \global\@lockR=\thr@@
1685     \else
1686     \global\@lockR=\z@
1687     \fi
1688     \or%           % 1005 = starting locking subnumber
1689     \global\sub@lockR=\@ne
1690     \or%           % 1006 = ending locking subnumber
1691     \ifnum\sub@lockR=\tw@
1692     \global\sub@lockR=\thr@@
1693     \else
1694     \global\sub@lockR=\z@
1695     \fi
1696     \or%           % 1007 = skipping numbering
1697     \l@dskipnumbertrue
1698     \or%           % 1008 = skipping numbering in stanza
1699     \l@dskipversenumberRtrue%
1700     \or%           % 1009 = hiding number
1701     \l@dhidenumbertrue%

```

```

1702 \or% % 1010 = inserting msdata
1703 \add@msdata%
1704 \else%
1705 \led@warn@BadAction
1706 \fi%
1707 }
1708
1709
1710 \newcommand*{\do@actionsR}{%
1711 \global\let\do@actions@nextR=\relax
1712 \@l@dttempcntb=\absline@numR
1713 \ifnum\@l@dttempcntb<\next@actionlineR\else
1714 \ifnum\next@actionR>-1001\relax
1715 \ifboolexpr{%
1716 bool{resumenumberingR@start}%
1717 and test {\ifdimgreater{\pagedepth}{\z@}}%
1718 }%
1719 {}%
1720 {\@firstlineofpageRtrue}%
1721 \global\page@numR=\next@actionR
1722 \ifcsdef{reset@lineR\the\absline@numR @\the\section@numR}%
1723 {%
1724 \global\line@numR=\z@ \global\subline@numR=\z@%
1725 }%
1726 {}%
1727 \global\resumenumberingR@startfalse%
1728 \add@msdata@firstlineofpage%
1729 \else
1730 \ifnum\next@actionR<-4999\relax%
1731 \@l@dttempcnta=-\next@actionR
1732 \advance\@l@dttempcnta by -5001\relax
1733 \ifsublines@R
1734 \global\subline@numR=\@l@dttempcnta
1735 \else
1736 \global\line@numR=\@l@dttempcnta
1737 \fi
1738 \else
1739 \@l@dttempcnta=-\next@actionR
1740 \advance\@l@dttempcnta by -1000\relax
1741 \do@actions@fixedcodeR
1742 \fi
1743 \fi
1744 \ifx\actionlines@listR\empty
1745 \gdef\next@actionlineR{1000000}%
1746 \else
1747 \gl@p\actionlines@listR\to\next@actionlineR
1748 \gl@p\actions@listR\to\next@actionR
1749 \global\let\do@actions@nextR=\do@actionsR
1750 \fi
1751 \fi
1752 \do@actions@nextR}
1753
1754 %

```



## VIII.4 Line number printing

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

```

\ch@cksub@l@ckR
\ch@ck@l@ckR
\fx@l@cksR
\affixline@numR
1755 \newcommand*{\l@dcalcnm}[3]{%
1756 \ifnum #1 > #2\relax
1757 \l@dtempcnta = #1\relax
1758 \advance\l@dtempcnta by -#2\relax
1759 \divide\l@dtempcnta by #3\relax
1760 \multiply\l@dtempcnta by #3\relax
1761 \advance\l@dtempcnta by #2\relax
1762 \else
1763 \l@dtempcnta=#2\relax
1764 \fi}
1765
1766 \newcommand*{\ch@cksub@l@ckR}{%
1767 \ifcase\sub@lockR
1768 \or
1769 \ifnum\sublock@disp=\@ne
1770 \l@dtempcntb \z@ \l@dtempcnta \@ne
1771 \fi
1772 \or
1773 \ifnum\sublock@disp=\tw@
1774 \else
1775 \l@dtempcntb \z@ \l@dtempcnta \@ne
1776 \fi
1777 \or
1778 \ifnum\sublock@disp=\z@
1779 \l@dtempcntb \z@ \l@dtempcnta \@ne
1780 \fi
1781 \fi}
1782
1783 \newcommand*{\ch@ck@l@ckR}{%
1784 \ifcase\@lockR
1785 \or
1786 \ifnum\lock@disp=\@ne
1787 \l@dtempcntb \z@ \l@dtempcnta \@ne
1788 \fi
1789 \or
1790 \ifnum\lock@disp=\tw@
1791 \else
1792 \l@dtempcntb \z@ \l@dtempcnta \@ne
1793 \fi
1794 \or
1795 \ifnum\lock@disp=\z@
1796 \l@dtempcntb \z@ \l@dtempcnta \@ne
1797 \fi
1798 \fi}
1799
1800 \newcommand*{\fx@l@cksR}{%
1801 \ifcase\@lockR
1802 \or
1803 \global\@lockR \tw@
1804 \or \or
1805

```

```

1806 \global\@lockR \z@
1807 \fi
1808 \ifcase\sub@lockR
1809 \or
1810 \global\sub@lockR \tw@
1811 \or \or
1812 \global\sub@lockR \z@
1813 \fi}
1814
1815
1816 \let\linenumberlistR\empty%
1817 \newcommand*{\affixline@numR}{%
1818 \ifledgroupnotesR@else\ifnumberline
1819 \ifl@dskipnumber
1820 \global\l@dskipnumberfalse
1821 \else
1822 \ifsublines@R
1823 \l@dttempcntb=\subline@numR
1824 \l@dcalcnum{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR
1825 }%
1826 \ch@cksub@l@ckR
1827 \else
1828 \l@dttempcntb=\line@numR
1829 \ifx\linenumberlistR\empty%
1830 \l@dcalcnum{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%
1831 \else
1832 \l@dttempcnta=\line@numR
1833 \edef\rem@inderR{\linenumberlistR,\number\line@numR,}%
1834 \edef\sc@n@list{\def\noexpand\sc@n@list
1835 ###1,\number\l@dttempcnta,###2|{\def\noexpand\rem@inderR{###2}}}%
1836 %
1837 \sc@n@list\expandafter\sc@n@list\rem@inderR|%
1838 \ifx\rem@inderR\empty\advance\l@dttempcnta@ne\fi%
1839 \fi
1840 \ch@ck@l@ckR
1841 \fi
1842 \ifnum\l@dttempcnta=\l@dttempcntb
1843 \ifl@dskipversenumberR\else
1844 \iftwocolumn
1845 \if@firstcolumn
1846 \gdef\l@dld@ta{\llap{\leftlinenumR}}%
1847 \iflinenumannotationotherside%
1848 \gdef\l@drd@ta{\rlap{\rightlinenumannotationR}}%
1849 \fi%
1850 \else
1851 \gdef\l@drd@ta{\rlap{\rightlinenumR}}%
1852 \iflinenumannotationotherside%
1853 \gdef\l@dld@ta{\llap{\leftlinenumannotationR}}%
1854 \fi%
1855 \else
1856 \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columnsR}{\m@ne}}}%
1857 {\l@dttempcntb=\line@margin@columnsR}%

```

```

1857     {\@l@dttempcntb=\line@marginR}%
1858     \ifnum\@l@dttempcntb>\@ne
1859     \advance\@l@dttempcntb by \numexpr\csuse{@page@forlineR@the\
section@numR @\the\absline@numR}+0\relax%
1860     \fi
1861     \ifboolexpr{%
1862     bool {l@dprintingcolumns}%
1863     and (%
1864     (test {\ifdefstring{\linenum@OnlyPages@ForColumnsR}{left}}%
1865     the\absline@numR @\the\absline@numR}+0}}%
1866     )%
1867     or%
1868     (test {\ifdefstring{\linenum@OnlyPages@ForColumnsR}{right}}%
1869     section@numR @\the\absline@numR}+0}}%
1870     )%
1871     )%
1872     }%
1873     }%
1874     {%
1875     \ifodd\@l@dttempcntb%
1876     \gdef\l@drd@ta{\rlap{{\rightlinenumR}}}%
1877     \iflinenumannotationotherside%
1878     \gdef\l@dld@ta{\llap{{\leftlinenumannotationR}}}%
1879     \fi%
1880     \else%
1881     \gdef\l@dld@ta{\llap{{\leftlinenumR}}}%
1882     \iflinenumannotationotherside%
1883     \gdef\l@drd@ta{\rlap{{\rightlinenumannotationR}}}%
1884     \fi%
1885     \fi%
1886     }%
1887     \fi
1888     \fi
1889     \f@x@l@cksR
1890 \fi
1891 \fi
1892 \fi}
1893 %
1894 %

```

## 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@numL95
\affixpstart@numR
\leftpstartnumR
\rightpstartnumR
\leftpstartnumL
\rightpstartnumL
\ifpstartnumR

```

```

1896 \newcommand*{\affixpstart@numL}{%
1897 \ifsidepstartnum
1898 \if@twocolumn
1899   \if@firstcolumn
1900     \gdef\l@dld@ta{\llap{\leftpstartnumL}}%
1901   \else
1902     \gdef\l@drd@ta{\rlap{\rightpstartnumL}}%
1903   \fi
1904 \else
1905   \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columns}{\m@ne}}}%
1906     {\@l@dttempcntb=\line@margin@columns}%
1907     {\@l@dttempcntb=\line@margin}%
1908   \ifnum\@l@dttempcntb>\@ne
1909     \advance\@l@dttempcntb \page@num
1910   \fi
1911   \ifodd\@l@dttempcntb
1912     \gdef\l@drd@ta{\rlap{\rightpstartnumL}}%
1913   \else
1914     \gdef\l@dld@ta{\llap{\leftpstartnumL}}%
1915   \fi
1916 \fi
1917 \fi
1918 }
1919 \newcommand*{\affixpstart@numR}{%
1920 \ifsidepstartnum
1921 \if@twocolumn
1922   \if@firstcolumn
1923     \gdef\l@dld@ta{\llap{\leftpstartnumR}}%
1924   \else
1925     \gdef\l@drd@ta{\rlap{\rightpstartnumR}}%
1926   \fi
1927 \else
1928   \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columnsR}{\m@ne}}}%
1929     {\@l@dttempcntb=\line@margin@columnsR}%
1930     {\@l@dttempcntb=\line@marginR}%
1931   \ifnum\@l@dttempcntb>\@ne
1932     \advance\@l@dttempcntb \page@numR
1933   \fi
1934   \ifodd\@l@dttempcntb
1935     \gdef\l@drd@ta{\rlap{\rightpstartnumR}}%
1936   \else
1937     \gdef\l@dld@ta{\llap{\leftpstartnumR}}%
1938   \fi
1939 \fi
1940 \fi
1941 }
1942
1943 \newcommand*{\leftpstartnumL}{
1944 \ifpstartnum
1945 \thepstartL
1946 \kern\linenumsep\global\pstartnumfalse\fi
1947 }

```

```

1948 \newcommand*{\rightpstartnumL}{
1949 \ifpstartnum\kern\linenumsep
1950 \thepstartL
1951 \global\pstartnumfalse\fi
1952 }
1953 \newif\ifpstartnumR
1954 \pstartnumRtrue
1955 \newcommand*{\leftpstartnumR}{
1956 \ifpstartnumR
1957 \thepstartR
1958 \kern\linenumsep\global\pstartnumRfalse\fi
1959 }
1960 \newcommand*{\rightpstartnumR}{
1961 \ifpstartnumR\kern\linenumsep
1962 \thepstartR
1963 \global\pstartnumRfalse\fi
1964 }
1965 %

```

### 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.

```

1966 \list@create{\inserts@listR}
1967 %

```

`\add@insertsR` The right text version.

```

\add@inserts@nextR
1968 \newcommand*{\add@insertsR}{%
1969 \global\let\add@inserts@nextR=\relax
1970 \ifx\inserts@listR\empty \else
1971 \ifx\next@insertR\empty
1972 \ifx\insertlines@listR\empty
1973 \global\noteschanged@true
1974 \gdef\next@insertR{100000}%
1975 \else
1976 \gl@p\insertlines@listR\to\next@insertR
1977 \fi
1978 \fi
1979 \ifnum\next@insertR=\absline@numR
1980 \gl@p\inserts@listR\to\@insertR
1981 \@insertR
1982 \global\let\@insertR=\undefined
1983 \global\let\next@insertR=\empty
1984 \global\let\add@inserts@nextR=\add@insertsR
1985 \fi
1986 \fi
1987 \add@inserts@nextR}
1988
1989 %

```

### VIII.7 Penalties

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

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

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

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

```
1990 \newcommand*{\add@penaltiesL}{}
1991 \newcommand*{\add@penaltiesR}{}
1992
1993 %
```

### 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.

```

1994 \newcommand*{\flush@notesR}{%
1995   \iftoggle{notfirststrun@jobname.\extensionchars\the\section@numR R}{%
1996     \xloop%
1997     \ifx\inserts@listR\empty \else%
1998       \gl@p\inserts@listR\to\@insertR%
1999       \@insertR%
2000       \global\let\@insertR=\undefined%
2001     \repeat%
2002   }{}%
2003 }%
2004
2005
2006 %

```

## IX Footnotes

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

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>:

```

\print@Xnotes@forpages
\restore@Xnotes@settings
\correct@Xfootins@box
\print@notesX@forpages
\restore@notesX@settings
\correct@footinsX@box

```

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

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

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

```
2007 \newcommand\print@Xnotes@forpages[1]{%
2008 %
```

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

```
2009 \ifcseempty{Xonlyside@#1}{%
2010 \csuse{#1footstart}{#1}%
2011 \csuse{#1footgroup}{#1}%
2012 }%
2013 %
```

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

```
2014 {%
2015 \ifboolexpr{%
2016 ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page
2017 }})%
2018 or%
2019 (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}})%
2020 }%
2021 %
```

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

```
2021 {%
2022 \csuse{#1footstart}{#1}%
2023 \csuse{#1footgroup}{#1}%
2024 %
```

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`.

```
2025 \global\count\csuse{#1footins}=\z@%
2026 \global\skip\csuse{#1footins}=\z@%
2027 \global\dimen\csuse{#1footins}=\maxdimen%
2028 }%
2029 %
```

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

```
2030 {%
2031 \restore@Xnotes@settings{#1}%
2032 }%
2033 %
```

End of `\print@Xnotes@forpages`.

```
2034 }%
2035 }%
2036 %
```

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

```
2037 \newcommand{\restore@Xnotes@settings}[1]{%
2038 \unless\ifnocritical@
2039 \global\count\csuse{#1footins}=\csuse{default@#1footins}%
2040 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
2041 \bgroup%
```



```

2042 \csuse{Xnotefontsize@#1}%
2043 \global\dimen\csuse{#1footins}=\csuse{Xmaxhnotes@#1}%
2044 \egroup%
2045 \global\setnamebox{#1footins@kept}=\box\namebox{#1footins}%
2046 \fi
2047 }%
2048 %

```

And now, the same for familiar footnotes.

```

2049 \newcommand\print@notesX@forpages[1]{%
2050 \ifcsemtyp{onlysideX@#1}{%
2051 \csuse{footstart#1}{#1}%
2052 \csuse{footgroup#1}{#1}%
2053 }%
2054 {%
2055 \ifboolexpr{%
2056 ((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page
2057 }})%
2058 or%
2059 (test {\ifcsstring{onlysideX@#1}{R}} and test{\ifnumodd{\c@page}}))%
2060 {%
2061 \csuse{footstart#1}{#1}%
2062 \csuse{footgroup#1}{#1}%
2063 \global\count\csuse{footins#1}=\z@%
2064 \global\skip\csuse{footins#1}=\z@%
2065 \global\dimen\csuse{footins#1}=\maxdimen%
2066 }%
2067 {%
2068 \restore@notesX@settings{#1}%
2069 }%
2070 }%
2071 }%
2072 \newcommand{\restore@notesX@settings}[1]{%
2073 \unless\ifnofamiliar@
2074 \global\count\csuse{footins#1}=\csuse{default@footins#1}%
2075 \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
2076 \bgroup%
2077 \csuse{Xnotefontsize@#1}%
2078 \global\dimen\csuse{footins#1}=\csuse{maxhnotesX@#1}%
2079 \egroup%
2080 \global\setnamebox{footins#1@kept}=\box\namebox{footins#1}%
2081 \fi
2082 }%
2083 %

```

`\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.

```

2084 \newcommand{\insert@notes@for@onlyside}{%
2085 \def\do##1{%
2086 \unless\ifnocritical@
2087 \ifvoid\csuse{##1footins@kept}\else%
2088 \expandafter\insert\csname ##1footins\endcsname%

```

```

2089     \bgroup%
2090     \unvnamebox{##1footins@kept}%
2091     \egroup%
2092     \fi%
2093     \restore@Xnotes@settings{##1}%
2094     \fi%
2095     \unless\ifnofamiliar@%
2096     \ifvoid\csuse{footins##1@kept}\else%
2097     \expandafter\insert\csname footins##1\endcsname%
2098     \bgroup%
2099     \unvnamebox{footins##1@kept}%
2100     \egroup%
2101     \fi%
2102     \restore@notesX@settings{##1}%
2103     \fi%
2104     }%
2105     \dolistloop{\@series}%
2106   }%
2107   %

```

## 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 `\edtext@later` and the other for `\edtextnow`.

```

\edtextnow
2108 \newcount\edtext@now%
2109 \newcount\edtext@later%
2110 %

2111 \newcommand{\edtextlater}[2]{%#1 lemma, #2 critical notes
2112 %

```

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

```

2113 \global\advance\edtext@later by \@ne%
2114 %

2115 % 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}.
2116 \advance\@edtext@level by \@ne%
2117 \ifcvoid{sw@list@edtext@the\@edtext@level}%
2118     {\global\let\sw@inthisedtext\empty}%
2119     {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
2120 %

```

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

```

2121 %
2122 \csxdef{edtext@later@the\edtext@later}{%
2123 %

```

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

```

2124 \noexpand\edtext{%
2125 }%
2126 %

```

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

```

2127 {%
2128 \unexpanded{%
2129 \lemma{\no@expands #1}%
2130 }%
2131 \noexpand\xxref{start:edtext:later:\the\edtext@later}{end:edtext:
later:\the\edtext@later}%
2132 \noexpand\linenum{|||||\edfont@info}%
2133 \unexpanded{\def\sw@inthisedtext}{\expandonce{\sw@inthisedtext}}%
2134 %

```

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.

```

2135 \ifledRcol%
2136 \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@true}}%
2137 {%
2138 \noexpand\setcounter{pstartR}{\noexpand\xpstartref{start:edtext:
later:\the\edtext@later}}%
2139 }%
2140 \else%
2141 \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@false}}%
2142 \unexpanded{\appto\@beforeinsertofthisedtext}%
2143 {%
2144 \noexpand\setcounter{pstartL}{\noexpand\xpstartref{start:edtext:
later:\the\edtext@later}}%
2145 }%
2146 \fi%
2147 %

```

And the footnote command of this `\edtextlater`.

```

2148 \unexpanded{#2}%
2149 }%
2150 }%
2151 %

```

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`

```

2152 \edlabel{start:edtext:later:\the\edtext@later}%
2153 \flag@start@later%
2154 \bgroup%
2155 \def\lemma##1{%
2156 \ifledRcol%
2157 \write\linenum@outR{\string\@lemma}%
2158 \else%
2159 \write\linenum@out{\string\@lemma}%
2160 \fi%

```

```

2161 }%
2162 \renewcommand{\do}[1]{\expandafter\renewcommandx\csname ##1footnote\
endcsname[2][1,usedefault]{}\unskip}%\unskip because of a spurious space in
\newcommandx
2163 \dolistloop{\@series}%
2164 #2%
2165 \egroup%
2166 \showlemma{#1}%
2167 \edlabel{end:edtext:later:\the\edtext@later}%
2168 \flag@end@later%
2169 %

```

We decrease the counter increased at the beginning.

```

2170 \advance\@edtext@level by -\@ne%
2171 }%
2172 %

```

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

```

2173 \newcommand{\edtextnow}[0]{%
2174 \global\advance\edtext@now by \@ne
2175 \csuse{edtext@later@\the\edtext@now}%
2176 }%
2177 %

```

## 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.

```

2178 \list@create{\labelref@listR}
2179
2180 %

```

**\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.

```

2181 \def\l@dmake@labelsR#1|#2|#3|#4|#5|#6|#7{%
2182 \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
\relax%
2183 \else%
2184 \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
2185 \fi%
2186 \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\@Rlineflag}%
2187 \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
2188 \global\toggletrue{label@#7@ledRcol}%
2189 \ignorespaces}
2190
2191 \AtBeginDocument{%
2192 \def\l@dmake@labelsR#1|#2|#3|#4|#5|#6|#7{%
2193 }

```

```
2194
2195 %
```

`\@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.

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

```
\sidenotemargin*
\sidenotemarginR
2196 \WithSuffix\newcommand\sidenotemargin*[1]{%
2197   \l@dgetsidenote@margin{#1}
2198   \global\sidenote@marginR=\@l@dtmpcntb
2199   \global\sidenote@margin=\@l@dtmpcntb
2200 }
2201 \newcommand{\sidenotemarginR}[1]{%
2202   \l@dgetsidenote@margin{#1}%
2203   \global\sidenote@marginR=\@l@dtmpcntb%
2204 }%
2205 \newcount\sidenote@marginR
2206 \global\sidenote@marginR=\@ne%
2207
2208 \ifmovecolumnsonrightpage%
2209   \sidenotemargin{inner}%
2210   \sidenotemarginR{outer}%
2211 \fi%
2212
2213 %
```

`\space@rightnote@leftcolumn` 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.

```
2214 \newdimen\@morespace@leftnote@rightcolumn%
2215 \newdimen\@morespace@rightnote@leftcolumn%
2216
2217
2218
2219 \newcommand{\get@sidenote@morespace@columns}{%
2220 %
```

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

```

2221 \if@sidenotesmarginpage%
2222 %
    Now, we can calculate \@morespace@leftnote@rightcolumn and \@morespace@rightnote@leftcol
2223 \global\@morespace@leftnote@rightcolumn=\dimexpr\intercolumns@width + \
Lcolwidth\relax%
2224 \global\@morespace@rightnote@leftcolumn=\dimexpr\intercolumns@width + \
Rcolwidth\relax%
2225 %
    If \if@sidenotesmarginpage is False
2226 \else%
2227 \global\@morespace@leftnote@rightcolumn=\z@%
2228 \global\@morespace@rightnote@leftcolumn=\z@%
2229 %
    End of \get@sidenote@morespace@columns.
2230 \fi%
2231 }%
2232 %

```

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

```

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

```

```

2261 \ifodd\@l@dttempcntb%
2262 \setl@drp@rbox{\sidenotecontent@}%
2263 \gdef\sidenotecontent@{}%
2264 \numdef{\itemcount@}{0}%
2265 \dolistloop{\l@dcsnotetext@l}%
2266 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
2267 \setl@dlp@rbox{\sidenotecontent@}%
2268 \else%
2269 \setl@dlp@rbox{\sidenotecontent@}%
2270 \gdef\sidenotecontent@{}%
2271 \numdef{\itemcount@}{0}%
2272 \dolistloop{\l@dcsnotetext@r}%
2273 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
2274 \setl@drp@rbox{\sidenotecontent@}%
2275 \fi%
2276 \fi%
2277 \fi%
2278 \advance\@edindex@for@note@{m@ne}%
2279 }
2280
2281 %

```

## 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.

```

\inserthangingsymbolL82 \newif\ifinserthangingsymbolR
\inserthangingsymbolR83 \newcommand{\inserthangingsymbolL}{%
2284 \ifinserthangingsymbol%
2285 \ifinstanzaL%
2286 \@hangingsymbol%
2287 \fi%
2288 \ifinstanzaL%
2289 \@hangingsymbol%
2290 \fi%
2291 \fi%
2292 }%
2293 \newcommand{\inserthangingsymbolR}{%
2294 \ifinserthangingsymbolR%
2295 \ifinstanzaR%
2296 \@hangingsymbol%
2297 \fi%
2298 \ifinstanzaR%
2299 \@hangingsymbol%
2300 \fi%
2301 \fi%
2302 }%
2303 %

```

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.

```

2304 \chardef\next=\catcode`&
2305 \catcode`&=\active
2306
2307 %
    astanza This is roughly an environmental form of \stanza, which treats its
    stanza-like contents as a single chunk.

2308 \newenvironment{astanza}[2][1,2,usedefault]{%
2309 \ifledRcol%
2310 \global\inastanzaRtrue%
2311 \else%
2312 \global\inastanzaLtrue%
2313 \fi%
2314 \catcode`&=\active
2315 \global\stanza@count\@ne\stanza@modulo\@ne
2316 \@advancestanzanumber%
2317 \newdimen\parindent@beforestanza%
2318 \parindent@beforestanza=\parindent%Keep in memory the standard \parindent
2319 \ifnum\usernamecount{sza@00}=\z@
2320 \let\stanza@hang\relax
2321 \let\endlock\relax
2322 \else
2323 \rightskip\z@ plus 1fil\relax
2324 \fi
2325 \ifnum\usernamecount{szp@00}=\z@
2326 \let\sza@penalty\relax
2327 \fi
2328 \def&{%
2329 \endlock\mbox{}%
2330 \sza@penalty
2331 \global\advance\stanza@count\@ne
2332 \@astanza@line}%
2333 \def&{\@stopastanza}%
2334 \ifboolexpr{%
2335 not test{\ifdefvoid{\at@every@stanza}}%
2336 and test{\ifstrempy{#1}}%
2337 and test{\ifstrempy{#2}}%
2338 {\pstart [] [\at@every@stanza] \at@start@every@stanza}%
2339 {\pstart [#1] [#2] \at@start@every@stanza}%
2340 \@astanza@line
2341 \@insertstanzanumber%
2342 \let\par\relax\ignorespaces%No paragraph in verses
2343 }{%
2344 \global\undef\parindent@beforestanza%
2345 }%
2346
2347 %

```

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

```

2348 \newcommand{\@stopastanza}[2][1,2,usedefault]{%

```



```

2349 \endlock\mbox{}}%
2350 \ifboolexpr{%
2351   not test{\ifdefvoid{\at@every@stop@stanza}}%
2352   and test{\ifstrempy{#1}}%
2353   and test{\ifstrempy{#2}}%
2354 }%
2355   {\before@every@stop@stanza\pend[] [\at@every@stop@stanza]}%
2356   {\before@every@stop@stanza\pend[#1] [#2]}%
2357 }%
2358 %

```

`\@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.

```

2359 \newcommand*{\@astanza@line}{%
2360   \stanza@indent%
2361   \endgraf
2362   \stanza@hang%
2363   \ignorespaces}
2364
2365 %

```

Lastly reset the modified category codes.

```

2366 \catcode`\&=\next
2367
2368 %

```

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

```

\thestanzaR
2369 \newcounter{stanzaL}
2370 \newcounter{stanzaR}
2371 \renewcommand{\thestanzaL}{%
2372   \textbf{\arabic{stanzaL}}%
2373 }
2374 \renewcommand{\thestanzaR}{%
2375   \textbf{\arabic{stanzaR}}%
2376 }
2377 %
2378 %

```

### 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.

`\l@dcchecklang`

`\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 `\`.

```

2379 \patchcmd{\bbl@set@language}%
2380   {\select@language{\language}}%
2381   {\edef\language{#1}\select@language{\language}}%
2382   {}%
2383   {}%
2384
2385 %

```

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`.

```

2386 \newcommand*\l@duselanguage}[1]{%
2387   \def\theledlanguageL{}%
2388   \def\theledlanguageR{}%
2389   \def\reledpar@init@language{}%Will be defined later, depending of language
2390   %

```

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

```

2391 \AtBeginDocument{%
2392   \ifpackageloaded{babel}{%
2393     \@ifundefined{bbl@main@language}{%
2394 %

```

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

```

2395   }{%
2396 %

```

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.

```

2397 \let\l@doldselectlanguage\selectlanguage
2398 \let\l@doldbbl@set@language\bbl@set@language
2399 \renewcommand{\selectlanguage}[1]{%
2400   \l@doldselectlanguage{#1}%
2401   \ifledRcol \gdef\theledlanguageR{#1}%
2402   \else      \gdef\theledlanguageL{#1}%
2403   \fi}
2404 %

```

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

```

2405 \renewcommand*\l@duselanguage}[1]{%
2406 \let\bbl@savelastskip\relax% Babel 3.61, which provokes spurious
vertical space
2407 \let\bbl@restorelastskip\relax%% Babel 3.61, which provokes spurious
vertical space
2408 \expandafter\l@doldselectlanguage\expandafter{#1}%
2409 }%
2410 %

```

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

```

2411 \renewcommand{\reledpar@init@language}{%
2412 \gdef\theledlanguageL{\bbl@main@language}%
2413 \gdef\theledlanguageR{\bbl@main@language}%
2414 }
2415 }%
2416 }{}%
2417 %

```

If use polyglossia

```

2418 \@ifpackageloaded{polyglossia}{%
2419 \@ifpackagelater{polyglossia}{2025/01/31}{\led@err@polyglossiaTooOld}
%
2420 \renewcommand*\l@duselanguage}[1]{%
2421 \csuse{no\language@name @numbers}%
2422 \selectlanguage{#1}%
2423 }%
2424 \AddToHook{env/Leftside/begin}{%
2425 \AddToHookNext{cmd/polyglossia@language@switched/after}{%
2426 \ifl@dpairing%Ensure we are still in leftside
2427 \unless\ifledRcol%
2428 \xdef\theledlanguageL{\language@name}%
2429 \fi%
2430 \fi%
2431 }%
2432 }%
2433 \AddToHook{env/Rightside/begin}{%
2434 \AddToHookNext{cmd/polyglossia@language@switched/after}{%
2435 \ifl@dpairing%Ensure we are still in rightside
2436 \ifledRcol%
2437 \xdef\theledlanguageR{\language@name}%
2438 \fi%
2439 \fi%
2440 }%
2441 }%
2442 \renewcommand{\reledpar@init@language}{%
2443 \xdef\theledlanguageL{\main@language@name}%
2444 \xdef\theledlanguageR{\main@language@name}%
2445 }
2446 %

```

That is it.

```

2447 }{}%
2448 }%

```

2449 %

## 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.  
`\l@dc@maxchunks`

```
2450 \newcount\l@dc@maxchunks
2451 \newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}
2452   \maxchunks{5120}
2453
2454 %
```

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

```
2455 \newcount\l@dnumpstartsR
2456
2457 %
```

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

```
2458 \newcount\l@dpscL
2459 \newcount\l@dpscR
2460
2461 %
```

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

```
2462 \newcommand*{\l@dsetuprawboxes}{%
2463   \@l@dttempcntb=\l@dc@maxchunks
2464   \loop\ifnum\@l@dttempcntb>\z@
2465     \newnamebox{\l@dLcolrawbox\the\@l@dttempcntb}
2466     \newnamebox{\l@dRcolrawbox\the\@l@dttempcntb}
2467     \advance\@l@dttempcntb \m@ne
2468     \repeat}
2469
2470 %
```

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

```
2471 \newcommand*{\l@dsetupmaxlinecounts}{%
2472   \@l@dttempcntb=\l@dc@maxchunks
2473   \loop\ifnum\@l@dttempcntb>\z@
2474     \newnamecount{\l@dmaxlinesinpar\the\@l@dttempcntb}
```

```

2475     \advance\@l@tempcntb \m@ne
2476     \repeat}
2477 \newcommand*{\l@dzzeromaxlinecounts}{%
2478     \begingroup
2479     \@l@tempcntb=\l@dc@maxchunks
2480     \loop\ifnum\@l@tempcntb>\z@
2481         \global\usenamecount{l@dmmaxlinesinpar\the\@l@tempcntb}=\z@
2482         \advance\@l@tempcntb \m@ne
2483     \repeat
2484     \endgroup}
2485
2486 %

```

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

```

2487 \AtBeginDocument{%
2488     \l@dsetupprawboxes
2489     \l@dsetupmaxlinecounts
2490     \l@dzeromaxlinecounts
2491     \l@dnumpststartsL=\z@
2492     \l@dnumpststartsR=\z@
2493     \l@dpscL=\z@
2494     \l@dpscR=\z@}
2495
2496 %

```

## XV Checking text to be processed

```

\if@pstarts \check@pstarts returns \@pstartstrue if there are any unprocessed chunks.
\@pstartstrue
\@pstartsfalse
\check@pstarts
\newif\if@pstarts
\newcommand*{\check@pstarts}{%
    \@pstartsfalse
    \ifnum\l@dnumpststartsL>\l@dpscL
        \@pstartstrue
    \else
        \ifnum\l@dnumpststartsR>\l@dpscR
            \@pstartstrue
        \fi
    \fi
}
%

```

```

\ifaraw@text \checkraw@text checks whether the current Left or Right box is void or not. If
\araw@texttrue one or other is not void it sets \araw@texttrue, otherwise both are void and it sets
\araw@textfalse \araw@textfalse.
\checkraw@text
\newif\ifaraw@text
\newcommand*{\checkraw@text}{%
    \araw@textfalse
    \ifvbox\namebox{l@dLcolrawbox\the\l@dpscL}

```

```

2514 \araw@texttrue
2515 \else
2516 \ifvbox\namebox{!@dRcolrawbox\the!@dpScR}
2517 \araw@texttrue
2518 \fi
2519 \fi
2520 }
2521 %
2522 %

```

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

```

2523 \newcommand*{\@writelinesinparL}{%
2524 \edef\next{%
2525 \write\linenum@out{\string\@pend[\the\@donereallinesL]}}%
2526 \next
2527 \global\@donereallinesL \z@}
2528 \newcommand*{\@writelinesinparR}{%
2529 \edef\next{%
2530 \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}%
2531 \next
2532 \global\@donereallinesR \z@}
2533 %
2534 %

```

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

```

2535 \@writepageofparL
2536 \newcommand*{\@writepageofparL}[0]{%
2537 \ifnum\@donereallinesL=\z@%
2538 \edef\next{%
2539 \write\linenum@out{\string\@pstart{\the!@dpScL}{\the!@c@page}{\the!@
2540 numpagelinesL}}%
2541 }%
2542 \next%
2543 \fi%
2544 }%
2545 \newcommand*{\@writepageofparR}[0]{%
2546 \ifnum\@donereallinesR=\z@%
2547 \edef\next{%
2548 \write\linenum@outR{\string\@pstartR{\the!@dpScR}{\the!@c@page}{\the!@
2549 numpagelinesR}}%
2550 }%
2551 \next%
2552 \fi%
2553 }%
2554 %

```

## XVI Parallel columns

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

```

2552 \newsavebox{\@eledsectionL}%
2553 \newsavebox{\@eledsectionR}%
2554 %

```

**\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.

```

2555 \newcommand*\Columns{%
2556   \ifl@dpairing%
2557     \led@err@Columns@InsideEnv%
2558   \fi%
2559   \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{
l@dLcolrawbox1}}}{%
2560     \ifcsboxvoid{l@dRcolrawbox1}{%
2561       \ifcsboxvoid{l@dLcolrawbox1}%
2562       {\led@err@Columns@WithoutEnv}%
2563       {\led@err@Columns@WithoutRightside}%
2564     }%
2565     {\led@err@Columns@WithoutLeftside}%
2566   }{%
2567     \global\l@dprintingcolumnstrue%
2568     \eledsection@correcting@skip=-\baselineskip% Correction for sections'
titles
2569     \ifnum\l@dnumstartsL=\l@dnumstartsR\else
2570       \led@err@BadLeftRightPstarts{\the\l@dnumstartsL}{\the\l@dnumstartsR}%
2571     \fi
2572 %

```

Start a group and zero counters, etc.

```

2573 \begingroup
2574   \l@dzeropenalties
2575   \endgraf\global\num@lines=\prevgraf
2576     \global\num@linesR=\prevgraf
2577   \global\par@line=\z@
2578   \global\par@lineR=\z@
2579   \global\l@dpscL=\z@
2580   \global\l@dpscR=\z@
2581   \get@familiarfootnote@number%
2582   \get@intercolumns@width%
2583   \get@sidenote@morespace@columns%
2584 %

```

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

```

2585 \check@pstarts
2586 \loop\if@pstarts
2587   \global\pstartnumtrue
2588   \global\pstartnumRtrue
2589 %

```

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.

```

2590 \global\advance\l@dpscl \@ne
2591 \global\advance\l@dpscR \@ne
2592 \global\advance\pstarts@typeset@L\@ne%
2593 \restore@pstartL@pc%
2594 \restore@pstartR@pc%
2595 %

```

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

```

2596 \Columns@print@before@pstart%
2597 %

```

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

```

2598 \checkraw@text
2599 { \loop\ifaraw@text
2600 %

```

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

```

2601 \l@duselanguage{\theledlanguageL}%
2602 \do@lineL
2603 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@L}%
2604 {%
2605 \ifdefstring{\@eledsectmark}{L}%
2606 {\csuse{eled@sectmark@the\pstarts@typeset@L}%
2607 }{%
2608 \global\csundef{eled@sectmark@the\pstarts@typeset@L}%
2609 \savebox{\@eledsectionL}{\parbox[t][t]{\Lcolwidth}{\vbox
{}\print@eledsectionL}}\vbox{-> prevent alignment troubles with RTL
language
2610 }%
2611 {}%
2612 \l@duselanguage{\theledlanguageR}%
2613 \do@lineR
2614 \xifinlist{\the\l@dpscR}{\eled@sectionsR@L}%
2615 {%
2616 \ifdefstring{\@eledsectmark}{R}%
2617 {\csuse{eled@sectmark@the\l@dpscR R}%
2618 }{%
2619 \global\csundef{eled@sectmark@the\l@dpscR R}%
2620 \savebox{\@eledsectionR}{\parbox[t][t]{\Rcolwidth}{\vbox
{}\print@eledsectionR}}\vbox{-> prevent alignment troubles with RTL
language
2621 }%
2622 \hb@xt@ \hsize{%
2623 \ifdefstring{\columns@position}{L}{\hfill }%
2624 \print@leftcolumn%
2625 \print@columnseparator%
2626 \print@rightcolumn%
2627 \ifdefstring{\columns@position}{R}{\hfill}%
2628 }%
2629 \checkraw@text
2630 \checkverseL
2631 \checkverseR

```



```

2632     \checkpb@columns
2633     \repeat}
2634 %

```

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.

```

2635     \@writelinesinparL
2636     \@writelinesinparR
2637     \check@pstarts
2638     \ifbypstart@%
2639         \unless\ifinstanza%
2640             \write\linenum@out{\string\@set[1]}%
2641             \resetprevline@%
2642         \fi%
2643     \fi
2644     \ifbypstart@R
2645         \unless\ifinstanza%
2646             \write\linenum@outR{\string\@set[1]}%
2647             \resetprevline@%
2648         \fi%
2649     \fi
2650     \Columns@print@after@pend%
2651 \repeat
2652 %

```

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.

```

2653     \flush@notes
2654     \flush@notesR
2655 \endgroup
2656 %

2657 \global\l@dpscl=\z@
2658 \global\l@dpscr=\z@
2659 \global\l@dnpstartsL=\z@
2660 \global\l@dnpstartsR=\z@
2661 \global\l@dprintingcolumnsfalse%
2662 \ignorespaces
2663 \global\instanzaLfalse%
2664 \global\instanzaRfalse%
2665 }%
2666 }%
2667
2668 %

```

`\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).

```

2669 \def\print@columnseparator{
2670     \ifdim\beforecolumnseparator<0pt%
2671         \hfill%
2672     \else%

```

```

2673   \hspace{\beforecolumnseparator}%
2674   \fi%
2675   \columnseparator%
2676   \ifdim\aftercolumnseparator<0pt%
2677     \hfill%
2678   \else%
2679     \hspace{\aftercolumnseparator}%
2680   \fi%
2681 }%
2682 %

```

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

`\intercolumns@width`

- 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.

```

2683 \newdimen\intercolumns@width%
2684 \newcommand{\get@intercolumns@width}{%
2685 %   \begin{macrocode}
2686   \global\intercolumns@width=\z@%
2687   \ifdefstring{\columns@position}{C}%
2688 %

```

First case, the columns are centered.

```

2689   {%
2690 %

```

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

```

2691   \ifboolexpr{%
2692     test {\ifdimless{\beforecolumnseparator}{\z@}}%
2693     and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2694   }%
2695   {%
2696     \global\intercolumns@width=\dimexpr%
2697       \columnrulewidth +%
2698       (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth)*1/2%
The total of the width before and after column separator
2699     \relax%
2700   }%
2701   {%
2702 %

```

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

```

2703   \ifboolexpr{%
2704     test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2705     and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%

```

```

2706 }%
2707 {%
2708   \global\intercolumns@width=\dimexpr%
2709     \columnrulewidth + \beforecolumnseparator + \aftercolumnseparator
2710 %
2711   \relax%
2712 }%
2713 {%

```

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

```

2714 \ifboolexpr{%
2715   test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2716   and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2717 }%
2718 {%
2719   \global\intercolumns@width=\dimexpr%
2720     \columnrulewidth + \beforecolumnseparator +%
2721     (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
beforecolumnseparator) * 1/3%
2722   \relax%
2723 }%
2724 {%
2725 %

```

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

```

2726 \ifboolexpr{%
2727   test {\ifdimless{\beforecolumnseparator}{\z@}}%
2728   and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2729 }%
2730 {%
2731   \global\intercolumns@width=\dimexpr%
2732     \columnrulewidth + \aftercolumnseparator +%
2733     (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
aftercolumnseparator) * 1/3%
2734   \relax%
2735 }%
2736 {%
2737 %

```

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

```

2738 }%
2739 %

```

Other case, the columns are left or right aligned.

```

2740 {%
2741 %

```

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

```

2742 \ifboolexpr{%
2743   test {\ifdimless{\beforecolumnseparator}{\z@}}%

```

```

2744 and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2745 }%
2746 {%
2747   \global\intercolumns@width=\dimexpr%
2748     \columnrulewidth +%
2749     (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth)*2/3%
The total of the width before and after column separator
2750   \relax%
2751 }%
2752 {}%
2753 %

```

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

```

2754 \ifboolexpr{%
2755   test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2756   and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2757 }%
2758 {%
2759   \global\intercolumns@width=\dimexpr%
2760     \columnrulewidth + \beforecolumnseparator + \aftercolumnseparator
%
2761   \relax%
2762 }%
2763 {}%
2764 %

```

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

```

2765 \ifboolexpr{%
2766   test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2767   and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2768 }%
2769 {%
2770   \global\intercolumns@width=\dimexpr%
2771     \columnrulewidth + \beforecolumnseparator +%
2772     (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
beforecolumnseparator) * 1/2%
2773   \relax%
2774 }%
2775 {}%
2776 %

```

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

```

2777 \ifboolexpr{%
2778   test {\ifdimless{\beforecolumnseparator}{\z@}}%
2779   and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2780 }%
2781 {%
2782   \global\intercolumns@width=\dimexpr%
2783     \columnrulewidth + \aftercolumnseparator +%
2784     (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
aftercolumnseparator) * 1/2%
2785   \relax%

```

```

2786     }%
2787     {}%
2788 %

```

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

```

2789 }%
2790 %

```

End of `\get@intercolumns@width`.

```

2791 }%
2792 %

```

`\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.

```

2793 \newcommand{\print@leftcolumn}{%
2794   \if\page@num>\page@numR%
2795     \@l@tempcntb=\page@num%
2796   \else%
2797     \@l@tempcntb=\page@numR%
2798   \fi%
2799   \ifboolexpr{%
2800     not bool {movecolumnspositiononrightpage}%
2801     or test {\ifnumodd{\@l@tempcntb}}%
2802   }%
2803   {%
2804     \unhbox\l@dleftbox%
2805     \ifhbox\@eledsectionL%
2806       \usebox{\@eledsectionL}%
2807     \fi%
2808   }%
2809   {%
2810     \unhbox\l@drightbox%
2811     \ifhbox\@eledsectionR%
2812       \usebox{\@eledsectionR}%
2813     \fi%
2814   }%
2815 }%
2816
2817
2818 \newcommand{\print@rightcolumn}{%
2819   \if\page@num>\page@numR%
2820     \@l@tempcntb=\page@num%
2821   \else%
2822     \@l@tempcntb=\page@numR%
2823   \fi%
2824   \ifboolexpr{%
2825     not bool {movecolumnspositiononrightpage}%
2826     or test {\ifnumodd{\@l@tempcntb}}%
2827   }%
2828   {%
2829     \unhbox\l@drightbox%

```

```

2830     \ifhbox\@eledsectionR%
2831       \usebox{\@eledsectionR}%
2832     \fi%
2833   }%
2834   {%
2835     \unhbox\l@dleftbox%
2836     \ifhbox\@eledsectionL%
2837       \usebox{\@eledsectionL}%
2838     \fi%
2839   }%
2840 }%
2841 %

```

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

```

2842
2843 \newcommand{\checkpb@columns}{%
2844   \newif\if@pb
2845   \newif\if@nopb
2846   \IfStrEq{\led@pb@setting}{before}{
2847     \numdef{\next@absline}{\the\absline@num+1}%
2848     \numdef{\next@abslineR}{\the\absline@numR+1}%
2849     \xifinlist{\next@absline}{\l@prev@pb}{\@pbtrue}{}%
2850     \xifinlist{\next@abslineR}{\l@prev@pbR}{\@pbtrue}{%
2851     \xifinlist{\next@absline}{\l@prev@nopb}{\@nopbtrue}{}%
2852     \xifinlist{\next@abslineR}{\l@prev@nopbR}{\@nopbtrue}{%
2853   }}
2854   \IfStrEq{\led@pb@setting}{after}{
2855     \xifinlist{\the\absline@num}{\l@prev@pb}{\@pbtrue}{}%
2856     \xifinlist{\the\absline@numR}{\l@prev@pbR}{\@pbtrue}{%
2857     \xifinlist{\the\absline@num}{\l@prev@nopb}{\@nopbtrue}{}%
2858     \xifinlist{\the\absline@numR}{\l@prev@nopbR}{\@nopbtrue}{%
2859   }}
2860   \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
2861   \if@pb\pagebreak[4]\fi
2862 }
2863 %

```

`\columnseparator` `\columnrulewidth` 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).

```

2864 \newcommand*{\columnseparator}{%
2865   \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}
2866 \newdimen\columnrulewidth
2867   \columnrulewidth=\z@
2868
2869 %

```

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

```

2870 \newcommand*{\columnsposition}[1]{%

```

```

2871 \xdef\columns@position{#1}%
2872 }%
2873 \xdef\columns@position{R}%
2874 %

```

`\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`.

```

2875 \newlength{\beforecolumnseparator}%
2876 \setlength{\beforecolumnseparator}{-2pt}%
2877
2878 \newlength{\aftercolumnseparator}%
2879 \setlength{\aftercolumnseparator}{-2pt}%
2880
2881 %

```

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.

```

2882 \newcommand{\setwidthliketwocolumns}{%
2883   \get@intercolumns@width%As \columns is not necessary called before \
setwidthliketwocolumns
2884   \hsize=\dimexpr \Lcolwidth + \intercolumns@width + \Rcolwidth \relax%
2885 }%
2886
2887 \newcommand{\setpositionliketwocolumns@L}{%
2888   \renewcommand{\ledrlfill}{\hfill}%
2889 }%
2890
2891 \newcommand{\setnotespositionliketwocolumns@L}{%
2892 }%
2893
2894 \newcommand{\setpositionliketwocolumns@C}{%
2895   \doinsidelinehook{\hfill}%
2896   \renewcommand{\ledrlfill}{\hfill}%
2897 }%
2898
2899 \newcommand{\setnotespositionliketwocolumns@C}{%
2900   \newdimen\temp%
2901   \newdimen\tempa%
2902   \temp=\hsize%
2903   \tempa=\Lcolwidth%
2904   \advance\tempa\Rcolwidth%
2905   \advance\temp-\tempa%
2906   \divide\temp by 2%
2907   \leftskip=\temp%
2908   \rightskip=-\temp%
2909 }%

```

```

2910
2911 \newcommand{\setpositionliketwocolumns@R}{%
2912   \doinsidelinehook{\hfill}%
2913 }%
2914 %

```

`\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.

```

2915 \newcommand{\Columns@print@before@pstart}{%
2916   \ifboolexpr{%
2917     test{\ifcsstring{before@pstartL@the\l@dpscl}{\at@every@pstart}}%
2918     and test {\ifcsstring{before@pstartR@the\l@dpscr}{\at@every@pstart}}%
2919     and test {\ifdefempty{\at@every@pstart}}}%
2920     {}%
2921     {%
2922     \ifboolexpr{not togl{before@pstartR@the\l@dpscr @par} and not togl{
before@pstartL@the\l@dpscl @par}}{%
2923       \csuse{before@pstartL@the\l@dpscl}%
2924       \csuse{before@pstartR@the\l@dpscr}%
2925     }{%
2926       \hb@xt@ \hsize{%
2927         \ifdefstring{columns@position}{L}{\hfill }%
2928         \par\parbox[t] [] [t]{\Lcolwidth}{%
2929           \csuse{before@pstartL@the\l@dpscl}%
2930         }%
2931         \print@columnseparator%
2932         \parbox[t] [] [t]{\Rcolwidth}{%
2933           \set@sectcountR%
2934           \csuse{before@pstartR@the\l@dpscr}%
2935         }%
2936         \ifdefstring{columns@position}{R}{\hfill}%
2937       }%
2938     }%
2939   }%
2940   \global\csundef{before@pstartL@the\l@dpscl}%
2941   \global\csundef{before@pstartR@the\l@dpscr}%
2942 }%
2943 \newcommand{\Columns@print@after@pend}{%
2944   \ifboolexpr{%
2945     test{\ifcsstring{after@pendL@the\l@dpscl}{\at@every@pend}}%
2946     and test {\ifcsstring{after@pendR@the\l@dpscr}{\at@every@pend}}%
2947     and test {\ifdefempty{\at@every@pend}}}%
2948     {}%
2949     {%
2950     \ifboolexpr{not togl{after@pendR@the\l@dpscr @par} and not togl{
after@pendL@the\l@dpscl @par}}{%
2951       \csuse{after@pendL@the\l@dpscl}%
2952       \csuse{after@pendR@the\l@dpscr}%
2953     }{%
2954       \hb@xt@ \hsize{%
2955         \ifdefstring{columns@position}{L}{\hfill }%
2956         \parbox[t] [] [t]{\Lcolwidth}{%

```



```

2957     \csuse{after@pendL@the\l@dpscL}%
2958   }%
2959   \print@columnseparator%
2960   \parbox[t] [] [t]{\Rcolwidth}{%
2961     \set@sectcountR%
2962     \csuse{after@pendR@the\l@dpscR}%
2963   }%
2964   \ifdefstring{\columns@position}{R}{-}{\hfill}%
2965 }%
2966 }%
2967 }%
2968 \global\csundef{after@pendL@the\l@dpscL}%
2969 \global\csundef{after@pendR@the\l@dpscR}%
2970 }%
2971 %

```

## 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`

```

2972 \newcount\numpagelinesL
2973 \newcount\numpagelinesR
2974 \newcount\l@dminpagelines
2975
2976 %

```

### 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.

```

2977 \newcommandx*{\Pages}[1][1,usedefault]{%
2978   \ifl@dpairing%
2979     \led@err@Pages@InsideEnv%
2980   \fi%
2981   \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{
l@dLcolrawbox1}}}{%
2982     \ifcsboxvoid{l@dRcolrawbox1}{%
2983       \ifcsboxvoid{l@dLcolrawbox1}%
2984       {\led@err@Pages@WithoutEnv}%
2985       {\led@err@Pages@WithoutRightside}%
2986     }%
2987     {\led@err@Pages@WithoutLeftside}%
2988   }{%
2989   \ifstrequal{#1}{mainmatter}{\Pages@mainmattertrue}{\Pages@mainmatterfalse
}%

```

```

2990 \eledsection@correcting@skip=-2\baselineskip% line correcting for section
titles.
2991 \parledgroup@notespacing@set@correction%
2992 \typeout{}%
2993 \typeout{***** PAGES *****}%
2994 \ifnum\l@dnumstartsL=\l@dnumstartsR\else%
2995 \led@err@BadLeftRightPstarts{\the\l@dnumstartsL}{\the\l@dnumstartsR}%
2996 \fi%
2997 %

```

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

```

2998 \clearto1@devenpage%
2999 \global\l@dprintingpagestrue%
3000 \begingroup%
3001 %

```

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

```

3002 \Lcolwidth=\Lcolwidth@pages%
3003 \Rcolwidth=\Rcolwidth@pages%
3004 %

```

```

3005 \l@dzeropenalties%
3006 \endgraf\global\num@lines=\prevgraf%
3007 \global\num@linesR=\prevgraf%
3008 \global\par@line=\z@%
3009 \global\par@lineR=\z@%
3010 \global\l@dpscL=\z@%
3011 \global\l@dpscR=\z@%
3012 \writtenlinesLfalse%
3013 \writtenlinesRfalse%
3014 \get@familiarfootnote@number%
3015 %

```

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

```

3016 \let\print@Xnotes\print@Xnotes@forpages%
3017 \let\print@notesX\print@notesX@forpages%
3018 %

```

Check if there are chunks to be processed.

```

3019 \check@pstarts%
3020 \loop\if@pstarts%
3021 %

```

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

```

3022 \global\advance\l@dpscL \@ne%
3023 \global\advance\l@dpscR \@ne%
3024 %

```

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

```

3025 \getlinesfromparlistL%
3026 \getlinesfromparlistR%
3027 \l@dcalc@maxoftwo{@cs@linesinparL}{@cs@linesinparR}%
3028 {\usernamecount{l@dmaxlinesinpar\the\l@dpscL}}%
3029 \check@pstarts%
3030 \repeat%
3031 %

```

Zero the counts again, ready for the next bit.

```

3032 \global\l@dpscL=\z0%
3033 \global\l@dpscR=\z0%
3034 %

```

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

```

3035 \getlinesfrompagelistL%
3036 \getlinesfrompagelistR%
3037 \l@dcalc@minoftwo{@cs@linesonpageL}{@cs@linesonpageR}%
3038 {\l@dminpagelines}%
3039 %

```

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.

```

3040 \check@pstarts%
3041 \if@pstarts%
3042 %

```

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

```

3043 \global\advance\l@dpscL \@ne%
3044 \global\advance\l@dpscR \@ne%
3045 \global\advance\pstarts@typeset@L\@ne%
3046 \restore@pstartL@pc%
3047 \restore@pstartR@pc%
3048 %

```

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

```

3049 \global\@donereallinesL=\z0%
3050 \global\@donetotallinesL=\z0%
3051 \global\@donereallinesR=\z0%
3052 \global\@donetotallinesR=\z0%
3053 %

```

Start a loop over the boxes (chunks).

```

3054 \checkraw@text%
3055 %

```

```

3056 % \begingroup
3057 { \loop\ifaraw@text%
3058 %

```

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

```

3059     \checkpageL%
3060     \leftpagehook%
3061     \l@duselanguage{\theledlanguageL}%
3062 {
3063   \loop\ifl@dsamepage%

```

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

```

3064     \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}%
3065     \csuse{before@pstartL@the\l@dpscL}%
3066     \global\csundef{before@pstartL@the\l@dpscL}%
3067     \do@lineL%
3068     \xifinlist{\the\pstarts@typeset@L}{\eled@sections@@}%
3069     {\print@eledsectionL}%
3070     {}%
3071     \advance\numpagelinesL \@ne%
3072 %

```

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

```

3073     \ifshiftedpstarts%
3074     \ifdim\ht\l@dleftbox>0pt%
3075     \parledgroup@correction@notespacing{L}%
3076     \hb@xt@ \hsize{\ledstrut\unhbox\l@dleftbox}%
3077     \else%
3078     \xifinlist{\the\pstarts@typeset@L}{\eled@sections@@
}%
3079     {\add@apparatusL}%
3080     {}%
3081     \unless\ifadvancedshiftedpstarts%
3082     \dimen0=\pagetotal%
3083     \advance\dimen0 by \baselineskip%
3084     \global\pagetotal=\dimen0%
3085     \else%
3086     \ifnomaxlines%
3087     \numdef{\@tmp}{\the\l@dpscL+1}%
3088     \ifcsdef{minpage@pstart@\@tmp}{%
3089     \ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}}%
3090     {\dimen0=\pagetotal%
3091     \advance\dimen0 by \baselineskip%
3092     \global\pagetotal=\dimen0%
3093     }%
3094     {}%
3095     }{}%
3096     \fi%
3097     \fi%
3098     \fi%

```

```

3099         \else%
3100             \parledgroup@correction@notespacing{L}%
3101             \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
3102         \fi%
3103 %

```

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

```

3104         \get@nextboxL%
3105         \global\l@dskipversenumberfalse%
3106         \ifprint@last@after@pendL%
3107             \csuse{after@pendL@the\l@dpscL}%
3108             \global\csundef{after@pendL@the\l@dpscL}%
3109         \fi%
3110         \checkpageL%
3111         \checkverseL%
3112         \checkpbL%
3113         \repeat%
3114 %

```

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

```

3115         \ifl@dpagefull%
3116             \@writelinesonpageL{\the\numpagelinesL}%
3117         \else%
3118             \@writelinesonpageL{1000}%
3119         \fi%
3120 %

```

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

```

3121         \numpagelinesL \z@%
3122         \parledgroup@correction@notespacing@init%
3123         \clearl@dleftpage }%
3124 %

```

Now do the same for the right text.

```

3125         \checkpageR%
3126         \Rightpagehook%
3127         \l@duselanguage{\theledlanguageR}%
3128 {
3129         \loop\ifl@dsamepage%
3130             \set@sectcountR%
3131             \ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{}%
3132             \csuse{before@pstartR@the\l@dpscR}%
3133             \global\csundef{before@pstartR@the\l@dpscR}%
3134             \do@lineR%
3135             \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
3136                 {\print@eledsectionR}%

```

```

3136     {}%
3137     \advance\numpagelinesR \@ne%
3138     \ifshiftedpstarts%
3139         \ifdim\ht\l@drightbox>0pt%
3140             \parledgroup@correction@notespacing{R}%
3141             \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
3142         \else%
3143             \xifinlist{\the\l@dpscr}{\eled@sectionsR@@}%
3144             {\add@apparatusR}%
3145             {}%
3146             \unless\ifadvancedshiftedpstarts%
3147                 \dimen0=\pagetotal%
3148                 \advance\dimen0 by \baselineskip%
3149                 \global\pagetotal=\dimen0%
3150             \else%
3151                 \ifnomaxlines%
3152                     \numdef{\@tmp}{\the\l@dpscr+1}%
3153                     \ifcsdef{minpage@pstart@\@tmp}{%
3154 minpage@pstart@\@tmp}}%
3155                     {\dimen0=\pagetotal%
3156                     \advance\dimen0 by \baselineskip%
3157                     \global\pagetotal=\dimen0%
3158                     }%
3159                     {}%
3160                 }{}%
3161             \fi%
3162             \fi%
3163             \fi%
3164         \else%
3165             \parledgroup@correction@notespacing{R}%
3166             \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
3167         \fi%
3168         \get@nextboxR%
3169         \global\l@dskipversenumberRfalse%
3170         \ifprint@last@after@pendR%
3171             \csuse{after@pendR@\the\l@dpscr}%
3172             \global\csundef{after@pendR@\the\l@dpscr}%
3173         \fi%
3174         \checkpageR%
3175         \checkverseR%
3176         \checkpbR%
3177     \repeat%
3178     \ifl@dpagfull%
3179         \@writelinesonpageR{\the\numpagelinesR}%
3180     \else%
3181         \@writelinesonpageR{1000}%
3182     \fi%
3183     \numpagelinesR=\z@%
3184     \parledgroup@correction@notespacing@init%
3185 %

```

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

```

3186     \clearl@drightpage}%

```

```
3187 %
```

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.

```
3188     \checkraw@text%
3189     \ifaraw@text%
3190         \getlinesfrompagelistL%
3191         \getlinesfrompagelistR%
3192         \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
3193             {\l@dminpagelines}%
3194     \fi%
3195     \repeat}%
3196 %
```

We have now output the text from all the chunks.

```
3197     \fi%
3198 %
```

Make sure that there are no inserts hanging around.

```
3199     \flush@notes%
3200     \flush@notesR%
3201     \endgroup%
3202 %
```

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

```
3203     \global\l@dpscL=\z@%
3204     \global\l@dpscR=\z@%
3205     \global\l@dnumpstartsL=\z@%
3206     \global\l@dnumpstartsR=\z@%
3207     \global\instanzaLfalse%
3208     \global\instanzaRfalse%
3209     \global\inastanzaLfalse%
3210     \global\inastanzaRfalse%
3211     \global\l@dprintingpagesfalse%
3212 %
```

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

```
3213     \ifnum\edtext@later=\edtext@now%
3214     \else%
3215         \led@error@edtext@later@now%
3216     \fi%
3217     \global\edtext@later=\z@%
3218     \global\edtext@now=\z@%
3219 %
```

Prevent final notes from overlapping the line number

```
3220     \finish@Pages@notes%
3221     \ignorespaces}}%
3222
3223
3224 %
```

### 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.

```
3225 \newcommand{\finish@Pages@notes}{%
3226   \def\do##1{%
3227   %
```

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

```
3228   \ifnocritical@%
3229     \global\newnamebox{##1footins}%
3230   \fi
3231   \ifnofamiliar@%
3232     \global\newnamebox{footins##1}%
3233   \fi
3234 %
```

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

```
3235   \restore@Xnotes@settings{##1}%
3236   \restore@notesX@settings{##1}%
3237 %
```

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

```
3238   \ifvoid\csuse{##1footins}%
3239     \ifvoid\csuse{footins##1}\else%
3240       \newpage\null%
3241       \listbreak%
3242     \fi%
3243   \else%
3244     \newpage\null%
3245     \listbreak%
3246   \fi%
3247 }%
3248 \dolistloop{\@series}%
3249 }%
3250 %
```

### XVII.4 Struts

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

```
\ledstrutR
3251 \newcommand*{\ledstrutL}{}
3252 \newcommand*{\ledstrutR}{}
3253
3254 %
```

### 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. `\clearto1@devenpage` is similar except that it first checks to see if it is already on an empty page.



```

3255 \providecommand{\cleartoevenpage}[1][\@empty]{%
3256   \clearpage
3257   \ifodd\c@page\hbox{#1}\clearpage\fi}
3258
3259 \newcommand*\cleartol@devenpage{%
3260   \ifdim\pagetotal<\topskip% on an empty page
3261   \else
3262     \clearpage
3263     \Pages@mainmatter%
3264   \fi
3265   \ifodd\c@page%
3266     \ifprevpgnotnumbered%
3267     \addtocounter{par@page}{-1}%
3268   \fi%
3269   \ifdef{\prevpgstyle}{\thispagestyle{\prevpgstyle}}{%
3270     \hbox{}}\clearpage%
3271   \fi%
3272 }%
3273 %

```

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

```

3274 \newcommand*\clearl@dleftpage{%
3275   \ifdim\pagetotal=0pt\hbox{}}\fi%
3276   \newpage%
3277   \insert@notes@for@onlyside%
3278   \ifodd\c@page\else
3279     \led@err@LeftOnRightPage
3280     \hbox{}}%
3281   \cleardoublepage
3282   \fi}
3283
3284 \newcommand*\clearl@drightpage{%
3285   \ifdim\pagetotal=0pt\hbox{}}\fi%
3286   \newpage%
3287   \insert@notes@for@onlyside%
3288   \ifodd\c@page
3289     \led@err@RightOnLeftPage
3290     \hbox{}}%
3291   \cleartoevenpage
3292   \fi}
3293
3294 %

```

## 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 `\getlinesfromparlistR` `\getlinesfromparlistR` gets the next entry from the `\linesinpar@listR` and puts it into `\@cs@linesinparR`; if the list is empty, it sets `\@cs@linesinparR` to 0.

```

for \getlinesfromparlistR.
3295 \newcommand*\getlinesfromparlistL}{%
3296   \ifx\linesinpar@listL\empty
3297     \gdef\@cs@linesinparL{0}%
3298   \else
3299     \glp\linesinpar@listL\to\@cs@linesinparL
3300   \fi}
3301 \newcommand*\getlinesfromparlistR}{%
3302   \ifx\linesinpar@listR\empty
3303     \gdef\@cs@linesinparR{0}%
3304   \else
3305     \glp\linesinpar@listR\to\@cs@linesinparR
3306   \fi}
3307
3308 %

```

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

```

\getlinesfrompagelistR
\@cs@linesonpageR
3309 \newcommand*\getlinesfrompagelistL}{%
3310   \ifx\linesonpage@listL\empty
3311     \gdef\@cs@linesonpageL{1000}%
3312   \else
3313     \glp\linesonpage@listL\to\@cs@linesonpageL
3314   \fi}
3315 \newcommand*\getlinesfrompagelistR}{%
3316   \ifx\linesonpage@listR\empty
3317     \gdef\@cs@linesonpageR{1000}%
3318   \else
3319     \glp\linesonpage@listR\to\@cs@linesonpageR
3320   \fi}
3321
3322 %

```

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

```

3323 \newcommand*\@writelinesonpageL}[1]{%
3324   \edef\next{\write\linenum@out{\string\@lopL{#1}}}%
3325   \next}
3326 \newcommand*\@writelinesonpageR}[1]{%
3327   \edef\next{\write\linenum@outr{\string\@lopR{#1}}}%
3328   \next}
3329
3330 %

```

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

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

```

3331 \newcommand*\l@dcalc@maxoftwo}[3]{%

```

```

3332 \ifnum #2>#1\relax
3333   #3=#2\relax
3334 \else
3335   #3=#1\relax
3336 \fi}
3337 \newcommand*{\l@dcalc@minoftwo}[3]{%
3338 \ifnum #2<#1\relax
3339   #3=#2\relax
3340 \else
3341   #3=#1\relax
3342 \fi}
3343
3344 %

```

## 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@dpagfull` is set FALSE and `\l@dsamepagetrue` `\ifl@dsamepage` is set TRUE. If the page is spatially full then `\ifl@dpagfull` is set TRUE and `\ifl@dsamepage` is set FALSE. If it is not spatially full but the maximum number of lines have been output then both `\ifl@dpagfull` and `\ifl@dsamepage` are set FALSE.

```

\ifl@dsamepage \checkpageL
\l@dsamepagetrue \ifl@dpagfull
\l@dsamepagefalse \ifl@dsamepage
\ifl@dpagfull TRUE
\l@dpagfulltrue \ifl@dsamepage
\l@dpagfullfalse are set FALSE.
\checkpageL
\checkpageR
3345 \newif\ifl@dsamepage
3346 \l@dsamepagetrue
3347 \newif\ifl@dpagfull
3348
3349 \newcommand*{\checkpageL}{%
3350 \l@dpagfulltrue
3351 \l@dsamepagetrue
3352 \check@goal
3353 \ifdim\pagetotal<\ledthegoal
3354 \ifnum\numpagelinesL<\l@dminpagelines
3355 \else
3356 \ifnomaxlines%
3357 \else%
3358 \l@dsamepagefalse%
3359 \l@dpagfullfalse%
3360 \fi%
3361 \fi
3362 \else
3363 \l@dsamepagefalse
3364 \l@dpagfulltrue
3365 \fi%
3366 \ifprint@last@after@pendL%
3367 \l@dpagfullfalse%
3368 \l@dsamepagefalse%
3369 \print@last@after@pendLfalse%
3370 \fi%
3371 }%
3372
3373 \newcommand*{\checkpageR}{%
3374 \l@dpagfulltrue

```

```

3375 \l@dsamepagetrue
3376 \check@goal
3377 \ifdim\pagetotal<\ledthegoal
3378 \ifnum\numpagelinesR<\l@dmminpagelines
3379 \else
3380 \ifnomaxlines%
3381 \else%
3382 \l@dsamepagefalse%
3383 \l@dpagefullfalse%
3384 \fi%
3385 \fi
3386 \else
3387 \l@dsamepagefalse
3388 \l@dpagefulltrue
3389 \fi%
3390 \ifprint@last@after@pendR%
3391 \l@dpagefullfalse%
3392 \l@dsamepagefalse%
3393 \print@last@after@pendRfalse%
3394 \fi%
3395 }%
3396
3397 %

```

`\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`.

```

3398 \newcommand{\checkpbL}{
3399 \IfStrEq{\led@pb@setting}{after}{
3400 \xifinlist{\the\absline@num}{\l@prev@pb}{\l@dpagefulltrue\
l@dsamepagefalse}{
3401 \xifinlist{\the\absline@num}{\l@prev@nopb}{\l@dpagefullfalse\
l@dsamepagetrue}{
3402 }{
3403 \IfStrEq{\led@pb@setting}{before}{
3404 \numdef{\next@absline}{\the\absline@num+1}
3405 \xifinlist{\next@absline}{\l@prev@pb}{\l@dpagefulltrue\
l@dsamepagefalse}{
3406 \xifinlist{\next@absline}{\l@prev@nopb}{\l@dpagefullfalse\
l@dsamepagetrue}{
3407 }{
3408 }
3409
3410 \newcommand{\checkpbR}{
3411 \IfStrEq{\led@pb@setting}{after}{
3412 \xifinlist{\the\absline@numR}{\l@prev@pbR}{\l@dpagefulltrue\
l@dsamepagefalse}{
3413 \xifinlist{\the\absline@numR}{\l@prev@nopbR}{\l@dpagefullfalse\
l@dsamepagetrue}{
3414 }{
3415 \IfStrEq{\led@pb@setting}{before}{
3416 \numdef{\next@abslineR}{\the\absline@numR+1}
3417 \xifinlist{\next@abslineR}{\l@prev@pbR}{\l@dpagefulltrue\
l@dsamepagefalse}{

```

```

3418 \xifinlist{\next@abslineR}{\l@prev@nopbR}{\l@dpagetruefalse\
l@dsamepagetrue}{\}
3419 }{\}
3420 }
3421 %

```

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

```

3422 \newcommand{\checkverseL}{%
3423 \ifinstanzaL%
3424 \iflednopbinverse%
3425 \ifinserthangingsymbol%
3426 \numgdef{\prev@abslineverse}{\the\absline@num-1}%
3427 \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{\}%
3428 \IfStrEq{\led@pb@setting}{before}{%
3429 \ifnum\numpagelinesL<3%
3430 \ledpbnum{\prev@abslineverse}%
3431 \fi%
3432 }{\}%
3433 \fi%
3434 \fi%
3435 \fi%
3436 }
3437 \newcommand{\checkverseR}{%
3438 \ifinstanzaR%
3439 \iflednopbinverse%
3440 \ifinserthangingsymbolR%
3441 \numgdef{\prev@abslineverse}{\the\absline@numR-1}%
3442 \IfStrEq{\led@pb@setting}{after}{\lednopbnumR{\prev@abslineverse}}{\}%
3443 \IfStrEq{\led@pb@setting}{before}{%
3444 \ifnum\numpagelinesR<3%
3445 \ledpbnumR{\prev@abslineverse}%
3446 \fi%
3447 }{\}%
3448 \fi%
3449 \fi%
3450 \fi%
3451 }
3452 %

```

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

```

3453 \newdimen\ledthegoal
3454 \ifshiftedpstarts
3455 \newcommand*{\@goalfraction}{0.95}
3456 \else
3457 \newcommand*{\@goalfraction}{0.9}
3458 \fi
3459
3460 \newcommand*{\check@goal}{%
3461 \ledthegoal=\@goalfraction\pagegoal}

```

```

3462 \newcommand{\setgoalfraction}[1]{%
3463   \xdef\@goalfraction{#1}%
3464 }
3465 %

```

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

```

\ifwrittenlinesL
3466 \newif\ifwrittenlinesL
3467 \newif\ifwrittenlinesR
3468
3469 %

```

## 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 `nosyncstarts` 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.

```

3470 \newif\if@getnextbox%
3471 %

```

`\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.

```

3472 \newcommand*{\get@nextboxL}{%
3473   \ifvbox\namebox{1@dLcolrawbox\the\l@dpscl}% box is not empty
3474 %

```

The current box is not empty; do nothing.

```

3475   \else%                               box is empty
3476 %

```

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

```

3477   \@getnextboxtrue%Should be local, but be cautious
3478 %

```

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

```

3479   \ifnum\usenamecount{1@dmaxlinesinpar\the\l@dpscl}>\@donetotallinesL
3480     \parledgroup@notes@endL%
3481     \unless\ifnosyncstarts%
3482       \@getnextboxfalse%
3483 %

```

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.

```

3484     \ifnomaxlines%
3485     \ifdim\pagetotal<\ledthegoal%
3486     \numdef{\@tmp}{\l@dpscl+1}%
3487     \ifcsdef{afterlines@pstart@\@tmp R}{%
3488     \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp R}}
%
3489         {}%
3490         {\ifcsdef{minpage@pstart@\@tmp}%
3491         {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
3492         {\ifnum\numpagelinesL=\l@dminpagelines%
3493         \@getnextboxtrue%
3494         \fi%
3495         }%
3496         {\@getnextboxtrue}}%
3497         {\@getnextboxtrue}%
3498         }%
3499     }%
3500     {}%
3501     \fi%
3502     \fi%
3503     \fi%
3504 \else%
3505     \ifnomaxlines%
3506     \numdef{\@tmp}{\the\l@dpscl+1}%
3507     \ifcsdef{minpage@pstart@\@tmp}{%
3508     \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3509     {\ifdimgreater{\pagetotal}{\ledthegoal}%
3510     {\@getnextboxtrue}%
3511     {\@getnextboxfalse}%
3512     }%
3513     {\@getnextboxtrue}%
3514     }-}%
3515     \fi%
3516 \fi%
3517 %

```

Sufficient lines have been output.

```

3518     \if@getnextbox%
3519     \ifnum\usenamecount{\l@dmaxlinesinpar\the\l@dpscl}=\@donetotallinesL
3520     \parledgroup@notes@endL
3521     \fi
3522     \ifwrittenlinesL\else
3523 %

```

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

```

3524     \@writelinesinparL
3525     \writtenlinesLtrue
3526     \fi
3527     \ifnum\l@dnumstartsL>\l@dpscl
3528 %

```

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.





```

3573         {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
3574         {\ifnum\numpagelinesR=\l@dminpagelines%
3575           \@getnextboxtrue%
3576           \fi%
3577         }%
3578         {\@getnextboxtrue}}%
3579         {\@getnextboxtrue}%
3580       }%
3581     }%
3582   }%
3583   \fi%
3584   \fi%
3585   \fi%
3586   \else%
3587     \ifnomaxlines%
3588       \numdef{\@tmp}{\the\l@dpscR+1}%
3589       \ifcsdef{minpage@pstart@\@tmp}{%
3590         \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3591         {\ifdimgreater{\pagetotal}{\ledthegoal}%
3592           {\@getnextboxtrue}%
3593           {\@getnextboxfalse}%
3594         }%
3595         {\@getnextboxtrue}%
3596       }{}
3597     \fi%
3598   \fi%
3599   \if@getnextbox%
3600     \ifnum\usenamecount{l@dmaxlinesinpar\the\l@dpscR}=\@donetotallinesR
3601       \parledgroup@notes@endR
3602     \fi
3603     \ifwrittenlinesR\else
3604       \@writelinesinparR
3605       \writtenlinesRtrue
3606     \fi
3607     \ifnum\l@dnumpstartsR>\l@dpscR
3608       \writtenlinesRfalse
3609       \ifbypstart@R
3610         \unless\ifinstanzaR%
3611           \global\line@numR=0%
3612           \resetprevline@%
3613         \fi%
3614       \fi
3615       \csuse{after@pendR@\the\l@dpscR}%
3616       \global\csundef{after@pendR@\the\l@dpscR}%
3617       \l@dcalc@maxoftwo{\the\usenamecount{l@dmaxlinesinpar\the\l@dpscR}}%
3618         {\the\@donetotallinesR}%
3619         {\usenamecount{l@dmaxlinesinpar\the\l@dpscR}}%
3620       \global\@donetotallinesR \z@
3621       \global\advance\l@dpscR \@ne
3622       \global\pstartnumRtrue%
3623       \restore@pstartR@pc%
3624       \parledgroup@notes@endR
3625       \parledgroup@correction@notes@spacing@final{R}
3626     \else

```

```

3627 \print@last@after@pendRtrue%
3628 \fi
3629 \fi
3630 \fi}
3631
3632 %

```

## 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@mainmatter 33 \newif\ifPages@mainmatter
\Pages@mainmatter 34 \newcommand{\Pages@mainmatter}{%
3635 \ifPages@mainmatter%
3636 \pagenumbering{arabic}%
3637 \addtocounter{page}{1}%
3638 \addtocounter{par@page}{-1}%
3639 \patchcmd{\thepage}{page}{par@page}{-}{-}%
3640 \fi%
3641 }
3642 %

```

## 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 343 \pretocmd{\chapter}{%
3644 \ifl@dprintingpages%
3645 \vbox{}%
3646 \fi%
3647 }%
3648 {}%
3649 {}%
3650 %

```

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

```

3651 \newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}}
3652 \eledsectnotoc{R}
3653 %

```

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

```

3654 \newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}}
3655 \eledsectmark{L}
3656 %

```

`\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`.

```

3657 \newskip\eledsection@correcting@skip
3658 %

```

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

```

3659 \newwrite\eled@sectioningR@out
3660 %

```

## 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).

```

3661 \def\l@prev@pbR{}
3662 \def\l@prev@nopbR{}
3663 %

```

`\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.

```

3664 \newcommand{\ledpbR}{\write\linenum@outR{\string\led@pbR}}
3665 \newcommand{\ledpbnumR}[1]{\write\linenum@outR{\string\led@pbnumR{#1}}}
3666 \newcommand{\lednopbR}{\write\linenum@outR{\string\led@nopbR}}
3667 \newcommand{\lednopbnumR}[1]{\write\linenum@outR{\string\led@nopbnumR{#1}}}
3668 %

```

`\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.

```

3669 \newcommand{\led@pbR}{\listxadd{\l@prev@pbR}{\the\absline@numR}}
3670 \newcommand{\led@pbnumR}[1]{\listxadd{\l@prev@pbR}{#1}}
3671 \newcommand{\led@nopbR}{\listxadd{\l@prev@nopbR}{\the\absline@numR}}
3672 \newcommand{\led@nopbnumR}[1]{\listxadd{\l@prev@nopbR}{#1}}
3673 %

```

## 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).

```
3674 \newmarks\parledgroup@
3675 \newmarks\parledgroup@series
3676 \newmarks\parledgroup@type
3677 %
```

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

```
3678 \newcommand{\parledgroup@notes@startL}{%
3679   \ifnum\usernamecount{1@dmaxlinesinpar\the\1@dpscL}>0%
3680   \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
3681     bhooknoteX@\splitfirstmarks\parledgroup@series}}{}%
3682   \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
3683     bhookXnote@\splitfirstmarks\parledgroup@series}}{}%
3684   \fi%
3685   \global\ledgroupnotesL@true%
3686   \insert@noterule@ledgroup{L}%
3687 }
3688 \newcommand{\parledgroup@notes@startR}{%
3689   \ifnum\usernamecount{1@dmaxlinesinpar\the\1@dpscR}>0%
3690   \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
3691     bhooknoteX@\splitfirstmarks\parledgroup@series}}{}%
3692   \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
3693     bhookXnote@\splitfirstmarks\parledgroup@series}}{}%
3694   \fi%
3695   \global\ledgroupnotesR@true%
3696   \insert@noterule@ledgroup{R}%
3697 }
3698 %
```

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

```
3695 \newcommand{\parledgroup@notes@endL}{%
3696   \global\ledgroupnotesL@false%
3697 }
3698 \newcommand{\parledgroup@notes@endR}{%
3699   \global\ledgroupnotesR@false%
3700 }
3701 %
```

`\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`

```
3702 \newcommand{\insert@noterule@ledgroup}[1]{
3703   \IfStrEq{\splitbotmarks\parledgroup@}{begin}{%
3704     \IfStrEq{\splitbotmarks\parledgroup@type}{Xfootnote}{
```

```

3705     \csuse{ifledgroupnotes#1@}
3706     \vskip\skip\csuse{mp\splitbotmarks\parledgroup@series footins}
3707     \csuse{\splitbotmarks\parledgroup@series footnoterule}
3708     \fi
3709   }
3710   {}
3711   \IfStrEq{\splitbotmarks\parledgroup@type}{footnoteX}{
3712     \csuse{ifledgroupnotes#1@}
3713     \vskip\skip\csuse{mpfootins\splitbotmarks\parledgroup@series}
3714     \csuse{footnoterule\splitbotmarks\parledgroup@series}
3715     \fi
3716     }{}
3717   }
3718   {}
3719 }
3720 %

```

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

```

3721 \newcommand{\setparledgroupnotespacing}[1]{\gdef\@parledgroupnotespacing
3722   {#1}}
3723 \newcommand{\@parledgroupnotespacing}{}
3724 %

```

`\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`.

```

3724 \dimdef{\parledgroup@notespacing@correction}{0pt}
3725 \newcommand{\parledgroup@notespacing@set@correction}{%
3726   {\@getfirstseries\csuse{\Xnotefontsize@\@firstseries}}%We suppose all the
3727   series has the same footnote size setup
3728   \@parledgroupnotespacing\dimgdef{\temp@spacing}{\baselineskip}}%
3729   \dimgdef{\parledgroup@notespacing@correction}{\baselineskip-\temp@spacing}
3730 }%
3731 }
3732 %

```

`\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.

```

3731 \newcommand{\parledgroup@correction@notespacing@init}{
3732   \dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}
3733   \dimdef{\parledgroup@notespacing@correction@modulo}{0pt}
3734 }
3735 \parledgroup@correction@notespacing@init
3736 %

```

`\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`.

```

3737 \newcommand{\parledgroup@correction@notespacing@final}[1]{
3738   \ifparledgroup
3739   \vspace{\parledgroup@notespacing@correction@accumulated}
3740   \parledgroup@correction@notespacing@init%
3741   \ifstrequal{#1}{L}{
3742     \numdef{\@checking}{\the\l@dpscL-1}
3743   }{
3744     \numdef{\@checking}{\the\l@dpscR-1}
3745   }
3746   \dimdef{\@beforenotes@current@diff}{\csuse{@parledgroup@beforenotes@
3747   @checking L}-\csuse{@parledgroup@beforenotes@@checking R}}%
3748   \ifstrequal{#1}{L}{%
3749     {% Left
3750       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{-\
3751   @beforenotes@current@diff}}%
3752     }%
3753     {% Right
3754       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{\
3755   @beforenotes@current@diff}}%
3756     }%
3757   \fi
3758 }
3759 %

```

`\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@correction@accumulated` and `\parledgroup@notespacing@correction@modulo`. If `\parledgroup@notespacing@correction@modulo` is equal or greater than `\baselineskip`:

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

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

```

3757 \newcommand{\parledgroup@correction@notespacing}[1]{%
3758   \csuse{ifledgroupnotes#1@}%
3759   \vspace{-\parledgroup@notespacing@correction}%
3760   \dimdef{\parledgroup@notespacing@correction@accumulated}{\
3761   parledgroup@notespacing@correction@accumulated+
3762   \parledgroup@notespacing@correction}%
3763   \dimdef{\parledgroup@notespacing@correction@modulo}{\
3764   parledgroup@notespacing@correction@modulo+
3765   \parledgroup@notespacing@correction}%
3766   \ifdimless{\parledgroup@notespacing@correction@modulo}{\baselineskip}
3767   }{\advance\numpagelinesL -\@one%
3768   \dimdef{\parledgroup@notespacing@correction@modulo}{\
3769   parledgroup@notespacing@correction@modulo-\baselineskip}%
3770   }% mean greater than equal
3771   \fi%
3772 }
3773 %

```

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

```
3768 \dimdef\parledgroup@beforenotesL{0pt}
3769 \dimdef\parledgroup@beforenotesR{0pt}
3770 %
```

`\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.

```
3771 \newcommand{\parledgroup@beforenotes@save}[1]{
3772   \ifparledgroup
3773     \csdimgdef{@parledgroup@beforenotes@\the\csuse{1@dnumstarts#1}#1}{\
csuse{parledgroup@beforenotes#1}}
3774     \csdimgdef{parledgroup@beforenotes#1}{0pt}
3775   \fi
3776 }
3777 %
```

## XXII Compatibility with eledmac

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

```
3778 \ifeledmaccompat%
3779
3780
3781   \unless\ifnocritical@
3782   \let\onlyXside\Xonlyside
3783   \fi
3784 \fi
3785 %
```

## 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. 388](#).

## 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. 15](#).

## 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> .....	<u>1</u>
<code>\@astanza@line</code> .....	<u>1</u>
<code>\@cs@linesinparL</code> .....	<u>1</u>
<code>\@cs@linesinparR</code> .....	<u>1</u>
<code>\@cs@linesonpageL</code> .....	<u>1</u>
<code>\@cs@linesonpageR</code> .....	<u>1</u>
<code>\@donereallinesL</code> .....	<u>1</u>

<code>\@donereallinesR</code> .....	<u>1</u>
<code>\@donetotallinesL</code> .....	<u>1</u>
<code>\@donetotallinesR</code> .....	<u>1</u>
<code>\@eledsectionL</code> .....	<u>1</u>
<code>\@eledsectionR</code> .....	<u>1</u>
<code>\@lab</code> .....	<u>1</u>
<code>\@lopL</code> .....	<u>1</u>
<code>\@lopR</code> .....	<u>1</u>
<code>\@morespace@leftnote@rightcolumn</code> .....	<u>1</u>
<code>\@morespace@rightnote@leftcolumn</code> .....	<u>1</u>
<code>\@nl</code> .....	<u>1</u>
<code>\@nl@regR</code> .....	<u>1</u>
<code>\@par@sync@option</code> .....	<u>1</u>
<code>\@par@this@sync@option</code> .....	<u>1</u>
<code>\@parledgroupnotespacing</code> .....	<u>1</u>
<code>\@pend</code> .....	<u>1</u>
<code>\@pendR</code> .....	<u>1</u>
<code>\@pstart</code> .....	<u>1</u>
<code>\@pstartR</code> .....	<u>1</u>
<code>\@pstartsfalse</code> .....	<u>1</u>
<code>\@pstartstrue</code> .....	<u>1</u>
<code>\@ref</code> .....	<u>1</u>
<code>\@ref@reg@parseR</code> .....	<u>1</u>
<code>\@ref@regR</code> .....	<u>1</u>
<code>\@set</code> .....	<u>1</u>
<code>\@stopastanza</code> .....	<u>1</u>
<code>\@writelinesinparL</code> .....	<u>1</u>
<code>\@writelinesinparR</code> .....	<u>1</u>
<code>\@writelinesonpageL</code> .....	<u>1</u>
<code>\@writelinesonpageR</code> .....	<u>1</u>
<code>\@writepageofparL</code> .....	<u>1</u>
<code>\@writepageofparR</code> .....	<u>1</u>
COMMAND\+ .....	53
COMMAND\@Rlineflag .....	84, 154
COMMAND\@adv .....	44, 152
COMMAND\@cs@linesinparL .....	113
COMMAND\@cs@linesonpageL .....	114
COMMAND\@eledsectionL .....	94
COMMAND\@eledsectionR .....	94
COMMAND\@eledsectmark .....	122
COMMAND\@eledsectnotoc .....	122
COMMAND\@goalfraction .....	14, 117
COMMAND\@l@dtempcnta .....	78
COMMAND\@lab .....	42, 85, 152
COMMAND\@lopL .....	48, 108, 114
COMMAND\@lopR .....	48, 114
COMMAND\@morespace@leftnote@rightcolumn .....	85, 86
COMMAND\@morespace@rightnote@leftcolumn .....	85, 86
COMMAND\@namedef .....	30
COMMAND\@namuse .....	30
COMMAND\@nl .....	43, 49, 50, 85, 152
COMMAND\@nl@regR .....	43
COMMAND\@page .....	85
COMMAND\@par@sync@option .....	42

COMMAND\@parledgroupnotespacing	125
COMMAND\@pend	47
COMMAND\@pendR	47
COMMAND\@pstart	48
COMMAND\@pstartstrue	93
COMMAND\@ref	45–47, 50, 153
COMMAND\@ref@later	47
COMMAND\@ref@reg@parseargR	47
COMMAND\@ref@regR	46
COMMAND\@set	44, 153
COMMAND\@sw	46
COMMAND\AtBeginPairs	9, 56, 151, 155
COMMAND\AtEveryPend	11, 151, 152
COMMAND\AtEveryPend*	11
COMMAND\AtEveryPstart	3, 11, 20, 64, 96, 151, 152
COMMAND\AtEveryPstart*	11
COMMAND\AtEveryPstartCall	3, 20, 64, 152
COMMAND\AtEveryStanza	154
COMMAND\AtEveryStopStanza	154
COMMAND\Clear the right lines for \read@linelist	42
COMMAND\Columns	8, 9, 20, 54, 95, 102, 148–150, 152–154
COMMAND\Columns@print@after@pend	104
COMMAND\Columns@print@before@pstart	104
COMMAND\Lcolwidth	9, 12, 106, 130
COMMAND\Lcolwidthth	155
COMMAND\Leftpagehook	21, 157
COMMAND\Leftsidehook	148
COMMAND\Leftsidehookend	148
COMMAND\MakePerPage	15
COMMAND\Pages	5, 8, 12, 14, 20, 27, 53, 54, 75, 79, 81, 105, 106, 112, 122, 125, 148, 150–154, 156
COMMAND\Pages@mainmatter	122
COMMAND\Pairs	53
COMMAND\Rcolwidth	9, 12, 106, 130
COMMAND\Rcolwidthth	155
COMMAND\Rightpagehook	21, 157
COMMAND\Rightsidehook	148
COMMAND\Rightsidehookend	148
COMMAND\Rlineflag	128
COMMAND\Xendlineflag	17, 154
COMMAND\Xlineflag	16, 129, 154
COMMAND\Xmaxhnotes	15
COMMAND\Xnoteswidthliketwocolumns	10, 151
COMMAND\Xonlyside	15, 79, 129, 155, 157
COMMAND\Xparindent	157
COMMAND\Xpstart	156
COMMAND\Xtxtbeforenotes	154
COMMAND\&	21
COMMAND\absline@numR	41
COMMAND\add@apparatusL	67
COMMAND\add@apparatusR	69
COMMAND\add@penalties	78
COMMAND\add@penaltiesL	78
COMMAND\advanceline	44, 50, 153
COMMAND\affixline@num	73

COMMAND\affixline@numR	73, 148, 149
COMMAND\affixpstart@num	75
COMMAND\affixpstart@numR	75
COMMAND\affixside@note	86
COMMAND\aftercolumnseparator	10, 103, 150, 155, 156
COMMAND\araw@textfalse	93
COMMAND\araw@texttrue	93
COMMAND\at@begin@pairs	56
COMMAND\autopar	20, 156
COMMAND\ballast@count	78
COMMAND\baselineskip	102, 126
COMMAND\bbl@set@language	90, 153
COMMAND\beforecolumnseparator	10, 103, 150, 156
COMMAND\begin	21
COMMAND\beginnumbering	10, 11, 17, 18, 20, 31, 42, 103, 149, 150, 152, 155, 156
COMMAND\beginnumberingR	49
COMMAND\bf	149
COMMAND\bfseries	149
COMMAND\brokenpenalty	78
COMMAND\by@autopar	60
COMMAND\chapter	122, 148
COMMAND\check@goal	117
COMMAND\check@pstarts	93
COMMAND\checkpageL	108, 115
COMMAND\checkpb@columns	102
COMMAND\checkpbL	116
COMMAND\checkpbR	116
COMMAND\checkraw@text	93
COMMAND\checkverseL	117
COMMAND\checkverseR	117
COMMAND\clear(double)page	112
COMMAND\clearl@dleftpage	113
COMMAND\clearl@drightpage	113
COMMAND\clearpage	113, 151
COMMAND\cleartoevenpage	112
COMMAND\cleartol@devenpage	112, 122
COMMAND\columnrulewidth	9, 102
COMMAND\columns	155
COMMAND\columns@position	102
COMMAND\columnseparator	9
COMMAND\columnspan	10, 150
COMMAND\continuousnumberingwithcolumns	156
COMMAND\correct@Xfootins@box	152
COMMAND\correct@footinsX@box	152
COMMAND\critext	151
COMMAND\csname	54
COMMAND\displaywidowpenalty	78
COMMAND\do@actions	71
COMMAND\do@actions@fixedcode	148
COMMAND\do@actions@nextR	71
COMMAND\do@actionsR	71, 148
COMMAND\do@ballast	78
COMMAND\do@ballastR	71
COMMAND\do@insidelineLhook	149

COMMAND\do@insidelineRhook	149
COMMAND\do@line	64
COMMAND\do@line(L/R)	68
COMMAND\do@lineL	65, 78, 148, 149
COMMAND\do@lineLhook	148
COMMAND\do@lineR	68, 148–150
COMMAND\do@lineRhook	148
COMMAND\do@lockoff	153
COMMAND\do@lockoffR	45
COMMAND\do@lockon	153
COMMAND\do@lockonR	44
COMMAND\doinsidelineLhook	21, 130, 151, 155
COMMAND\doinsidelineRhook	21, 130, 151, 155
COMMAND\doinsidelinehook	21
COMMAND\doinsidethislinehook	157, 158
COMMAND\dolineLhook	21, 151
COMMAND\dolineRhook	21, 151
COMMAND\dolinehook	21
COMMAND\edindex	151
COMMAND\edlabel	149, 151
COMMAND\edlineref	155
COMMAND\edtext	16, 45, 50, 82, 151, 155, 156
COMMAND\edtext@later	82, 84, 111
COMMAND\edtext@now	111
COMMAND\edtextlater	16, 82, 83, 155, 156
COMMAND\edtextnow	16, 82–84, 155
COMMAND\eled@sectioningR@out	123
COMMAND\eledchapter	152
COMMAND\eledsection	151–153, 157
COMMAND\eledsection@correcting@skip	123
COMMAND\eledsectmark	23, 122
COMMAND\eledsectnotoc	23, 122
COMMAND\eledxxx	150
COMMAND\end	21
COMMAND\endgraf	62
COMMAND\endlock	50, 153
COMMAND\endnumbering	10, 11, 17, 20, 33, 152, 155, 156
COMMAND\endnumberingR	49
COMMAND\endsub	50, 153
COMMAND\endumbering	17
COMMAND\expandafter	52
COMMAND\extensionchars	31
COMMAND\firstlinenum	18, 150, 153, 154
COMMAND\firstsublinenum	150, 153, 154
COMMAND\fix@page	44, 153
COMMAND\flag@end	50, 148, 150
COMMAND\flag@start	50, 150
COMMAND\flush@notesR	78
COMMAND\footnote	54
COMMAND\footnoteX	51, 52
COMMAND\footnoteX@reading	154
COMMAND\footnoteXmk	16
COMMAND\footnoteXnomk	16, 52, 158
COMMAND\frontmatter	14, 23

COMMAND\get@intercolumns@width	98, 101
COMMAND\get@nextboxL	149
COMMAND\get@nextboxR	149
COMMAND\get@sidenote@morespace@columns	85, 86
COMMAND\getline@numL	70
COMMAND\getline@numR	70
COMMAND\getlinesfrompagelistL	114
COMMAND\getlinesfrompagelistR	114
COMMAND\getlinesfromparlistL	113
COMMAND\getlinesfromparlistR	114
COMMAND\gl@p	52
COMMAND\goalfraction	128
COMMAND\hangingsymbol	128, 148
COMMAND\hfill	97, 103
COMMAND\hidenumbering	18, 152
COMMAND\if@getnextbox	118
COMMAND\if@sidenotesmarginpage	85, 86
COMMAND\ifPages@mainmatter	122
COMMAND\ifbypage@	152
COMMAND\ifbypstart@R	152
COMMAND\ifcsboxvoid	31
COMMAND\ifdim	97
COMMAND\ifinserthangingsymbol	87
COMMAND\ifinserthangingsymbolR	87
COMMAND\ifl@dpagefull	115
COMMAND\ifl@dpageing	27, 151
COMMAND\ifl@dpairing	27, 148
COMMAND\ifl@dsamelang	150
COMMAND\ifl@dsamepage	115
COMMAND\ifl@pagefull	108
COMMAND\iflledRcol	27
COMMAND\iflinenumberLevenifblank	40
COMMAND\iflinenumberRevenifblank	40
COMMAND\iflledRcol	148
COMMAND\ifmovecolumnspositiononrightpage	26
COMMAND\ifnumberedpar@	58
COMMAND\ifnumberingR	148
COMMAND\ifnumberline	54
COMMAND\ifnumberpstart	54
COMMAND\ifpst@rtedL	31, 33, 59, 148
COMMAND\ifpst@rtedR	31
COMMAND\ifsublines@	44
COMMAND\ifsublines@R	157
COMMAND\insert@countR	45
COMMAND\insert@noterule@ledgroup	124
COMMAND\insert@notes@for@onlyside	81
COMMAND\insertlines@list	45
COMMAND\insertlines@listR	46
COMMAND\inserts@list	58
COMMAND\inserts@listR	77
COMMAND\intercolumns@width	98
COMMAND\l@d@nums	50
COMMAND\l@d@set	44, 50, 153
COMMAND\l@dLcolrawbox	58

COMMAND\l@dLcolrawbox1	92
COMMAND\l@dLcolrawbox2	92
COMMAND\l@dRcolrawbox	58
COMMAND\l@dcalc@maxoftwo	114
COMMAND\l@dcalc@minoftwo	114
COMMAND\l@dchecklang	148, 150
COMMAND\l@dcsnote	150
COMMAND\l@dleftbox	64, 108, 151
COMMAND\l@dlinenumR	40, 148
COMMAND\l@dlsnote	150
COMMAND\l@dmake@labels	84
COMMAND\l@dmaxlinesinpar	106
COMMAND\l@dmaxlinesinpar1	92
COMMAND\l@dminpagelines	107, 148
COMMAND\l@dnumpstartsL	92, 148
COMMAND\l@dprintingcolumnstrue	151
COMMAND\l@dprintingpagestrue	151
COMMAND\l@dpscL	95, 106, 107, 119
COMMAND\l@dpscR	95, 106, 107
COMMAND\l@drsnote	150
COMMAND\l@dsetupmaxlinecounts	92
COMMAND\l@duselanguage	90, 148
COMMAND\l@dzeromaxlinecounts	92
COMMAND\l@prev@nopbR	123
COMMAND\l@prev@pbR	123
COMMAND\labelpstarttrue	149
COMMAND\labelref@list	85
COMMAND\labelref@listR	84
COMMAND\lang	90
COMMAND\last@page@numR	44
COMMAND\led	149
COMMAND\led@nopbR	123
COMMAND\led@nopbnumR	123
COMMAND\led@pbR	123
COMMAND\led@pbnumR	123
COMMAND\ledinnernote	22
COMMAND\ledleftnote	22
COMMAND\lednopb	22, 102, 116
COMMAND\lednopbR	123
COMMAND\lednopbnumR	123
COMMAND\ledouternote	22
COMMAND\ledpb	102, 116
COMMAND\ledpbR	123
COMMAND\ledpbnumR	123
COMMAND\ledrightnote	22
COMMAND\ledsidenote	22
COMMAND\ledstrutL	148
COMMAND\ledstrutR	148, 153
COMMAND\ledthegoal	117
COMMAND\ledtrutL	148, 153
COMMAND\leftlinenumR	40, 148
COMMAND\leftlinenumannotationR	41
COMMAND\lemma	83
COMMAND\let	52



COMMAND\line@list@R	46
COMMAND\line@list@stuff	42, 49
COMMAND\line@margin	36
COMMAND\line@marginR	36, 148
COMMAND\line@numR	41
COMMAND\lineation	18, 152, 156
COMMAND\lineation*	18, 36, 150
COMMAND\lineationR	18, 35, 152, 155
COMMAND\linenum	83
COMMAND\linenum@out	85
COMMAND\linenum@outR	48
COMMAND\linenumOnlyPagesForColumns	19, 37, 156
COMMAND\linenumOnlyPagesForColumnsR	19, 37, 156, 158
COMMAND\linenumannotationothersidetrue	157
COMMAND\linenumberLevenifblanktrue	19, 154
COMMAND\linenumberRevenifblank	154
COMMAND\linenumberRevenifblanktrue	19
COMMAND\linenumberannotation	156
COMMAND\linenumberlist	18, 129, 155
COMMAND\linenumberlistR	18, 129, 155
COMMAND\linenumberstyle	18, 129, 155
COMMAND\linenumberstyle*	129
COMMAND\linenumberstyleR	18, 129
COMMAND\linenumincrement	18, 150, 153, 154
COMMAND\linenummargin	19, 36, 148, 153, 154
COMMAND\linenummargin*	19, 36, 153
COMMAND\linenummarginColumns	19, 37
COMMAND\linenummarginColumns*	19
COMMAND\linenummarginColumnsR	19
COMMAND\linenummarginR	19, 36, 153
COMMAND\linenumrepR	39, 148
COMMAND\linesinpar@listL	47, 113
COMMAND\linesonpage@listL	48, 114
COMMAND\lock@off	45
COMMAND\lock@on	44
COMMAND\lockoff	158
COMMAND\lockon	158
COMMAND\mainmatter	3, 14, 23, 153
COMMAND\makeatletter	67
COMMAND\markboth	130, 155
COMMAND\maxchunks	8, 21, 92, 93
COMMAND\maxdimen	79, 80
COMMAND\maxhnotesX	15
COMMAND\memorydump	17, 35
COMMAND\movecolumnsonrightpage	101
COMMAND\msdata	155
COMMAND\n@num	152
COMMAND\new@lineL	49
COMMAND\new@lineR	50
COMMAND\newhookcommand@series	53
COMMAND\newif	152
COMMAND\newpage	112, 113, 151
COMMAND\newseries	54
COMMAND\newseries@	51

COMMAND\newseries@par	51, 53, 54
COMMAND\next@line@list@stuffR	49, 156
COMMAND\noeledxxx	151
COMMAND\nomark@	52
COMMAND\nomaxlines	48
COMMAND\notesXwidthliketwocolumns	10, 151
COMMAND\num@lines	78
COMMAND\num@lines(R)	58
COMMAND\numberingR	33
COMMAND\numberlinefalse	8, 157
COMMAND\numberlinetrue	157
COMMAND\numberonlyfirstinline	149
COMMAND\numberpstartfalse	18, 156
COMMAND\numberpstarttrue	18, 148, 153, 156
COMMAND\numberstantztrue	157
COMMAND\one@line	58, 87
COMMAND\one@lineR	58
COMMAND\onlyXside	129
COMMAND\onlysideX	15, 79, 81, 152, 155, 157
COMMAND\page@action	44, 153
COMMAND\pagenumbering	153
COMMAND\pages	14
COMMAND\pagetotal	108, 151
COMMAND\par@line	78
COMMAND\par@line(R)	58
COMMAND\par@sync@option	25
COMMAND\parindentX	157
COMMAND\parledgroup@	124
COMMAND\parledgroup@beforenotes@save	127
COMMAND\parledgroup@beforenotesL	127
COMMAND\parledgroup@beforenotesR	127
COMMAND\parledgroup@correction@notespacing	126
COMMAND\parledgroup@correction@notespacing@final	125
COMMAND\parledgroup@correction@notespacing@init	125
COMMAND\parledgroup@notes@endL	124
COMMAND\parledgroup@notes@endR	124
COMMAND\parledgroup@notes@startL	124
COMMAND\parledgroup@notes@startR	124
COMMAND\parledgroup@notespacing@correction	125, 126
COMMAND\parledgroup@notespacing@correction@accumulated	126
COMMAND\parledgroup@notespacing@correction@modulo	126
COMMAND\parledgroup@notespacing@set@correction	125
COMMAND\parledgroup@series	124
COMMAND\parledgroup@type	124
COMMAND\parledgroupnotespacing	128
COMMAND\parledgrouptrue	22
COMMAND\patchcmd	153
COMMAND\pausenumbering	10, 11, 34, 35, 154
COMMAND\pend	4, 8, 11, 12, 20–22, 50, 54, 58, 62–64, 92, 104, 150–153
COMMAND\pendL	150
COMMAND\pendR	150
COMMAND\pends	20
COMMAND\perpage	15
COMMAND\prev@nopbR	123

COMMAND\prev@pbR	123
COMMAND\prevpgstyle	26
COMMAND\print@Xnotes	79
COMMAND\print@Xnotes@forpages	79, 80, 152
COMMAND\print@columnseparator	97, 150
COMMAND\print@eledsectionL	66
COMMAND\print@leftcolumn	101
COMMAND\print@line	66
COMMAND\print@lineL	66
COMMAND\print@notesX@forpages	152
COMMAND\print@rightcolumn	101
COMMAND\printlinenumR	129
COMMAND\printlinesR	154
COMMAND\pstart	4,
8, 11, 12, 18, 20–22, 35, 44, 50, 54, 58, 59, 62, 64, 92, 95, 96, 104, 148, 150, 152, 153, 155–157	
COMMAND\pstartL	64, 150
COMMAND\pstartR	64, 149, 150
COMMAND\pstartinfootnote	152
COMMAND\pstarts@typeset@L	95
COMMAND\raw@text	92
COMMAND\read@linelist	42, 132, 153
COMMAND\ref@reg	46
COMMAND\ref@regR	46, 153
COMMAND\relax	129
COMMAND\reledmac	153
COMMAND\renewcommand	21, 128
COMMAND\resumenumbering	10, 11, 34, 35, 150, 154
COMMAND\resumenumberingR	151
COMMAND\rightlinenumR	40, 148
COMMAND\rightlinenumannotationR	41
COMMAND\sameword	83
COMMAND\section	148
COMMAND\section@num	31
COMMAND\selectlanguage	20, 90
COMMAND\set@line	50, 153
COMMAND\set@line@action	44, 153
COMMAND\setRlineflag	19, 128
COMMAND\setgoalfraction	14, 128
COMMAND\sethangingsymbol	21
COMMAND\setline	44, 50, 153
COMMAND\setlinenum	44, 50, 153
COMMAND\setmsdataposition	155
COMMAND\setnoteposition...	103
COMMAND\setparledgroupnotespacing	128, 153
COMMAND\setposition...	103
COMMAND\setstanzaindents	10, 21, 128, 130
COMMAND\setwidthliketwocolumns	103
COMMAND\sidenotemargin	22, 150
COMMAND\sidenotemargin*	22, 150
COMMAND\sidenotemarginR	22, 155
COMMAND\skipnumbering	18, 152
COMMAND\sloppy	9
COMMAND\stanza	8, 10, 18, 21, 88, 129, 130, 149
COMMAND\stanzanumtrue	22

COMMAND\startlock	50, 153
COMMAND\startsub	50, 153
COMMAND\sub@action	44, 153
COMMAND\sub@off	85
COMMAND\sub@on	85
COMMAND\subline@numR	41
COMMAND\sublinenumberstyle	18, 129
COMMAND\sublinenumberstyle*	129
COMMAND\sublinenumberstyleR	18, 129
COMMAND\sublinenumincrement	150, 153, 154
COMMAND\sublinenumrepR	39, 148
COMMAND\syntaxonly	154
COMMAND\sza@0@	21
COMMAND\textheight	15
COMMAND\textwidth	56
COMMAND\the@labelX	154
COMMAND\theledlanguageL	90
COMMAND\theledlanguageR	90
COMMAND\thepage	23
COMMAND\thepstartL	18, 149
COMMAND\thepstartR	18, 149
COMMAND\thestanzaL	22
COMMAND\thestanzaR	22
COMMAND\this@section@last@page@numR	44
COMMAND\vbox	59
COMMAND\vskip	124
COMMAND\vsplit	78
COMMAND\widthliketwocolumns	10
COMMAND\widthliketwocolumnsfalse	10
COMMAND\widthliketwocolumnstrue	10
COMMAND\xflagref	154
COMMAND\xright@appenditem	52
COMMAND\xspace	24
COMMAND\xxxfootstart	103
ENVIRONMENTLeftside	11, 56, 101
ENVIRONMENTRightside	11, 57, 101
ENVIRONMENTastanza	21, 88, 129, 130, 153, 154, 156, 157
ENVIRONMENTcolumns	33, 154
ENVIRONMENTledgroup	7, 154
ENVIRONMENTleft	18
ENVIRONMENTpages	27, 33, 55, 130, 152, 154, 155
ENVIRONMENTpairs	55, 152, 154
PACKAGEEDMAC	130
PACKAGEEDSTANZA	130
PACKAGEEledmac	152
PACKAGEEledpar	152
PACKAGETABMAC	130
PACKAGEbabel	20, 89–91, 154, 157
PACKAGEcalc	157
PACKAGEeledmac	5, 92, 127, 128, 130, 150, 152
PACKAGEeledpar	5, 6, 14, 38, 128, 129, 150, 151
PACKAGEetoolbox	31, 97, 123
PACKAGEledmac	6
PACKAGEledpar	2, 6

PACKAGEMemoir	130
PACKAGEMusixtex	150
PACKAGEperpage	3, 15, 129, 154
PACKAGEpolyglossia	20, 89–91, 154, 157, 158
PACKAGEreledmac	2, 4, 6–8, 10, 11, 14, 15, 18–24, 26, 27, 31, 33, 36, 38, 41, 42, 44–54, 66, 75, 84, 85, 87, 106, 112, 128, 129, 152–157
PACKAGEreledpar	2, 4–8, 10–14, 21–26, 35, 41, 42, 47–49, 51, 53, 54, 84, 98, 128–130, 152
PACKAGEsetspace	3, 23
PACKAGESyntonly	154
PACKAGEXkeyval	24
PACKAGEXR	154

## A

<code>\absline@numR</code>	<a href="#">1</a>
<code>\actionlines@listR</code>	<a href="#">1</a>
<code>\actions@listR</code>	<a href="#">1</a>
<code>\add@apparatusL</code>	<a href="#">1</a>
<code>\add@apparatusR</code>	<a href="#">1</a>
<code>\add@inserts@nextR</code>	<a href="#">1</a>
<code>\add@insertsR</code>	<a href="#">1</a>
<code>\add@penaltiesL</code>	<a href="#">1</a>
<code>\add@penaltiesR</code>	<a href="#">1</a>
<code>\advanceline</code>	<a href="#">1</a>
<code>\affixline@numR</code>	<a href="#">1</a>
<code>\affixpstart@numL</code>	<a href="#">1</a>
<code>\affixpstart@numR</code>	<a href="#">1</a>
<code>\affixside@noteR</code>	<a href="#">1</a>
<code>\aftercolumnseparator</code>	<a href="#">1</a> , <a href="#">10</a>
<code>\annot@listR</code>	<a href="#">1</a>
<code>\araw@textfalse</code>	<a href="#">1</a>
<code>\araw@texttrue</code>	<a href="#">1</a>
<code>astanza (env.)</code>	<a href="#">21</a>
<code>\AtBeginPairs</code>	<a href="#">1</a> , <a href="#">9</a>
<code>\AtEveryPstartCall</code>	<a href="#">1</a>
<code>\autopar</code>	<a href="#">20</a>

## B

<code>\bbl@set@language</code>	<a href="#">1</a>
<code>\beforecolumnseparator</code>	<a href="#">1</a> , <a href="#">10</a>
<code>\beginnumbering</code>	<a href="#">17</a>
<code>\beginnumberingR</code>	<a href="#">1</a>

## C

<code>\c@firstlinenumR</code>	<a href="#">1</a>
<code>\c@firstsublinenumR</code>	<a href="#">1</a>
<code>\c@linenumincrementR</code>	<a href="#">1</a>
<code>\c@sublinenumincrementR</code>	<a href="#">1</a>
<code>\ch@ck@l@ckR</code>	<a href="#">1</a>
<code>\ch@cksub@l@ckR</code>	<a href="#">1</a>
<code>\chapter</code>	<a href="#">1</a>
<code>\chapterinpages</code>	<a href="#">1</a>
<code>\check@goal</code>	<a href="#">1</a>
<code>\check@pstarts</code>	<a href="#">1</a>
<code>\checkpageL</code>	<a href="#">1</a>

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

**D**

<code>\do@actions@fixedcodeR</code>	<u>1</u>
<code>\do@actions@nextR</code>	<u>1</u>
<code>\do@actionsR</code>	<u>1</u>
<code>\do@ballastR</code>	<u>1</u>
<code>\do@insidelineLhook</code>	<u>1</u>
<code>\do@insidelineRhook</code>	<u>1</u>
<code>\do@lineL</code>	<u>1</u>
<code>\do@lineLhook</code>	<u>1</u>
<code>\do@lineR</code>	<u>1</u>
<code>\do@lineRhook</code>	<u>1</u>
<code>\do@lockoff</code>	<u>1</u>
<code>\do@lockoffR</code>	<u>1</u>
<code>\do@lockon</code>	<u>1</u>
<code>\do@lockonR</code>	<u>1</u>
<code>\doinsidelineLhook</code>	<u>1</u> , <u>21</u>
<code>\doinsidelineRhook</code>	<u>1</u> , <u>21</u>
<code>\dolineLhook</code>	<u>1</u> , <u>21</u>
<code>\dolineRhook</code>	<u>1</u> , <u>21</u>
<code>\dump@pstartL@pc</code>	<u>1</u>
<code>\dump@pstartR@pc</code>	<u>1</u>

**E**

<code>\edlabel</code>	<u>1</u>
<code>\edtext</code>	<u>1</u>
<code>\edtext@later</code>	<u>1</u>
<code>\edtext@now</code>	<u>1</u>
<code>\edtextlater</code>	<u>1</u> , <u>16</u>
<code>\edtextnow</code>	<u>1</u> , <u>16</u>
<code>\eled@sectioningR@out</code>	<u>1</u>

<code>\eledsection@correcting@skip</code> .....	<a href="#">1</a>
<code>\eledsectmark</code> .....	<a href="#">1</a> , <a href="#">23</a>
<code>\eledsectnotoc</code> .....	<a href="#">1</a> , <a href="#">23</a>
<code>\endlock</code> .....	<a href="#">1</a>
<code>\endnumbering</code> .....	<a href="#">1</a> , <a href="#">17</a>
<code>\endnumberingR</code> .....	<a href="#">1</a>
<code>\endsub</code> .....	<a href="#">1</a>
environments:	
<code>astanza</code> .....	<a href="#">21</a>
<code>Leftside</code> .....	<a href="#">17</a>
<code>pages</code> .....	<a href="#">12</a>
<code>pairs</code> .....	<a href="#">9</a>
<code>Rightside</code> .....	<a href="#">17</a>

## F

<code>\f@x@l@cksR</code> .....	<a href="#">1</a>
<code>\finish@Pages@notes</code> .....	<a href="#">1</a>
<code>\first@linenum@out@Rfalse</code> .....	<a href="#">1</a>
<code>\first@linenum@out@Rtrue</code> .....	<a href="#">1</a>
<code>\firstlinenum</code> .....	<a href="#">1</a> , <a href="#">18</a>
<code>\firstlinenum*</code> .....	<a href="#">1</a> , <a href="#">18</a>
<code>\firstlinenumR</code> .....	<a href="#">1</a> , <a href="#">18</a>
<code>\firstsublinenum</code> .....	<a href="#">1</a> , <a href="#">18</a>
<code>\firstsublinenum*</code> .....	<a href="#">1</a> , <a href="#">18</a>
<code>\firstsublinenumR</code> .....	<a href="#">1</a> , <a href="#">18</a>
<code>\fix@page</code> .....	<a href="#">1</a>
<code>\flag@end</code> .....	<a href="#">1</a>
<code>\flag@start</code> .....	<a href="#">1</a>
<code>\flush@notesR</code> .....	<a href="#">1</a>
<code>\footnote@reading</code> .....	<a href="#">1</a>
<code>\footnote@typeset</code> .....	<a href="#">1</a>
<code>\footnoteXmk</code> .....	<a href="#">16</a>
<code>\footnoteXnomk</code> .....	<a href="#">16</a>

## G

<code>\get@familiarfootnote@number</code> .....	<a href="#">1</a>
<code>\get@intercolumns@width</code> .....	<a href="#">1</a>
<code>\get@nextboxL</code> .....	<a href="#">1</a>
<code>\get@nextboxR</code> .....	<a href="#">1</a>
<code>\get@sidenote@morespace@columns</code> .....	<a href="#">1</a>
<code>\getline@numR</code> .....	<a href="#">1</a>
<code>\getlinesfrompagelistL</code> .....	<a href="#">1</a>
<code>\getlinesfrompagelistR</code> .....	<a href="#">1</a>
<code>\getlinesfromparlistL</code> .....	<a href="#">1</a>
<code>\getlinesfromparlistR</code> .....	<a href="#">1</a>
<code>\goalfraction</code> .....	<a href="#">1</a>

## H

<code>\hidenumbering</code> .....	<a href="#">18</a>
-----------------------------------	--------------------

## I

<code>\if@getnextbox</code> .....	<a href="#">1</a>
<code>\if@pstarts</code> .....	<a href="#">1</a>
<code>\if@sidenotesmarginpage</code> .....	<a href="#">1</a>
<code>\ifaraw@text</code> .....	<a href="#">1</a>

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

**L**

<code>\l@d@set</code> .....	<u>1</u>
<code>\l@dc@maxchunks</code> .....	<u>1</u>
<code>\l@dcalc@maxoftwo</code> .....	<u>1</u>
<code>\l@dcalc@minoftwo</code> .....	<u>1</u>
<code>\l@dcalcnun</code> .....	<u>1</u>
<code>\l@dchecklang</code> .....	<u>1</u>
<code>\l@dleftbox</code> .....	<u>1</u>
<code>\l@dlinenumR</code> .....	<u>1</u>
<code>\l@dmake@labelsR</code> .....	<u>1</u>
<code>\l@dminpagelines</code> .....	<u>1</u>
<code>\l@dnumpstartsL</code> .....	<u>1</u>
<code>\l@dnumpstartsR</code> .....	<u>1</u>
<code>\l@dpagefullfalse</code> .....	<u>1</u>
<code>\l@dpagefulltrue</code> .....	<u>1</u>
<code>\l@dpscL</code> .....	<u>1</u>
<code>\l@dpscR</code> .....	<u>1</u>
<code>\l@drightbox</code> .....	<u>1</u>
<code>\l@dsamepagefalse</code> .....	<u>1</u>
<code>\l@dsamepagetrue</code> .....	<u>1</u>
<code>\l@dsetupmaxlinecounts</code> .....	<u>1</u>



<code>\l@dsetuprawboxes</code>	<a href="#">1</a>
<code>\l@dskipversenumberR</code>	<a href="#">1</a>
<code>\l@duselanguage</code>	<a href="#">1</a>
<code>\l@dzeromaxlinecounts</code>	<a href="#">1</a>
<code>\labelref@listR</code>	<a href="#">1</a>
<code>\last@page@numR</code>	<a href="#">1</a>
<code>\Lcolwidth</code>	<a href="#">1, 9, 12</a>
<code>\led@err@BadLeftRightPstarts</code>	<a href="#">1</a>
<code>\led@err@Columns@InsideEnv</code>	<a href="#">1</a>
<code>\led@err@Columns@WithoutEnv</code>	<a href="#">1</a>
<code>\led@err@Columns@WithoutLeftside</code>	<a href="#">1</a>
<code>\led@err@Columns@WithoutRightside</code>	<a href="#">1</a>
<code>\led@err@LeftOnRightPage</code>	<a href="#">1</a>
<code>\led@err@Leftside@PreviousNotPrinted</code>	<a href="#">1</a>
<code>\led@err@Pages@InsideEnv</code>	<a href="#">1</a>
<code>\led@err@Pages@WithoutEnv</code>	<a href="#">1</a>
<code>\led@err@Pages@WithoutLeftside</code>	<a href="#">1</a>
<code>\led@err@Pages@WithoutRightside</code>	<a href="#">1</a>
<code>\led@err@polyglossiaTooOld</code>	<a href="#">1</a>
<code>\led@err@RightOnLeftPage</code>	<a href="#">1</a>
<code>\led@err@Rightside@PreviousNotPrinted</code>	<a href="#">1</a>
<code>\led@err@TooManyPstarts</code>	<a href="#">1</a>
<code>\led@error@edtext@later@now</code>	<a href="#">1</a>
<code>\led@error@fail@patch@mempnum</code>	<a href="#">1</a>
<code>\led@error@fail@patch@outputpage</code>	<a href="#">1</a>
<code>\led@error@fail@patch@pagenumbering</code>	<a href="#">1</a>
<code>\led@error@fail@patch@thepage</code>	<a href="#">1</a>
<code>\led@error@missing@numbering</code>	<a href="#">1</a>
<code>\led@error@note@called@onleftside</code>	<a href="#">1</a>
<code>\led@error@note@called@onrightside</code>	<a href="#">1</a>
<code>\led@nopbnumR</code>	<a href="#">1</a>
<code>\led@nopbR</code>	<a href="#">1</a>
<code>\led@pbnnumR</code>	<a href="#">1</a>
<code>\led@pbR</code>	<a href="#">1</a>
<code>\led@warn@ChangeSyncOption</code>	<a href="#">1</a>
<code>\led@warn@setting@in@rightside</code>	<a href="#">1</a>
<code>\lednopbnum</code>	<a href="#">1</a>
<code>\lednopbnumR</code>	<a href="#">1</a>
<code>\ledpbnnumR</code>	<a href="#">1</a>
<code>\ledpbR</code>	<a href="#">1</a>
<code>\ledstrutL</code>	<a href="#">1</a>
<code>\ledstrutR</code>	<a href="#">1</a>
<code>\ledthegoal</code>	<a href="#">1</a>
<code>\leftlinenumannotationR</code>	<a href="#">1</a>
<code>\leftlinenumR</code>	<a href="#">1</a>
<code>\Leftpagehook</code>	<a href="#">1, 21</a>
<code>\leftpstartnumL</code>	<a href="#">1</a>
<code>\leftpstartnumR</code>	<a href="#">1</a>
<code>Leftside (env.)</code>	<a href="#">17</a>
<code>\Leftsidehook</code>	<a href="#">1</a>
<code>\Leftsidehookend</code>	<a href="#">1</a>
<code>\line@list@stuffR</code>	<a href="#">1</a>
<code>\line@listR</code>	<a href="#">1</a>
<code>\line@marginR</code>	<a href="#">1</a>

<code>\line@numR</code> .....	<u>1</u>
<code>\lineation*</code> .....	<u>1</u> , 18
<code>\lineationR</code> .....	<u>1</u> , 18
<code>\linenum@outR</code> .....	<u>1</u>
<code>\linenumberLevenifblanktrue</code> .....	19
<code>\linenumberRevenifblanktrue</code> .....	19
<code>\linenumberstyle*</code> .....	<u>1</u> , 18
<code>\linenumberstyleR</code> .....	<u>1</u> , 18
<code>\linenumincrement</code> .....	<u>1</u> , 18
<code>\linenumincrement*</code> .....	<u>1</u> , 18
<code>\linenumincrementR</code> .....	<u>1</u> , 18
<code>\linenummargin</code> .....	<u>1</u>
<code>\linenummargin*</code> .....	<u>1</u> , 19
<code>\linenummarginColumns</code> .....	<u>1</u> , 19
<code>\linenummarginColumns*</code> .....	<u>1</u> , 19
<code>\linenummarginColumnsR</code> .....	<u>1</u> , 19
<code>\linenummarginR</code> .....	<u>1</u> , 19
<code>\linenumOnlyPagesForColumns</code> .....	<u>1</u> , 19
<code>\linenumOnlyPagesForColumnsR</code> .....	<u>1</u> , 19
<code>\linenumrepR</code> .....	<u>1</u>
<code>\linesinpar@listL</code> .....	<u>1</u>
<code>\linesinpar@listR</code> .....	<u>1</u>
<code>\list@clearing@regR</code> .....	<u>1</u>
<code>\list@pstartL@pc</code> .....	<u>1</u>
<code>\list@pstartR@pc</code> .....	<u>1</u>
<code>\lock@off</code> .....	<u>1</u>

## M

<code>\maxchunks</code> .....	<u>1</u> , 8
<code>\maxlinesinpar@list</code> .....	<u>1</u>
<code>\memorydump</code> .....	17
<code>\memorydumpL</code> .....	<u>1</u>
<code>\memorydumpR</code> .....	<u>1</u>

## N

<code>\n@num</code> .....	<u>1</u>
<code>\namebox</code> .....	<u>1</u>
<code>\new@lineL</code> .....	<u>1</u>
<code>\new@lineR</code> .....	<u>1</u>
<code>\newnamebox</code> .....	<u>1</u>
<code>\newnamecount</code> .....	<u>1</u>
<code>\newseries@par</code> .....	<u>1</u>
<code>\next@line@list@stuffR</code> .....	<u>1</u>
<code>\notesXwidthliketwocolumns</code> .....	10
<code>\num@linesR</code> .....	<u>1</u>
<code>\numberpstartfalse</code> .....	18
<code>\numberpstarttrue</code> .....	18
<code>\numpagelinesL</code> .....	<u>1</u>
<code>\numpagelinesR</code> .....	<u>1</u>

## O

<code>\one@lineR</code> .....	<u>1</u>
<code>\onlysideX</code> .....	15
<code>optionadvancedshiftedpstarts</code> .....	12, 13, 25

optionauxdir	154
optioncontinuousnumberingwithcolumns	10, 154, 156
optionmainmatter	122
optionmovecolumnsonrightpage	11, 155
optionnomaxlines	12, 13, 25
optionnopbinverse	156
optionnosyncpstarts	13, 25, 118
optionprevpgnotnumbered	155
optionprevpgstyle	155
optionsameparallelpagenumber	152, 157
optionshiftedpstarts	7, 13, 25, 149
optionshiftedverses	148, 149
optionsidenotesmarginpage	22, 26, 156
optionwidthliketwocolumns	10, 156

## P

<code>\page@action</code>	<u>1</u>
<code>\page@numR</code>	<u>1</u>
<code>\Pages</code>	<u>1</u> , <u>12</u>
<code>pages</code> (env.)	<u>12</u>
<code>\Pages@mainmatter</code>	<u>1</u>
<code>pairs</code> (env.)	<u>9</u>
<code>\par@lineR</code>	<u>1</u>
<code>\parledgroup@</code>	<u>1</u>
<code>\parledgroup@beforenotes@save</code>	<u>1</u>
<code>\parledgroup@beforenotesL</code>	<u>1</u>
<code>\parledgroup@beforenotesR</code>	<u>1</u>
<code>\parledgroup@correction@notespacing</code>	<u>1</u>
<code>\parledgroup@correction@notespacing@final</code>	<u>1</u>
<code>\parledgroup@correction@notespacing@init</code>	<u>1</u>
<code>\parledgroup@notes@startL</code>	<u>1</u> , <u>1</u>
<code>\parledgroup@notes@startR</code>	<u>1</u> , <u>1</u>
<code>\parledgroup@notespacing@correction</code>	<u>1</u>
<code>\parledgroup@notespacing@set@correction</code>	<u>1</u>
<code>\parledgroupseries@</code>	<u>1</u>
<code>\parledgroupstype@</code>	<u>1</u>
<code>\pausenumbering@page@numR</code>	<u>1</u>
<code>\pausenumberingR</code>	<u>1</u>
<code>\pend</code>	<u>20</u>
<code>\pendL</code>	<u>1</u>
<code>\pendR</code>	<u>1</u>
<code>\prev@nopbR</code>	<u>1</u>
<code>\prev@pbR</code>	<u>1</u>
<code>\prevpgstyle</code>	<u>1</u>
<code>\print@columnseparator</code>	<u>1</u>
<code>\print@eledsectionL</code>	<u>1</u>
<code>\print@eledsectionR</code>	<u>1</u>
<code>\print@leftcolumn</code>	<u>1</u>
<code>\print@lineL</code>	<u>1</u>
<code>\print@lineR</code>	<u>1</u>
<code>\print@notesX@forpages</code>	<u>1</u>
<code>\print@rightcolumn</code>	<u>1</u>
<code>\print@Xnotes@forpages</code>	<u>1</u>
<code>\pstart</code>	<u>20</u>

<code>\pstartL</code> .....	<a href="#">1</a>
<code>\pstartR</code> .....	<a href="#">1</a>

## R

<code>\Rcolwidth</code> .....	<a href="#">1</a> , <a href="#">9</a> , <a href="#">12</a>
<code>\read@linelist</code> .....	<a href="#">1</a>
<code>\reledpar@error</code> .....	<a href="#">1</a>
<code>\reledpar@warning</code> .....	<a href="#">1</a>
<code>\restore@notesX@settings</code> .....	<a href="#">1</a>
<code>\restore@pstartL@pc</code> .....	<a href="#">1</a>
<code>\restore@pstartR@pc</code> .....	<a href="#">1</a>
<code>\restore@Xnotes@settings</code> .....	<a href="#">1</a>
<code>\resumenumberingR</code> .....	<a href="#">1</a>
<code>\rightlinenumannotationR</code> .....	<a href="#">1</a>
<code>\rightlinenumR</code> .....	<a href="#">1</a>
<code>\Rightpagehook</code> .....	<a href="#">1</a> , <a href="#">21</a>
<code>\rightpstartnumL</code> .....	<a href="#">1</a>
<code>\rightpstartnumR</code> .....	<a href="#">1</a>
<code>Rightside (env.)</code> .....	<a href="#">17</a>
<code>\Rightsidehook</code> .....	<a href="#">1</a>
<code>\Rightsidehookend</code> .....	<a href="#">1</a>
<code>\Rlineflag</code> .....	<a href="#">1</a>

## S

<code>\save@familiarfootnote@number</code> .....	<a href="#">1</a>
<code>\save@section@number</code> .....	<a href="#">1</a>
<code>\section@numR</code> .....	<a href="#">1</a>
<code>\selectlanguage</code> .....	<a href="#">1</a>
<code>\set@line</code> .....	<a href="#">1</a>
<code>\set@line@action</code> .....	<a href="#">1</a>
<code>\set@sectcountR</code> .....	<a href="#">1</a>
<code>\setgoalfraction</code> .....	<a href="#">1</a> , <a href="#">14</a>
<code>\sethangingsymbol</code> .....	<a href="#">21</a>
<code>\setline</code> .....	<a href="#">1</a>
<code>\setlinenum</code> .....	<a href="#">1</a>
<code>\setnamebox</code> .....	<a href="#">1</a>
<code>\setnotepositionliketwocolumns@C</code> .....	<a href="#">1</a>
<code>\setnotepositionliketwocolumns@L</code> .....	<a href="#">1</a>
<code>\setnotepositionliketwocolumns@R</code> .....	<a href="#">1</a>
<code>\setpositionliketwocolumns@C</code> .....	<a href="#">1</a>
<code>\setpositionliketwocolumns@L</code> .....	<a href="#">1</a>
<code>\setpositionliketwocolumns@R</code> .....	<a href="#">1</a>
<code>\setRlineflag</code> .....	<a href="#">19</a>
<code>\setwidthliketwocolumns</code> .....	<a href="#">1</a>
<code>\sidenote@marginR</code> .....	<a href="#">1</a>
<code>\sidenotemargin*</code> .....	<a href="#">1</a>
<code>\sidenotemarginR</code> .....	<a href="#">1</a>
<code>\skip@lockoff</code> .....	<a href="#">1</a>
<code>\skipnumbering</code> .....	<a href="#">1</a> , <a href="#">18</a>
<code>\startlock</code> .....	<a href="#">1</a>
<code>\startsub</code> .....	<a href="#">1</a>
<code>\sub@action</code> .....	<a href="#">1</a>
<code>\subline@numR</code> .....	<a href="#">1</a>
<code>\sublinenumberstyle*</code> .....	<a href="#">1</a> , <a href="#">18</a>

<code>\sublinenumberstyleR</code> .....	<a href="#">1</a> , <a href="#">18</a>
<code>\sublinenumincrement</code> .....	<a href="#">1</a> , <a href="#">18</a>
<code>\sublinenumincrement*</code> .....	<a href="#">1</a> , <a href="#">18</a>
<code>\sublinenumincrementR</code> .....	<a href="#">1</a> , <a href="#">18</a>
<code>\sublinenumrepR</code> .....	<a href="#">1</a>

## T

<code>\theledlanguageL</code> .....	<a href="#">1</a>
<code>\theledlanguageR</code> .....	<a href="#">1</a>
<code>\thepstartL</code> .....	<a href="#">18</a>
<code>\thepstartR</code> .....	<a href="#">18</a>
<code>\thestanzaL</code> .....	<a href="#">1</a> , <a href="#">22</a>
<code>\thestanzaR</code> .....	<a href="#">1</a> , <a href="#">22</a>
<code>\this@section@last@page@numR</code> .....	<a href="#">1</a>
<code>\this@section@page@numR</code> .....	<a href="#">1</a>

## U

<code>\unhnamebox</code> .....	<a href="#">1</a>
<code>\unvnamebox</code> .....	<a href="#">1</a>
<code>\usernamecount</code> .....	<a href="#">1</a>

## W

<code>\widthliketwocolumns</code> .....	<a href="#">10</a>
-----------------------------------------	--------------------

## X

<code>\Xendlineflag</code> .....	<a href="#">16</a>
<code>\Xlineflag</code> .....	<a href="#">16</a>
<code>\Xnoteswidthliketwocolumns</code> .....	<a href="#">10</a>
<code>\Xonlyside</code> .....	<a href="#">15</a>

## Change History

v0.1.0	
General: First public release	1
v0.2.0	
General: Added section of babel related code	89
Fix babel problems	1
\Columns: Added \l@dchecklang and \l@duselanguage to \Columns	96
\Pages: Added \l@duselanguage to \Pages	107
v0.3.0	
General: Added \do@lineLhook and \do@lineRhook	68
Added hooks into Leftside environment	56
Reorganize for ledarab	1
\affixline@numR: Changed \affixline@numR to match new eledmac	73
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR	71
\do@lineL: Added \do@lineLhook to \do@lineL	65
Simplified \do@lineL by using macros for some common code	65
\do@lineR: Changed \do@lineR similarly to \do@lineL	68
\flag@end: Removed extraneous spaces from \flag@end	50
\ifledRcol: Moved \ifl@pairing to eledmac	27
\ifpst@rtedR: Moved \ifpst@rtedL to eledmac	31
\l@dlinenumR: Simplified \leftlinenumR and \rightlinenumR by introducing \l@dlinenumR	40
\l@dnumpstartsR: Moved \l@dnumpstartsL to eledmac	92
\ledstrutR: Added \ledtrutL and \ledstrutR	112
\Pages: Added \ledstrutL to \Pages	108
Added \ledstrutR to \Pages	109
\Rightsidehookend: Added \Leftsidehook, \Leftsidehookend, \Rightsidehook and \Rightsidehookend	57
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR	39
v0.3.1	
General: Minor \linenummargin fix	1
\line@marginR: Do not just set \line@marginR in \linenummargin	36
v0.3.2	
General: Improved parallel page balancing	1
\Pages: Added \l@dminpagelines calculation for succeeding page pairs	111
v0.3.3	
General: Compatibilty with Polyglossia	1
v0.4.0	
General: No more ledparpatch. All patches are now in the main file.	1
v0.5.0	
General: Corrections about \section and other titles in numbered sections	1
v0.6.0	
General: Be able to us \chapter in parallel pages.	1
v0.7.0	
General: Add shiftedverses option which makes there is no blank between two parallel verses with inequal length.	1
v0.8.0	
General: Possibility to have a symbol on each hanging of verses, like in the french typography. Redefine the commande \hangingsymbol to define the character.	1
v0.9.0	
General: Possibility to number \pstart.	18
Possibility to number the pstart with the commands \numberpstarttrue.	1
\ifledRcol: Moved \iflledRcol and \ifnumberingR to eledmac	27

v0.9.1	General: The numbering of the pstarts restarts on each <code>\beginnumbering</code> . . . . .	1
v0.9.2	General: Debug: with <code>\Columns</code> , the hanging indentation now runs on the left columns and the hanging symbol is shown only when <code>\stanza</code> is used. . . . .	1
v0.9.3	General: <code>\thepstartL</code> and <code>\thepstartR</code> use now <code>\bfseries</code> and not <code>\bf</code> , which is deprecated and makes conflicts with memoir class. . . . .	1
v0.10.0	General: <code>\edlabel</code> commands on the right side are now correctly indicated. . . . .	1
	<code>\edlabel</code> commands which start a paragraph are now put in the right place. . . . .	1
v0.11.0	General: Change <code>\do@lineL</code> and <code>\do@lineR</code> to allow line numbering by pstart (like in eledmac 0.15). . . . .	65
	Lineation can be by pstart (like in eledmac 0.15). . . . .	35
	New management of hangingsymbol insertion, preventing undesirable insertions. . .	87
	<code>\affixline@numR</code> : Changed <code>\affixline@numR</code> to allow to disable line numbering (like in eledmac 0.15). . . . .	73
	<code>\Columns</code> : Line numbering by pstart. . . . .	97
	<code>\get@nextboxR</code> : Change <code>\get@nextboxL</code> and <code>\get@nextboxR</code> to allow to disable line numbering (like in eledmac 0.15). . . . .	118
	Pstart number can be printed in side . . . . .	119
	<code>\inserthangingsymbolR</code> : Prevent the column separator for hanging verse from shifting	87
v0.12.0	General: New management of hangingsymbol insertion, preventing undesirable insertions. . . . .	87
v1.0.0	General: Compatibility with eledmac. Change name to eledpar. . . . .	1
	Debug in lineation by pstart . . . . .	35
v1.0.1	General: Correction on <code>\numberonlyfirstinline</code> with lineation by pstart or by page. . .	1
v1.1.0	General: The <code>shiftedverses</code> options becomes <code>shiftedpstarts</code> . . . . .	1
	<code>\pstartR</code> : Add <code>\labelpstarttrue</code> (from eledmac). . . . .	58
v1.1.1	<code>\pstartR</code> : Correct <code>\pstartR</code> bug introduced by 1.1. . . . .	58
v1.1.2	<code>\affixside@noteR</code> : Remove spurious space between line number and line content . .	86
v1.2.0	General: Support for <code>\led&lt;section&gt;</code> commands in parallel texts. . . . .	1
v1.2.1	<code>\set@sectcountR</code> : For the right section, the counter is defined only once. . . . .	33
v1.3.0	<code>\edtext</code> : Manage RTL language. . . . .	50
v1.3.2	General: Debug with some classes. . . . .	1
v1.3.3	General: Debugging the left notes of the right column. . . . .	86
v1.3.4	General: Allow use of commands in sidenotes, as introduced by eledmac 1.0. . . . .	86
v1.4.0	General: Added <code>\do@insidelineLhook</code> and <code>\do@insidelineRhook</code> . . . . .	68
v1.4.1	General: Enable the use of <code>stanzaindentsrepetition</code> within <code>astanza</code> environment. . . . .	88

v1.4.3	
General: Corrects a false hanging verse when a verse is exactly the length of a line. . . . .	1
<code>\inserthangingsymbolR</code> : Hanging verse is no longer automatically flush right. . . . .	87
<code>\pendL</code> : Spurious spaces in <code>\pendL</code> . . . . .	62
<code>\pendR</code> : Spurious spaces in <code>\pstartR</code> . . . . .	63
<code>\pstartR</code> : Spurious spaces in <code>\pstartL</code> and <code>\pstartR</code> . . . . .	58
v1.5.0	
General: Add, as in <code>eledmac</code> , features to manage page breaks. . . . .	1
<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. . . . .	38
v1.6.0	
General: Add tool and documentation for parallel ledgroups . . . . .	22
v1.7.0	
General: Add, as in <code>eledmac</code> , features to make crossrefs with <code>pstart</code> numbers. . . . .	1
v1.8.0	
General: <code>\beginnumbering</code> is defined only on <code>eledmac</code> , not on <code>eledpar</code> . . . . .	31
<code>\l@dlsnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in <code>eledmac</code> . . . . .	86
Add <code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code> . . . . .	10
Add <code>\columnspan</code> . . . . .	10
Add, as in <code>eledmac</code> , new system of sectioning commands. . . . .	1
Add, as in <code>eledmac</code> , option to insert something after <code>\pends</code> / verses. . . . .	1
Add, as in <code>eledmac</code> , option to insert something between <code>\pstarts</code> / verse. . . . .	1
Change <code>\do@lineR</code> and <code>\do@lineR</code> to allow new sectioning commands. . . . .	65
Compatibility with <code>musixtex</code> . . . . .	1
Debug <code>eledmac</code> sectioning command after using <code>\resumenumbering</code> . . . . .	1
New sectioning commands, as in <code>eledmac</code> . . . . .	23
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. . . . .	89
<code>\Columns</code> : Modify <code>\Columns</code> to enable to add section's title. . . . .	95
Suppress <code>\l@dchecklang</code> from <code>\Columns</code> . . . . .	96
<code>\l@dchecklang</code> : Suppress <code>\l@dchecklang</code> which did not work and was not logical, because both columns could have the same language but not the main language of the document. . . . .	90
<code>\Pages</code> : Modify <code>\Pages</code> to enable to add section's title. . . . .	105
<code>\pendL</code> : As in <code>eledmac</code> , <code>\pendL</code> can have an optional argument. . . . .	62
<code>\pendR</code> : As in <code>eledmac</code> , <code>\pendR</code> can have an optional argument. . . . .	63
<code>\print@columnseparator</code> : Move some code of <code>\Columns</code> to <code>\print@columnseparator</code> . . . . .	97
<code>\pstartR</code> : As in <code>eledmac</code> , <code>\pendL</code> and <code>\pendR</code> can have an optional argument. . . . .	58
<code>\sidenotemarginR</code> : <code>\sidenotemargin</code> is now directly defined in <code>eledmac</code> to be able to manage <code>eledpar</code> . . . . .	85
Add <code>\sidenotemargin*</code> . . . . .	85
<code>\theledlanguageR</code> : Correct left/right language setting with <code>polyglossia</code> . . . . .	91
v1.8.1	
<code>\do@lineL</code> : Fix a bug with critical notes a the beginning of a page, (maybe added by v1.8.0) (?). . . . .	65
<code>\do@lineR</code> : Fix a bug with critical notes a the beginning of a page, added by v1.8.0 (?). . . . .	68
v1.8.2	
General: Debug <code>\eledxxx</code> with some paper sizes . . . . .	1
Debug left and side note (bugs added by 1.8.0) . . . . .	1
<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> . . . . .	50
<code>\lineation*</code> : Add <code>\lineation*</code> . . . . .	36



\reledpar@error: Errors specific to eledpar send to eledpar handbook	27
v1.8.3	
General: Add \noeledxxx, as in eledmac	1
\doinsidelineRhook: Added \dolineLhook, \dolineRhook, \doinsidelineLhook and \doinsidelineRhook	67
\Pages: Debug blank pages when using optional argument in the last \pend.	105
\pausenumbering@page@numR: Debug \resumenumberingR	34
v1.9.0	
General: Add \AtBeginPairs macro.	9
Compatibility with \Xnoteswidthliketwocolumns and \notesXwidthliketwocolumns	1
\ifwidthliketwocolumns: Added widthliketwocolumns option	26
\theledlanguageR: Debug left/right language switching with polyglossia. Do not write in .aux file when setting left/right lines.	91
v1.9.1	
\ifledRcol: Moved \ifl@dpadding to eledmac	27
v1.10.0	
General: Compatibility with \AtEveryPstart and \AtEveryPend	1
Restore critical notes in \eledsection in parallel columns (this bug was added in 1.8.2).	1
\Pages: Debug wrong pages splitting when no optional argument is used in last \pend (bug was added in v.1.8.3).	105
Debug wrong parallel pages synchronization when an \edtext falls across two pages.	105
v1.10.1	
\next@line@list@stuffR: Revert modification of 1.4.2, which makes bugs with numbering. Leave vertical mode to solve spurious space before minipage.	49
v1.11.0	
General: Compatibility of standard footnotes with some biblatex styles.	1
\edtext: \critext and \edtext are now defined only in eledmac.	50
v1.12.0	
General: Compatibility with Lua <sup>ℒ</sup> TeX RTL languages.	1
\Columns: Add \l@dprintingcolumnstrue	95
\edlabel: \edlabel and \edindex works now with hyperref when using eledpar. \edlabel is now defined only one time for both eledmac and eledpar	84
\Pages: Add \l@dprintingpagestrue	105
\print@eledsectionL: Compatibility with Lua <sup>ℒ</sup> TeX RTL languages.	66
\print@eledsectionR: Compatibility with Lua <sup>ℒ</sup> TeX RTL languages.	69
\print@lineL: Compatibility with Lua <sup>ℒ</sup> TeX RTL languages.	66
v1.12.1	
\print@eledsectionL: Fix bug with Lua <sup>ℒ</sup> TeX RTL \eledsection.	66
v1.13.0	
General: Enable the use of optional argument of & in astanza environment.	88
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
\clearl@drighthouse: Use \newpage instead of \clearpage.	113
\ifledRcol: Remove false boolean settings which are not needed.	27
\Pages: Prevent false overfull hboxes when using \Pages outside of pages environment.	106
When using shiftedpstarts option, a \l@dleftbox with a null height will advance the \pagetotal in any case.	105

v1.13.1		
	<code>\correct@footinsX@box</code> : Call <code>\correct@footinsX@box</code> and	
	<code>\correct@Xfootins@box</code> directly in <code>\print@notesX@forpages</code> and	
	<code>\print@Xnotes@forpages</code> . . . . .	79
	Correct <code>\correct@footinsX@box</code> and <code>\correct@Xfootins@box</code> . . . . .	79
	<code>\Pages</code> : Prevent false empty page after <code>\Pages</code> (bug added in 1.13.0) . . . . .	105
v1.14.0		
	General: Fix a bug with line number position when using <code>\eledsection</code> and similar	
	commands for RTL texts with Lua $\TeX$ . . . . .	1
	The <code>\newifs</code> are not followed by boolean values set to false, because it is the $\TeX$	
	default setting. . . . .	1
v1.15.0		
	General: Add <code>\AtEveryPstartCall</code> . . . . .	1
	Add <code>sameparallelpagelength</code> option. . . . .	14
	Fix vertical spurious space before right <code>\eledchapter</code> (bug added in v1.13.0). . . . .	1
	Prevent vertical space when using <code>\AtEveryPstart</code> or <code>\AtEveryPend</code> with a	
	command which prints nothing . . . . .	1
	<code>\do@actions@nextR</code> : Add actions 1008 and 1009 . . . . .	71
	<code>\inserthangingsymbolR</code> : Prevent more efficiently the column separator from shifting	
	when a verse is hanging . . . . .	87
	<code>\lineationR</code> : As <code>\lineation</code> , <code>\lineationR</code> automatically set the	
	<code>\pstartinfootnote</code> . . . . .	35
	<code>\n@num</code> : <code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code> . . . . .	45
	<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and	
	<code>Eledpar</code> . . . . .	50
v1.16.0		
	General: Error message when calling <code>\Pages</code> inside <code>pages</code> environment and <code>\Columns</code>	
	inside <code>pairs</code> environment. . . . .	1
	Error message when starting a Leftside/a Rightside while the previous one has not	
	been yet typeset. . . . .	1
	Error message when using <code>\beginnumbering... \endnumbering</code> without <code>\pstart</code> . . . . .	1
	Fix a bug with <code>nofamiliar / nocritical</code> option of <code>eledmac</code> . . . . .	1
	New package option <code>sameparallelpagelength</code> to have the same page number for both	
	left and right side. . . . .	1
	<code>\newsseries@par</code> : Fix a bug with <code>\onlysideX</code> . . . . .	51
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>\hidenumbers</code> . . . . .	18
v2.0.0		
	General: <code>\@nl</code> is now defined only in <code>reledmac</code> . . . . .	43
	<code>\ifbypage@</code> and <code>\ifbypstart@R</code> defined in <code>eledmac</code> . . . . .	35
	Fix some bugs with <code>sameparallelpagelength</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>sameparallelpagelength</code> option. . . . .	1
	<code>\@adv</code> : <code>\@adv</code> defined only in <code>reledmac</code> . . . . .	44
	<code>\@lab</code> : <code>\@lab</code> defined only in <code>eledmac</code> . . . . .	85

\@ref@regR: \@ref defined only in reledmac, code specific to right side moved in \ref@regR. . . . .	45
\@set: \@set defined only in reledmac. . . . .	44
\advanceline: \advanceline defined only in reledmac. . . . .	50
\bb1@set@language: Patch \bb1@set@language instead of redefining it . . . . .	90
\do@lockonR: \do@lockon defined only in reledmac. . . . .	44
\endlock: \startlock and \endlock defined only in reledmac. . . . .	50
\endsub: \startsub and \endsub defined only in reledmac. . . . .	50
\fix@page: \fix@page is defined only once in reledmac . . . . .	44
chapterinpages: Deleting the old system of managing parallel chapter, keep only the new one with \patchcmd. . . . .	56
\l@d@set: \l@d@set defined only in reledmac. . . . .	44
\line@marginR: \linenummargin now defined only once time in reledmac. . . . .	36
\page@action: \page@action defined only in reledmac. . . . .	44
\read@linelist: \read@linelist is defined only once time in \reledmac. . . . .	42
\set@line: \set@line defined only in reledmac. . . . .	50
\set@line@action: \set@line@action defined only in reledmac. . . . .	44
\setline: \setline defined only in reledmac. . . . .	50
\setlinenum: \setlinenum defined only in reledmac. . . . .	50
\skip@lockoff: \do@lockoff defined only in reledmac. . . . .	45
\sub@action: \sub@action defined only in reledmac. . . . .	44
\sublinenumincrement*: \firstlinenum, \linenumincrement, \firstsublinenum, \sublinenumincrement are now defined only in reledmac. . . . .	38
v2.1.0	
General: Fix a bug when using \eledsection and related on right pages when page width is short. . . . .	1
Fix a bug when using \pagenumbering with memoir (bug added in v2.0.0). . . . .	1
Fix a bug with \setparledgroupnotespacing with the shiftedpstarts option. . . . .	1
Fix incompatibility between optional argument of \pstart and \numberpstarttrue . . . . .	1
Options to custom empty right page before \Pages. . . . .	1
v2.2.0	
General: astanza environment can take an optional argument, which will be the optional argument of \pstart started by this environment. . . . .	1
New tools to number stanza . . . . .	1
v2.2.1	
General: Fix a bug with optional argument of last left \pend . . . . .	1
v2.3.0	
General: Change some internal codes in order to provide compatibility with L <sup>A</sup> T <sub>E</sub> X release of october 2015 . . . . .	1
Fix a bug with title number in parallel columns . . . . .	1
New line setting command suffixed by R to set only the right side. . . . .	1
\Pages: Fix a bug when calling \Columns after a \Pages (bug added in v1.13.0). . . . .	106
v2.4.0	
General: New way of (not) synchronizing the parallel pages. . . . .	1
Option to switch to \mainmatter when calling \Pages . . . . .	1
\ledstrutR: Deleted \ledstrutL and \ledstrutR . . . . .	112
Fix a bug with dotted letter . . . . .	112
v2.5.0	
General: Disable empty lines as paragraph in astanza. . . . .	1
Fix a bug introduced in v1.15.0 which made hanging indentation in verse not work anymore. . . . .	1
New commands \linenummarginR and \linenummargin* . . . . .	1

v2.5.1	General: Fix spurious space when using optional argument of <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	General: <code>\Xlineflag</code> and <code>\Xendlineflag</code> added . . . . .	1
	<code>\printlinesR</code> deleted . . . . .	1
	Error message when calling <code>\Pages</code> or <code>\Columns</code> without previous pages or pairs environment. . . . .	1
	Fix 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 pages or columns environments. . . . .	1
	Fix compatibility with <code>babel</code> (broken in v2.0.0). . . . .	1
	No error messages about ends of left / right page when using the <code>\syntaxonly</code> command of the <code>syntonly</code> package. . . . .	1
	<code>\l@dmake@labelsR: \@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> . . . . .	84
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	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
	<code>\newseries@par</code> : The $\TeX$ counter <code>\footnotex@reading</code> is defined in <code>reledmac</code> . . . . .	51
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	General: Add new tools to make apparatuses of manuscripts . . . . .	1
	<code>\do@actions@nextR</code> : Add action 1010 . . . . .	71
v2.11.0	General: Compatibility with <code>reledmac</code> 's <code>auxdir</code> option. . . . .	1

\correct@footinsX@box: Clarification in the handbook about the use of \Xonlyside and onlysideX and error message if mesuse. . . . .	79
New implementation of \Xonlyside and \onlysideX hooks, prevent trouble with vertical spacing. . . . .	79
v2.12.0	
General: Add \edtextlater and \edtextnow. . . . .	1, 82
Fix a bug with \AtBeginPairs. . . . .	1
Fix a bug with \lineationR. . . . .	1
v2.13.0	
General: Compatibility with the new features of reledmac 2.15.0 . . . . .	1
v2.13.1	
General: Simplification of the code concerning the sectioning command . . . . .	1
v2.14.0	
General: Add \linenumberlistR, equivalent to \linenumberlist 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 \edtext, referring to the line number where the sidenote is called. . . . .	1
v2.14.1	
General: Fix a bug when using \markboth on \doinsidelineLhook and \doinsidelineRhook. . . . .	1
Fix a bug with vertical space before sectioning command in optional argument of \pstart (bug added in v2.13.1). . . . .	1
v2.14.2	
General: Fix spurious space with \edtextlater. . . . .	1
v2.14.3	
General: Fix a bug in \edtextnow. . . . .	1
v2.14.4	
General: Take into account \linenumberstyle when using \edlineref. . . . .	1
v2.15.0	
General: Add movecolumnpositiononrightpage 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 \msdata when using multiple \beginnumbering... \endnumbering . . . . .	1
Fix a bug with page break when using \columns (bug added in v. 2.14.1). . . . .	1
New macro \sidenotemarginR, fix default value to right. . . . .	1
v2.16.0	
General: Compatibility with \setmsdataposition . . . . .	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 latexmk. . . . .	1
v2.16.3	
General: Fix a bug when setting \aftercolumnseparator. . . . .	1
v2.17.0	
General: Change code for compatibility with reledmac 2.18.0 . . . . .	1
v2.17.1	
General: Fix a bug with prevpgstyle option when not using prevpgnotnumbered. . . . .	1
v2.17.2	
\Pages: Fix a bug when changing \Lcolwidth and \Rcolwidth in pages environment. . . . .	106

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	General: Add <code>\linenumOnlyPagesForColumns</code> and <code>\linenumOnlyPagesForColumnsR</code> to tell to print the line numbers associated with a column only on some specific pages (for example, only on right pages). . . . .	1
	Add <code>sidenotesmarginpage</code> option to make the sidenotes refer to page margins and not column margins. . . . .	1
	Fix a bug with <code>continuousnumberingwithcolumns</code> when using multiple series of continued numbering text. Now, the line number are correctly reset at each <code>\beginnumbering</code> . . . . .	1
	Fix a bug with <code>continuousnumberingwithcolumns</code> and inner / outer notes and line numbering. . . . .	1
	Fix a bug with <code>continuousnumberingwithcolumns</code> when a <code>\beginnumbering</code> is called after another <code>\beginnumbering... \endnumbering</code> . . . . .	1
	Fix a bug with <code>widthliketwocolumns</code> when manually setting <code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code> . . . . .	1
	Fix compatibility between <code>continuousnumberingwithcolumns</code> and <code>\numberpstartfalse</code> . . . . .	1
	Fix incompatibility between <code>\lineation{page}</code> and <code>continuousnumberingwithcolumns</code> option . . . . .	1
	<code>\next@line@list@stuffR</code> : Add <code>\next@line@list@stuffR</code> internal hook. . . . .	49
v2.20.1	General: Fix bug with <code>continuousnumberingwithcolumns</code> when left column and right column have not the same size. . . . .	1
v2.20.2	General: Fix bugs with <code>continuousnumberingwithcolumns</code> when going to a new page	1
v2.21.0	General: Add compatibility with new features of <code>reledmac</code> 2.26.0 . . . . .	1
v2.21.1	General: Fix bug with <code>\linenumberannotation</code> . . . . .	1
v2.21.2	General: Fix bug with <code>\Xpstart</code> and <code>\edtextlater</code> . . . . .	1
v2.21.3	General: Fix bug when using <code>\numberpstarttrue</code> and alternating between manual <code>\pstart</code> and automatic <code>\pstart</code> produced by <code>\autopar</code> . . . . .	1

v2.21.4	General: Fix bug with using <code>\eledsection</code> and related when alternate normal typesetting and parallel typesetting . . . . .	1
v2.21.5	General: Compatibility with <code>reledmac</code> v. 2.26.6 . . . . .	1
v2.21.6	General: Fix page number in footnote and endnotes when using <code>sameparallelpagelength</code> option . . . . .	1
v2.21.7	General: Compatibility with <code>reledmac</code> 2.26.7 . . . . .	1
v2.21.8	General: Fix bug with critical footnotes line number when alternating between <code>\numberlinefalse</code> and <code>\numberlinetrue</code> . . . . .	1
	Fix bug with marginal line number in parallel typesetting when alternating between <code>\numberlinefalse</code> and <code>\numberlinetrue</code> . . . . .	1
	Workaround with a bug of <code>ETEX</code> 2018/04/01 patch level 3 to generate the <code>.sty</code> file from a <code>.ins</code> file . . . . .	1
v2.21.9	General: Fix (little) bug when using <code>calc</code> package. . . . .	1
v2.22.0	General: Compatibility with <code>\doinsidethislinehook</code> macro. . . . .	1
v2.22.1	General: Really reset line number annotation at each line . . . . .	1
v2.22.2	General: Fix bug with <code>\Xonlyside</code> and <code>\onlysideX</code> . . . . .	1
v2.22.3	General: When using <code>\parindentX</code> or <code>\Xparindent</code> and stanza, use the normal indentation in the footnote, not the verse indentation. . . . .	1
v2.22.4	<code>\theledlanguageR</code> : Provide compatibility with the latest version of <code>polyglossia</code> (v1.45). . . . .	91
v2.23.0	General: Add <code>\Leftpagehook</code> and <code>\Rightpagehook</code> . . . . .	1
	Add hanging symbol feature to <code>astanza</code> . . . . .	1
	<code>\theledlanguageR</code> : Restore compatibility with versions of <code>polyglossia</code> prior to v1.45. . . . .	91
v2.23.1	General: Fix bug when using <code>polyglossia</code> v1.45 or later. . . . .	1
v2.24.0	General: Add antilabe feature. . . . .	1
	Add <code>startstanzaindentsrepetition</code> feature. . . . .	1
v2.24.1	General: Fix incompatibility between lineation by <code>\pstart</code> and <code>(a)stanza</code> . . . . .	1
	Makes <code>\numberstantzatrue</code> working with <code>astanza</code> . . . . .	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 <code>\linenumannotationothersidetrue</code> of <code>reledmac</code> . . . . .	1
v2.25.1	General: <code>\ifsublines@R</code> migrated to <code>reledmac</code> . . . . .	1
v2.25.2	General: Fix incompatibility with <code>babel</code> v3.61 . . . . .	1

v2.25.3	General: Fix incompatibility with <code>polyglossia v1.55</code> . . . . .	1
v2.25.4	General: Fix bug with <code>\linenumOnlyPagesForColumnsR</code> and multiple numbered sections . . . . .	1
v2.25.5	General: Fix bug with hyperlink when using <code>\footnoteXnomk</code> . . . . .	1
v2.25.6	General: Make <code>\doinsidethislinehook</code> working on right side . . . . .	1
v2.25.8	General: Fix compatibility with <code>polyglossia 2.4</code> . . . . .	1
v2.25.9	General: Fix bug with <code>\lockon</code> immediately following a (explicit or implicit) <code>\lockoff</code> . . . . .	1
v2.25.10	General: Fix detection of left/right side language with <code>polyglossia</code> when using implicitly the main language of document . . . . .	1