

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

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

## Abstract

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

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

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

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

## Contents

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

---

\*This file (`reledpar.dtx`) has version number v2.14.0, last revised 2016/10/02.

<sup>†</sup>maieul at maieul dot net

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

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

<b>II Preliminaries</b>	<b>22</b>
II.1 Package's meta-data . . . . .	22
II.2 Package's requirement . . . . .	22
II.3 Package's options . . . . .	22
II.4 Package's options . . . . .	23
II.4.1 Synchronization's options . . . . .	23
II.4.2 Other options . . . . .	24
II.5 Determining side and category of parallel processing . . . . .	24
II.6 Text's width . . . . .	25
II.7 Messages . . . . .	25
II.8 Naming macros . . . . .	28
<b>III Sectioning commands</b>	<b>29</b>
<b>IV Line counting</b>	<b>32</b>
IV.1 Setting lineation reset . . . . .	32
IV.2 Setting line number margin . . . . .	34
IV.3 Setting lineation start and step . . . . .	35
IV.4 Setting line flag . . . . .	36
IV.5 Setting line number style . . . . .	37
IV.6 Print marginal line number . . . . .	37
IV.7 Line-number counters and lists . . . . .	38
IV.7.1 Correspond to those in <code>reledmac</code> for regular or left text . . . . .	38
IV.7.2 Specific to <code>reledpar</code> . . . . .	39
IV.8 Reading the line-list file . . . . .	39
IV.9 Commands within the line-list file . . . . .	40
IV.10 Writing to the line-list file . . . . .	46
<b>V Marking text for notes</b>	<b>47</b>
V.1 Specific hooks and commands for notes . . . . .	48
V.1.1 Notes to be printed on one side only . . . . .	48
V.1.2 Tools specific to familiar footnotes . . . . .	48
V.1.3 Get correct footnote number . . . . .	50
V.2 Create hooks . . . . .	51
V.3 Init standards series (A,B,C,D,E,Z) . . . . .	51
V.4 Tools specific to $\LaTeX$ 's classical footnotes . . . . .	51
<b>VI Pstart numbers dumping and restoration</b>	<b>51</b>
<b>VII Parallel environments</b>	<b>52</b>
<b>VIII Paragraph decomposition and reassembly</b>	<b>55</b>
VIII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code> . . . . .	55
VIII.2 Processing one line . . . . .	60
VIII.3 Line and page number computation . . . . .	65
VIII.4 Line number printing . . . . .	68
VIII.5 Pstart number printing in side . . . . .	70

VIII.6 Add insertions to the vertical list . . . . .	72
VIII.7 Penalties . . . . .	72
VIII.8 Printing leftover notes . . . . .	73
<b>IX Footnotes</b>	<b>74</b>
IX.1 Footnotes output specific to \Pages . . . . .	74
IX.2 Critical footnote printed in right side . . . . .	77
<b>X Cross referencing</b>	<b>79</b>
<b>XI Side notes</b>	<b>80</b>
<b>XII Verse</b>	<b>81</b>
<b>XIII Fixing babel and polyglossia</b>	<b>84</b>
<b>XIV Counts and boxes for parallel texts</b>	<b>86</b>
<b>XV Checking text to be processed</b>	<b>87</b>
<b>XVI Parallel columns</b>	<b>89</b>
<b>XVII Parallel pages</b>	<b>97</b>
XVII.1 Specific counters . . . . .	97
XVII.2 Main macro . . . . .	97
XVII.3 Ensure all notes are printed at the end of parallel pages . . . . .	104
XVII.4 Struts . . . . .	105
XVII.5 Page clearing . . . . .	105
XVII.6 Lines managing . . . . .	106
XVII.7 Page break managing . . . . .	107
XVII.8 Getting boxes content . . . . .	111
<b>XVIII Page numbering</b>	<b>115</b>
XVIII.1 Global options . . . . .	115
XVIII.2 mainmatter option of \Pages . . . . .	117
<b>XIX Sections' titles' commands</b>	<b>117</b>
<b>XX Page break/no page break, depending on the specific line</b>	<b>118</b>
<b>XXI Parallel ledgroup</b>	<b>119</b>
<b>XXII Compatibility with eledmac</b>	<b>122</b>
<b>XXIII The End</b>	<b>123</b>

<b>Appendix A Some things to do when changing version</b>	<b>124</b>
Appendix A.1 Migration to <code>eledpar</code> 1.4.3 . . . . .	124
Appendix A.2 Migration from <code>eledpar</code> to <code>reledpar</code> . . . . .	124
Appendix A.2.1 Deprecated options . . . . .	124
Appendix A.2.2 <code>\renewcommandreplaced</code> with <code>command</code> . . . . .	124
Appendix A.2.3 Commands the names of which have changed . . . . .	125
Appendix A.3 Migration to <code>reledpar</code> 2.2.0 . . . . .	125
Appendix A.4 Migration to <code>reledpar</code> 2.3.0 . . . . .	125
Appendix A.5 Migration to <code>reledpar</code> 2.4.0 . . . . .	125
Appendix A.6 Migration to <code>reledpar</code> 2.5.0 . . . . .	125
Appendix A.7 Migration to <code>reledpar</code> 2.6.0 . . . . .	125
Appendix A.8 Migration to <code>reledpar</code> 2.6.1 . . . . .	125
Appendix A.9 Migration to <code>reledpar</code> 2.14.0 . . . . .	126
<b>References</b>	<b>126</b>
<b>Index</b>	<b>126</b>
<b>Change History</b>	<b>146</b>

# 1 Introduction

## 1.1 Aim of this package

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

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

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

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

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

## 1.2 Historical overview

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

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

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

## 2 Options

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

### 2.1 Synchronization's options

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

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

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

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

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

### 2.2 Other options

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

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

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

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

---

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

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

### 3 General

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

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

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

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

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

```
\maxchunks{5120}
```

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

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

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

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

---

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

in this respect.

## 4 Parallel columns

### 4.1 Basic use

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

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

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

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

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

There is no required pagebreak before or after the columns.

### 4.2 Setting

#### 4.2.1 Column's width

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

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

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

#### 4.2.2 Column's separator

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



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

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

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

### 4.2.3 Column's positions

`\columnspan`

By default, columns are positioned to the right of the page. However, you can use `\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`

`\notesXwidthliketwocolumns`

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

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

## 5 Facing pages

### 5.1 Basic usage

`pages`

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

Within the environment the text for the lefthand and righthand pages is placed within the `Leftside` and `Rightside` environments, respectively.

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

```
\begin{pages}
\begin{Leftside} reledmac numbered text commands \end{Leftside}
\begin{Rightside} reledmac numbered text commands \end{Rightside}
\begin{Leftside} reledmac numbered text commands \end{Leftside}
...
\end{pages}
\Pages
```

The `Leftside` text is set on lefthand (even numbered) pages and the `Rightside` text is set on righthand (odd numbered) pages. Each `\Pages` command starts a new even numbered page. After parallel typesetting is finished, a new page is started. Note that the `\Pages` **must be** outside of the `pages` environment.

## 5.2 Setting

### 5.2.1 Text width

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

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

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

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

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

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

The aim of the default setting is twofold:

---

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

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

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

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

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

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

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

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

### 5.2.3 Page number

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

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

Please also read advising in 11 p. 21.

### 5.2.4 Page breaking

`\setgoalfraction`

When doing parallel pages `reledpar` has to guess where  $\TeX$  is going to put pagebreaks and hopefully get there first in order to put the pair of texts on their proper pages. When it thinks that the fraction `\@goalfraction` of a page has been filled, it finishes that page and starts on the other side’s text. The standard value is 0.9.

If you think you can get more on a page, increase this. On the other hand, if some left text overflows onto an odd numbered page or some right text onto an even page, try reducing it. You can change it using `\setgoalfraction{<newvalue>}`.

### 5.2.5 Right page before `\Pages`

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

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

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

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

### 5.2.6 Notes about `\mainmatter`

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

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

### 5.3 Critical and familiar footnotes

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

#### 5.3.1 Notes height setting

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

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

So, you **MUST** change the default setting via `\Xmaxhnotes` (for critical notes) and `\maxhnotesX` (for familiar notes). Both commands are explained in `reledmac` handbook (7.13.6 p. 46). As an advisable setting:

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

#### 5.3.2 About the numbering of familiar footnotes

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

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

#### 5.3.3 Using `perpage` package

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

So do not set:

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

---

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

```
\MakePerPage{footnoteB}
```

But set:

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

### 5.3.4 Notes for one side only

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

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

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

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

`\footnoteXmk`

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

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

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

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

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

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

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

## 5.5 Using line flag

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

# 6 Left and right texts

## 6.1 Environments

Parallel texts are divided into Leftside and Rightside. The form of the contents of these two are independent of whether they will be set in columns or pages.

`Leftside` The left text is put within the Leftside environment and the right text likewise in  
`Rightside` the Rightside environment. The number of Leftside and Rightside environments must be the same.

## 6.2 Numbering text lines and paragraphs

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

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

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

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

`\memorydump`

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

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

`\numberstarttrue`

`\numberstartfalse`

`\thepstartL`

`\thepstartR`

`\skipnumbering`

`\hidenumbering`

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

The command `\skipnumbering` when inserted in a line of parallel text causes the numbering of that particular line to be skipped. This can be useful if you are putting some kind of marker (even if it is only a blank line) between stanzas. Remember, parallel texts must be numbered and this provides a way to slip in an “unnumbered” line. When inserted into a numbered line the macro `\hidenumbering` causes the number for that particular line to be hidden; namely, no line number will print. Note that if you use it in `\stanza`, you must call it at the beginning of the verse.

### 6.3 Lineation system

`\firstlinenum`

`\linenumincrement`

`\firstsublinenum`

`\sublinenumincrement`

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

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



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

<code>\firstlinenum*</code> <code>\linenumincrement*</code> <code>\firstsublinenum*</code> <code>\sublinenumincrement*</code> <code>\firstlinenumR</code> <code>\linenumincrementR</code> <code>\firstsublinenumR</code> <code>\sublinenumincrementR</code> <code>\lineationR</code> <code>\lineation*</code> <code>\linenumberstyleR</code> <code>\sublinenumberstyleR</code> <code>\linenumberstyle*</code> <code>\sublinenumberstyle*</code> <code>\linenummarginR</code> <code>\linenummargin*</code> <code>\linenummarginColumns</code> <code>\linenummarginColumnsR</code> <code>\linenummarginColumns*</code>  <code>\setRlineflag</code>  <code>\linenumberLevenifblanktrue</code> <code>\linenumberRevenifblanktrue</code>	<p>The starred versions change both left and right numbering schemes.</p> <p>The suffixed version change the right side, without regard to the position they are called.</p> <p><code>\lineationR</code> macro is the equivalent of <code>reledmac \lineation</code> macro for the right side.</p> <p><code>\lineation*</code> macro is the equivalent of <code>reledmac \lineation</code> macro for both sides.</p> <p><code>reledmac</code> allows you to define a <code>\linenumberlist</code> to explicitly define (5.3.2 p. 19) 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 <code>\linenumberlistR</code>.</p> <p><code>\linenumberstyleR</code> is the equivalent of <code>reledmac \linenumberstyle</code> for right text. <code>\sublinenumberstyleR</code> is the equivalent of <code>reledmac \sublinenumberstyle</code> for right text. The starred version are for both side.</p> <p><code>\linenummarginR{&lt;margin&gt;}</code> sets the line margin for right side. <code>\linenummargin*{&lt;margin&gt;}</code> sets for both side. <code>&lt;margin&gt;</code> can be, as for <code>reledmac</code>'s <code>\linenummargin</code> one of these values: left, right, inner, outer.</p> <p>Suppose you typeset parallel texts both in parallel pages and in parallel columns. In this case, your setting of line margin in parallel pages 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 you are in parallel pages, 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.</p> <p>In this case, you can use <code>\linenummarginColumns</code>, which overrides the default setting for the left column, <code>\linenummarginColumnsR</code> which overrides the default setting for the right column, and <code>\linenummarginColumns*</code>, which overrides the default for both left and right columns.</p> <p>A “R” is appended to the line numbers of the right texts. This may be useful for parallel columns but for parallel pages it might be more appropriate to redefine it using <code>\setRlineflag{&lt;flag&gt;}</code>. Use <code>\setRlineflag{}</code> to empty it.</p> <p>By default, when a blank line is printed on one side, in order to synchronize with the other side, no line number is printed. However, you can decide to print them for blank lines, also. Use <code>\linenumberLevenifblanktrue</code> to enable it on the left side, and <code>\linenumberRevenifblanktrue</code> to enable it on right side.</p>
--	--

## 6.4 Chunks

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

`\Pages` macros. Several chunks may be specified within a `Leftside` or `Rightside` environment. A multi-chunk text then looks like:

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

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

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

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

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

Consequently:

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

## 6.6 Language setting

If you are using the `babel` package or the `polyglossia` package, with different languages (via, say, `\selectlanguage`) for the left and right texts it is particularly important to select the appropriate language within the `Leftside` and `Rightside` environments. The initial language selected for the right text is the `babel` package's default. Also, it is the *last* language setting in a side that controls the language used in any notes for that side when they get printed. If you are using multilingual notes then it is probably safest to explicitly specify the language(s) for each note rather than relying on the language selection for the side. The right side language is also applied to the right side line numbers.

## 7 Verse

If you are typesetting verses with `reledmac` you can use the `\stanza` construct, and you can also use this in right or left parallel texts. In this case each verse line is a chunk which has two implications. (1) you can unexpectedly exceed the `\maxchunks` limit or

the overall limit on the number of boxes, and (2) left and right verse lines are matched, which may not be desirable if one side requires more print lines for verse lines than the other does.

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

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

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

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

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

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

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

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

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

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

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

`\thestanzaL` `\thestanzaR` When using `\stanzanumtrue` (9.9 p. 50) in parallel typesetting, stanza counter is replaced by `stanzaL` counter in left side and by `stanzaR` counter in right side. Consequently, you can redefine `\thestanzaL` and `\thestanzaR` to change their aspect.

## 8 Side notes

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

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

## 9 Parallel ledgroups

### 9.1 General

You can also make parallel ledgroups (see the documentation of `reledmac` about ledgroups, 10 p. 51). To do it you have:

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

See the following example:

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

### 9.2 Parallel ledgroups and setspace package

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

`\setparledgroupnotespacing{\singlespacing}`

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

## 10 Sectioning commands

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

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

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

## 11 Notes about page number

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

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

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

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

## I Implementation overview

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

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

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

## II Preliminaries

### II.1 Package’s meta-data

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

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledpar}[2016/10/02 v2.14.0 reledmac extension for
  parallel texts]%
4
5 %
```

### II.2 Package’s requirement

Few commands use `\xspace` command.

```

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

### II.3 Package’s options

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

```

8 \RequirePackage{xkeyval}
9 %
```

## II.4 Package's options

### II.4.1 Synchronization's options

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

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

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

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

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

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

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

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

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

```

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

```

### II.4.2 Other options

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

```

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

```

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

```

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

```

`\ifsameparallelpagenumber` Options related to page numbering

`\ifprevpgnotnumbered`

```

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

```

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

```

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

```

```

51 \ProcessOptionsX%
52 %

```

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

As noted above, much of the code is a duplication of the original `reledmac` code to handle the extra box(es) for the right hand side text, and sometimes for the left hand side as well. In order to distinguish we use ‘R’ or ‘L’ in the names of macros for the right and left code. The specifics of ‘L’ and ‘R’ are normally hidden from the user by letting the `Leftside` and `Rightside` environments set things up appropriately.



`\ifl@dpairing` `\ifl@dpairing` is set TRUE if we are processing parallel texts and `\ifl@dpaging` is also set TRUE if we are doing parallel pages. `\ifledRcol` is set TRUE if we are doing the right hand text. They are defined in `reledmac`.

## II.6 Text's width

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

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

## II.7 Messages

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

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

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

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

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

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

```

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

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

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

\led@err@Pages@WithoutRightside97 \newcommand*{\led@err@Pages@WithoutRightside}{%
\led@err@Pages@WithoutLeftside98   \reledpar@error{\string\Pages\space called without previous `Rightside`
environment (or environment without content)}{\@ehc}}%
\led@err@Columns@WithoutRightside99 \newcommand*{\led@err@Pages@WithoutLeftside}{%
100   \reledpar@error{\string\Pages\space called without previous `Leftside`
environment (or environment without content)}{\@ehc}}%
101 \newcommand*{\led@err@Columns@WithoutRightside}{%
102   \reledpar@error{\string\Columns\space called without previous `Rightside`
environment (or environment without content)}{\@ehc}}%
103 \newcommand*{\led@err@Columns@WithoutLeftside}{%
104   \reledpar@error{\string\Columns\space called without previous `Leftside`
environment (or environment without content)}{\@ehc}}%
105 %

\led@error@fail@patch@thepage106 \newcommand{\led@error@fail@patch@thepage}{%
107   \reledpar@error{Fail to patch \string\thepage\space command.}{\@ehc}%
108 }%
109 %

```

```

\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}%
}%
%
```

```

\newcommand{\led@error@edtext@later@now}{%
  \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?)
}%
%
```

```

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

```

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

```

\led@error@missing@numbering\newcommand{\led@error@missing@numbering}[1]{%
144   \reledpar@error{Missing \string\...pstart\string\pend\space inside `#1`
environment}{\@ehc}%
145 }%
146 %

```

## II.8 Naming macros

The  $\text{\LaTeX}$  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`

```

147 \providecommand*\newnamebox}[1]{%
148   \expandafter\newbox\csname #1\endcsname}
149 \providecommand*\setnamebox}[1]{%
150   \expandafter\setbox\csname #1\endcsname}
151 \providecommand*\unhnamebox}[1]{%
152   \expandafter\unhbox\csname #1\endcsname}
153 \providecommand*\unvnamebox}[1]{%
154   \expandafter\unvbox\csname #1\endcsname}
155 \providecommand*\namebox}[1]{%
156   \csname #1\endcsname}
157
158 %

```

`\ifcsboxvoid` The `\ifcsboxvoid` is a complement to the `etoolbox` package. The first argument is a box name, the second is a the  $\langle true \rangle$  value, the third the  $\langle false \rangle$  value. If the first argument refers to a not existing box, or if it refers to a void box, the  $\langle true \rangle$  value is returned. In other case, the  $\langle false \rangle$  is returned.

```

159 \newcommand*\ifcsboxvoid}[1]{%
160   \ifcsname#1\endcsname%
161   \expandafter\ifvoid\csname #1\endcsname%
162   \expandafter\expandafter%
163   \expandafter\@firstoftwo%
164   \else%
165   \expandafter\expandafter%
166   \expandafter\@secondoftwo%
167   \fi%
168   \else%
169   \expandafter\@firstoftwo%
170   \fi%
171 }%
172 %

```

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

`\usenamecount`

```

173 \providecommand*{\newnamecount}[1]{%
174   \expandafter\newcount\csname #1\endcsname}
175 \providecommand*{\usernamecount}[1]{%
176   \csname #1\endcsname}
177
178 %

```

### 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.

```

179 \newcount\section@numR
180 \section@numR=\z@
181 %

```

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

```

182 \pst@rtedLfalse
183 \newif\ifpst@rtedR
184
185 %

```

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

```

186 \newcommand*{\beginnumberingR}{%
187   \ifnumberingR
188     \led@err@NumberingStarted
189     \endnumberingR
190   \fi
191   \global\l@dnumpstartsR \z@
192   \global\pst@rtedRfalse
193   \global\numberingRtrue
194   \global\advance\section@numR \@ne
195   \global\absline@numR \z@
196   \gdef\normal@page@breakR{}
197   \gdef\l@prev@pbR{}
198   \gdef\l@prev@nopbR{}
199   \global\line@numR \z@
200   \global\@lockR \z@
201   \global\sub@lockR \z@
202   \global\sublines@false
203   \global\let\next@page@numR\relax

```

```

204 \global\let\sub@change\relax
205 \global\stopmsdata@inserted@true%
206 \set@continuousnumberingforR%
207 \message{Section \the\section@numR R }%
208 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
209 \l@dend@stuff
210 \setcounter{pstartR}{1}
211 \begingroup
212 \initnumbering@sectcountR
213 \gdef\eled@sectionsR@{ }%
214 \if@noeled@sec\else%
215 \makeatletter\inputIfFileExists{\jobname.eledsec\the\section@numR R
216 }{}{}\makeatother%
217 \immediate\openout\eled@sectioningR@out=\l@auxdir\jobname.eledsec\the\
218 section@numR R\relax%
219 \fi%
220 }
221 %

```

**\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.

```

220 \def\endnumberingR{%
221 \ifnumberingR
222 \global\numberingRfalse
223 \normal@pars
224 \ifnum\l@dnumpstartsR=0%
225 \led@err@NumberingWithoutPstart%
226 \fi%
227 \ifl@dpairing
228 \global\pst@rtedRfalse
229 \else
230 \ifx\insertlines@listR\empty\else
231 \global\noteschanged@true
232 \fi
233 \ifx\line@listR\empty\else
234 \global\noteschanged@true
235 \fi
236 \fi
237 \ifnoteschanged@
238 \led@mess@NotesChanged
239 \fi
240 \else
241 \led@err@NumberingNotStarted
242 \fi
243 \endgroup

```

```

244 \ifnoeled@sec\else%
245 \immediate\closeout\eled@sectioningR\out%
246 \fi%
247 }
248
249 %

```

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

`\save@section@number`

`\set@sectcountR`

```

250 \newcounter{chapterR}
251 \newcounter{sectionR}
252 \newcounter{subsectionR}
253 \newcounter{subsubsectionR}
254
255 \newcount\old@chapter%
256 \newcount\old@section%
257 \newcount\old@subsection%
258 \newcount\old@subsubsection%
259 \newcommand{\save@section@number}{%
260 \ifdefined\c@chapter%
261 \global\old@chapter\value{chapter}%
262 \fi%
263 \global\old@section\value{section}%
264 \global\old@subsection\value{subsection}%
265 \global\old@subsubsection\value{subsubsection}%
266 }%
267 \newcommand{\initnumbering@sectcountR}{%
268 \ifdefined\c@chapter%
269 \setcounter{chapterR}{\old@chapter}%
270 \fi%
271 \setcounter{sectionR}{\old@section}%
272 \setcounter{subsectionR}{\old@subsection}%
273 \setcounter{subsubsectionR}{\old@subsubsection}%
274 \set@sectcountR%
275 }
276 \newcommand{\set@sectcountR}{%
277 \let\c@chapter\c@chapterR%
278 \let\c@section\c@sectionR%
279 \let\c@subsection\c@subsectionR%
280 \let\c@subsubsection\c@subsubsectionR%
281 }%
282 %

```

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

`\resumenumberingR`

```

283 \newcommand*{\pausenumberingR}{%
284 \endnumberingR\global\numberingRtrue}
285 \newcommand*{\resumenumberingR}{%

```

```

286 \ifnumberingR
287   \global\pst@rtedRtrue
288   \global\advance\section@numR \@ne
289   \led@mess@SectionContinued{\the\section@numR R}%
290   \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
291   \l@dend@stuff
292   \begingroup%
293   \initnumbering@sectcountR%
294   \set@continuousnumberingforR%
295 \else
296   \led@err@numberingShouldHaveStarted
297   \endnumberingR
298   \beginnumberingR
299 \fi}
300
301 %

```

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

```

302 \newcommand*{\memorydumpL}{%
303   \endnumbering
304   \numberingtrue
305   \global\pst@rtedLtrue
306   \global\advance\section@num \@ne
307   \led@mess@SectionContinued{\the\section@num}%
308   \line@list@stuff{\jobname.\extensionchars\the\section@num}%
309   \l@dend@stuff}
310
311 \newcommand*{\memorydumpR}{%
312   \endnumberingR
313   \numberingRtrue
314   \global\pst@rtedRtrue
315   \global\advance\section@numR \@ne
316   \led@mess@SectionContinued{\the\section@numR R}%
317   \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
318   \l@dend@stuff}
319
320 %

```

## 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.

```

321 \newcommand*{\lineationR}[1]{%
322   \ifnumbering
323     \led@err@LineationInNumbered
324   \else
325     \def\@tempa{#1}\def\@tempb{page}%
326     \ifx\@tempa\@tempb
327       \global\bypage@Rtrue
328       \global\bypstart@Rfalse
329       \unless\ifnocritical@%
330         \Xpstart[] [false]%
331       \fi%
332     \else
333       \def\@tempb{pstart}%
334       \ifx\@tempa\@tempb
335         \global\bypage@Rfalse
336         \global\bypstart@Rtrue
337         \unless\ifnocritical@%
338           \Xpstart%
339         \fi%
340       \else
341         \def\@tempb{section}
342         \ifx\@tempa\@tempb
343           \global\bypage@Rfalse%
344           \global\bypstart@Rfalse%
345           \unless\ifnocritical@%
346             \Xpstart[] [false]%
347           \fi%
348         \else
349           \led@warn@BadLineation
350         \fi%
351       \fi
352     \fi
353   \fi}}
354 %

```

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

```

355 \newcommand{\set@continuousnumberingforR}{%
356   \ifcontinuousnumberingwithcolumns%
357     \ifnum\line@numR<\line@num%
358       \expandafter\setlinenum\expandafter{\the\line@num}%
359     \fi%
360     \ifnum\last@page@num>\last@page@numR%
361       \global\last@page@numR=\last@page@num%
362     \fi%

```

```

363 \fi%
364 }
365 %

```

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

```

366 \WithSuffix\newcommand\lineation*[1]{%
367   \lineation{#1}%
368   \lineationR{#1}%
369 }%
370 %

```

## IV.2 Setting line number margin

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

**\line@marginR** 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`.

```

371 \newcount\line@marginR
372 %

```

By default put right text numbers at the right.

```

373 \line@marginR=\@ne
374
375 %

```

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

```

376 \newcommand{\linenummarginR}[1]{%
377   \l@getline@margin{#1}%
378   \ifnum\@l@tempcntb>\m@ne%
379     \global\line@marginR=\@l@tempcntb%
380   \fi%
381 }
382 \WithSuffix\newcommand\linenummargin*[1]{%
383   \l@getline@margin{#1}%
384   \ifnum\@l@tempcntb>\m@ne%
385     \global\line@marginR=\@l@tempcntb%
386     \global\line@margin=\@l@tempcntb%
387   \fi%
388 }
389 %

```

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

```

390
391 \newcount\line@margin@columnsR%
392 \line@margin@columnsR=\m@ne%
393
394 \newcommand{\linenummarginColumns}[1]{%
395   \l@getline@margin{#1}%
396   \ifnum\@l@tempcntb>\m@ne%
397     \global\line@margin@columns=\@l@tempcntb%
398   \fi%
399 }%
400
401 \WithSuffix\newcommand\linenummarginColumns*[1]{%
402   \l@getline@margin{#1}%
403   \ifnum\@l@tempcntb>\m@ne%
404     \global\line@margin@columns=\@l@tempcntb%
405     \global\line@margin@columnsR=\@l@tempcntb%
406   \fi%
407 }%
408
409 \newcommand{\linenummarginColumnsR}[1]{%
410   \l@getline@margin{#1}%
411   \ifnum\@l@tempcntb>\m@ne%
412     \global\line@margin@columnsR=\@l@tempcntb%
413   \fi%
414 }%
415
416 %

```

### 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.

```

417 \newcounter{firstlinenumR}
418 \setcounter{firstlinenumR}{5}
419 \newcounter{linenumincrementR}
420 \setcounter{linenumincrementR}{5}
421 %

```

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

```

422 \newcounter{firstsublinenumR}
423 \setcounter{firstsublinenumR}{5}

```

```

424 \newcounter{sublinenumincrementR}
425 \setcounter{sublinenumincrementR}{5}
426
427 %

```

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

`\linenumincrement`

`\firstsublinenum`

`\sublinenumincrement`

`\firstlinenum*`

`\linenumincrement*`

`\firstsublinenum*`

`\sublinenumincrement*`

```

428 \WithSuffix\newcommand\firstlinenum*[1]{%
429 \setcounter{firstlinenumR}{#1}%
430 \setcounter{firstlinenum}{#1}%
431 }
432 \WithSuffix\newcommand\linenumincrement*[1]{%
433 \setcounter{linenumincrementR}{#1}%
434 \setcounter{linenumincrement}{#1}%
435 }
436 \WithSuffix\newcommand\firstsublinenum*[1]{%
437 \setcounter{subfirstlinenumR}{#1}%
438 \setcounter{subfirstlinenum}{#1}%
439 }
440 \WithSuffix\newcommand\sublinenumincrement*[1]{%
441 \setcounter{sublinenumincrementR}{#1}%
442 \setcounter{sublinenumincrement}{#1}%
443 }
444 %

```

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

`\linenumincrementR`

`\firstsublinenumR`

`\sublinenumincrementR`

```

445 \newcommand\firstlinenumR[1]{%
446 \setcounter{firstlinenumR}{#1}%
447 }
448 \newcommand\linenumincrementR[1]{%
449 \setcounter{linenumincrementR}{#1}%
450 }
451 \newcommand\firstsublinenumR[1]{%
452 \setcounter{subfirstlinenumR}{#1}%
453 }
454 \newcommand\sublinenumincrementR[1]{%
455 \setcounter{sublinenumincrementR}{#1}%
456 }
457 %

```

#### IV.4 Setting line flag

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

```

458 \newcommand{\setRlineflag}[1]{%
459 \gdef\@Rlineflag{#1}%
460 }

```

```

461 \setRlineflag{R}
462 %

```

## IV.5 Setting line number style

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

```

463 \newcommand*{\linenumrepR}[1]{\@arabic{#1}}
464 \newcommand*{\sublinenumrepR}[1]{\@arabic{#1}}
465
466 %

```

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

```

467 \newcommand*{\linenumberstyleR}[1]{%
468   \def\linenumrepR##1{\@nameuse{##1}}
469 \newcommand*{\sublinenumberstyleR}[1]{%
470   \def\sublinenumrepR##1{\@nameuse{##1}}
471 %

```

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

```

472 \WithSuffix\newcommand\linenumberstyle*[1]{%
473   \linenumberstyle{#1}%
474   \linenumberstyleR{#1}%
475 }%
476
477 \WithSuffix\newcommand\sublinenumberstyle*[1]{%
478   \sublinenumberstyle{#1}%
479   \sublinenumberstyleR{#1}%
480 }%
481 %
482 %

```

## IV.6 Print marginal line number

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

```

483 \newif\iflinenumberLevenifblank
484 \newif\iflinenumberRevenifblank
485 %

```

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

```

486 \newcommand*{\leftlinenumR}{%
487   \l@dlinenumR
488   \kern\linenumsep}
489 \newcommand*{\rightlinenumR}{%
490   \kern\linenumsep
491   \l@dlinenumR}
492 \newcommand*{\l@dlinenumR}{%
493   \numlabfont\linenumrepR{\line@numR}\@Rlineflag%
494   \ifsublines@
495     \ifnum\subline@num>\z@
496       \unskip\fullstop\sublinenumrepR{\subline@numR}%
497     \fi
498   \fi}
499
500 %

```

## 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.

```

501 \newcount\line@numR
502 \newcount\subline@numR
503 \newcount\absline@numR
504
505 %

```

`\line@listR` `\insertlines@listR` `\actionlines@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.

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

```

506 \list@create{\line@listR}
507 \list@create{\insertlines@listR}
508 \list@create{\actionlines@listR}
509 \list@create{\actions@listR}
510
511 %

```

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

```

512 \newcount\page@numR
513
514 %

```

#### 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.

```

515 \list@create{\linesinpar@listL}
516 \list@create{\linesinpar@listR}
517 \list@create{\maxlinesinpar@list}
518
519 %

```

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

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

```

520 \newcommand{\list@clearing@regR}{%
521   \list@clear{\line@listR}%
522   \list@clear{\insertlines@listR}%
523   \list@clear{\actionlines@listR}%
524   \list@clear{\actions@listR}%
525   \list@clear{\linesinpar@listR}%
526   \list@clear{\linesonpage@listR}
527 }
528 %

```

`\@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.

```

529 \newcommand{\@par@sync@option}[1]{%
530   \IfStrEq{#1}{\@par@this@sync@option}%
531   {%
532     {\ifledRcol%
533       \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
534     }%
535     \else%
536       \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
537     }%
538   }%
539 }%
540 %

```

`\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.

```

541 \newcommand{\@nl@regR}{%
542   \ifx\l@dchset@num\relax \else
543     \advance\absline@numR \@ne
544     \set@line@action
545     \let\l@dchset@num\relax
546     \advance\absline@numR \m@ne
547     \advance\line@numR \m@ne%    % do we need this?
548   \fi
549   \advance\absline@numR \@ne
550   \ifx\next@page@numR\relax \else
551     \page@action
552     \let\next@page@numR\relax
553   \fi
554   \ifx\sub@change\relax \else
555     \ifnum\sub@change>\z@
556       \sublines@true
557     \else
558       \sublines@false
559     \fi
560     \sub@action
561     \let\sub@change\relax
562   \fi
563   \ifcase\@lockR
564   \or
565     \@lockR \tw@
566   \or\or
567     \@lockR \z@
568   \fi
569   \ifcase\sub@lockR
570   \or
571     \sub@lockR \tw@
572   \or\or
573     \sub@lockR \z@
574   \fi
575   \ifsublines@

```



```

576 \ifnum\sub@lockR<\tw@
577 \advance\subline@numR \@ne
578 \fi
579 \else
580 \ifnum\@lockR<\tw@
581 \advance\line@numR \@ne \subline@numR \z@
582 \fi
583 \fi}
584
585
586 %

```

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

```

587 \newcount\last@page@numR
588 \last@page@numR=-10000
589
590 %

```

`\@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` `\do@lockon` adds an entry to the action-code list to turn line number locking on. The current setting of the sub-lineation flag tells us whether this applies to line numbers or sub-line numbers. It is defined in `reledmac`, however the code specific to right side is defined here, in `\do@lockonR`.

```

591 \newcount\@lockR
592 \newcount\sub@lockR
593
594 \newcommand*{\do@lockonR}{%
595 \xright@appenditem{\the\absline@numR}\to\actionlines@listR

```

```

596 \ifsublines@
597 \xright@appenditem{-1005}\to\actions@listR
598 \ifnum\sub@lockR=\z@
599 \sub@lockR \@ne
600 \else
601 \ifnum\sub@lockR=\thr@@
602 \sub@lockR \@ne
603 \fi
604 \fi
605 \else
606 \xright@appenditem{-1003}\to\actions@listR
607 \ifnum\@lockR=\z@
608 \@lockR \@ne
609 \else
610 \ifnum\@lockR=\thr@@
611 \@lockR \@ne
612 \fi
613 \fi
614 \fi}
615
616 %

```

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

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

`\skip@lockoff` `\skip@lockoff`

```

617
618
619 \newcommand{\do@lockoffR}{%
620 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
621 \ifsublines@
622 \xright@appenditem{-1006}\to\actions@listR
623 \ifnum\sub@lockR=\tw@
624 \sub@lockR \thr@@
625 \else
626 \sub@lockR \z@
627 \fi
628 \else
629 \xright@appenditem{-1004}\to\actions@listR
630 \ifnum\@lockR=\tw@
631 \@lockR \thr@@
632 \else
633 \@lockR \z@
634 \fi
635 \fi}
636
637
638 %

```

`\n@num`

`\@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`.

```
639 \newcount\insert@countR
640 %
```

- #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.

```
641 \newcommand*{\@ref@regR}[2]{%
642   \global\advance\@edtext@level by 1%
643   \global\insert@countR=#1\relax
644   \loop\ifnum\insert@countR>\z@
645     \xright@appenditem{\the\absline@numR}\to\insertlines@listR
646     \global\advance\insert@countR \m@ne
647   \repeat
648   %
```

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.

```
649 \begingroup
650   \let\@ref=\dummy@ref
651   \let\@lopR\@gobble
652   \let\page@action=\relax
653   \let\sub@action=\relax
654   \let\set@line@action=\relax
655   \let\@lab=\relax
656   \let\@lemma=\relax
657   \let\@sw\@gobblethree%
658   #2
659   \global\endpage@num=\page@numR
660   \global\endline@num=\line@numR
661   \global\endsubline@num=\subline@numR
662 \endgroup
663 %
```

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

```

664 \xright@appenditem%
665   {\the\page@numR|\the\line@numR|%
666    \ifsublines@ \the\subline@numR \else 0\fi}%
667   \the\endpage@num|\the\endline@num|%
668   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@listR
669 %

```

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

```

670 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@the\
@edtext@level\endcsname}%
671 %

```

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

```

672 \@ref@reg@parseR{#2}%
673 %

```

Decrease edtext level counter.

```

674 \global\advance\@edtext@level by -1%
675 }
676 %

```

```

\@ref@reg@parseR \newcommand{\@ref@reg@parseR}[1]{%
678 %

```

Declare and init boolean for lemma in this level.

```

679 \providebool{lemmacommand@the\@edtext@level}%
680 \boolfalse{lemmacommand@the\@edtext@level}%
681 %

```

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

```

682 #1%
683 % Now, we store the list of \protect\cs{@sw} of this current \protect\cs{
edtext} as an element of
684 % the global list of list of \protect\cs{@sw} for a \protect\cs{edtext}
depth.
685 % \begin{macrocode}
686 \ifnum\@edtext@level>0%
687 \def\create@this@edtext@level{\expandafter\list@create\expandafter{
csname sw@list@edtextR@the\@edtext@level\endcsname}}%
688 \ifcsundef{sw@list@edtextR@the\@edtext@level}{\
create@this@edtext@level}%
689 \letcs{\@tmp}{sw@list@edtextR@the\@edtext@level}%
690 \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}%
691 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
692 \global\cslet{sw@list@edtextR@the\@edtext@level}{\@tmp}%
693 \fi%

```

```
694 }%
695 %
```

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

```
696 \renewcommand*{\@pend}[1]{%
697   \ifbypstart@global\line@num=0\fi%
698   \xright@appenditem{#1}\to\linesinpar@listL}
699 \renewcommand*{\@pendR}[1]{%
700   \ifbypstart@R\global\line@numR=0\fi
701   \xright@appenditem{#1}\to\linesinpar@listR}
702
703 %
```

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

```
704 \newcommand{\@pstart}[3]{%
705   \ifcsdef{minpage@pstart@#1}%
706     {\ifnumgreater{#2}{\csuse{minpage@pstart@#1}}%
707       {\csnumgdef{minpage@pstart@#1}{#2}}%
708       }%
709     }%
710     {\csnumgdef{minpage@pstart@#1}{#2}}
711   \csnumgdef{afterlines@pstart@#1L}{#3}%
712   }%
713
714 \newcommand{\@pstartR}[3]{%
715   \numdef{\@tmp}{#2-1}%Because we have not to know in which page the pstart
716   starts, but in which pair of facing page
717   \ifcsdef{minpage@pstart@#1}%
718     {\ifnumgreater{\@tmp}{\csuse{minpage@pstart@#1}}%
719       {\csnumgdef{minpage@pstart@#1}{\@tmp}}%
720       }%
721     {\csnumgdef{minpage@pstart@#1}{\@tmp}}
722   \csnumgdef{afterlines@pstart@#1R}{#3}%
723   }%
724 %
```

**\@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.

```
725 \renewcommand*{\@lopL}[1]{%
726   \xright@appenditem{#1}\to\linesonpage@listL}
727 \renewcommand*{\@lopR}[1]{%
```

```

728 \xright@appenditem{#1}\to\linesonpage@listR}
729
730 %

```

## 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`.

```

731 \newwrite\linenum@outR
732 %

```

`\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
733 \newif\iffirst@linenum@out@R
734 \first@linenum@out@Rtrue
735 %

```

`\line@list@stuffR` This is the right text version of the `\line@list@stuff{<file>}` macro. It is called by `\beginnumberingR` and performs all the line-list operations needed at the start of a section. Its argument is the name of the line-list file.

```

736 \newcommand*{\line@list@stuffR}[1]{%
737   \read@linelist{#1}%
738   \iffirst@linenum@out@R
739     \immediate\closeout\linenum@outR
740     \global\first@linenum@out@Rfalse
741     \immediate\openout\linenum@outR=\l@auxdir#1%
742     \immediate\write\linenum@outR{\string\line@list@version{\
this@line@list@version}}%
743     \ifl@dpaging%
744       \immediate\write\linenum@outR{\string\@par@sync@option{\
@par@this@sync@option}}%
745     \fi%
746   \else
747     \if@minipage%
748       \leavevmode%
749     \fi%
750     \closeout\linenum@outR%
751     \openout\linenum@outR=\l@auxdir#1%
752   \fi}
753
754 %

```

**\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.

```

755 \newcommand*{\new@lineL}{\%
756   \write\linenum@out{\string\@nl[\the\c@page][\thepage]}}
757 \%

```

**\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.

```

758 \newcommand*{\new@lineR}{\%
759   \write\linenum@outR{\string\@nl[\the\c@page][\thepage]}}
760 \%

```

**\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. They are both defined in `reledmac`.

**\endsub**

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

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

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

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

**\endlock**

**\skipnumbering**

## V Marking text for notes

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

**\critext**

**\edtext**

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

## V.1 Specific hooks and commands for notes

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

```
\newseries@par61 \newcommand{\newseries@par}[1]{%  
762 %
```

### 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.

```
763 \unless\ifnofamiliar@%  
764 \csgdef{onlysideX@#1}{}%  
765 \newnamebox{footins#1@kept}%  
766 \fi%  
767 \unless\ifnocritical@%  
768 \csgdef{Xonlyside@#1}{}%  
769 \newnamebox{#1footins@kept}%  
770 \fi%  
771 %
```

### V.1.2 Tools specific to familiar footnotes

```
772 \unless\ifnofamiliar@%  
773 %
```

**Managing correct number** One problem with using familiar footnotes in parallel typesetting is the fact that the order of reading notes is not the same as the order they are typeset, because  $\text{\LaTeX}$  reads first all the notes on one side, then all the notes on the other side. Then, however,  $\text{\LaTeX}$  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  $\text{\LaTeX}$  counter `footnoteX`. This the only one manipulated by user, and the only one finally printed.
2. A  $\text{\TeX}$  counter `footnoteX@reading`. Its value is incremented when reading the `\footnoteX` command in left or right side environments. It is used to get the correct footnote number from the `.aux` file to be typeset in the main text. This counter is already defined in `reledmac`, as it is also used for hyperlink.



3. A  $\TeX$  counter `footnoteX@typeset`. Its value is increased when inserting footnotes. Its value is used in the files to be used on the next run for the main text.

So here, we only defined the new counter.

```
774 \newcounter{footnote#1@typeset}%
775 %
```

**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.

```
776 \expandafter\list@create\csname footnote#1@mk\endcsname%
777 %
```

Then, declare the `\footnoteXnomk` command.

```
778 \expandafter\newcommand\csname footnote#1nomk\endcsname[1]{%
779 %
```

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

```
780 \toggletrue{nomk@}%
781 \csuse{footnote#1}{##1}%
782 \togglefalse{nomk@}%
783 %
```

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`.

```
784 \letcs{\@tmp}{footnote#1@mk}%
785 \numdef\@tmpa{\csuse{c{footnote#1}}}%
786 \global\xright@appenditem{\@tmpa}\to\@tmp%
787 \global\cslet{footnote#1@mk}{\@tmp}%
788 }%
789 %
```

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

```
790 \expandafter\newcommand\csname footnote#1mk\endcsname{%
791 %
```

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

```
792 \letcs{\@tmp}{footnote#1@mk}%
793 \gl@p\@tmp\to\@tmpa%
794 \global\cslet{footnote#1@mk}{\@tmp}%
795 %
```

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

```
796 \letcs{\old@footnote}{c@footnote#1}%
797 \setcounter{footnote#1}{\@tmpa}%
798 %
```

Define the footnote mark and print it

```
799 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
800 \csuse{@footnotemark#1}%
801 %
```

Restore previous footnote counter and finally add space.

```
802 \setcounter{footnote#1}{\old@footnote}%
803 \xspace%
804 }%
805 %
```

End of tools specific to familiar notes.

```
806 \fi
807 %
```

End of \newseries@par.

```
808 }%
809 %
```

### V.1.3 Get correct footnote number

`\get@familiarfootnote@number`  
`\save@familiarfootnote@number`

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

```
810 \newcommand{\save@familiarfootnote@number}{%
811 \unless\ifnofamiliar@%
812 \def\do##1{\csxdef{saved@footnote##1}{\the\csname c@footnote##1\endcsname}}%
813 \dolistloop{\@series}%
814 \fi%
815 \xdef\saved@footnote{\the\c@footnote}%
816 }
817 \newcommand{\get@familiarfootnote@number}{%
818 \unless\ifnofamiliar@%
819 \def\do##1{\setcounter{footnote##1@typeset}{\csuse{saved@footnote##1}}}%
820 \dolistloop{\@series}%
821 \fi%
822 \setcounter{footnote@typeset}{\saved@footnote}%
823 }
824 %
```

## V.2 Create hooks

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

```

825 \unless\ifnocritical@%
826   \newhookcommand@series{Xonlyside}%
827 \fi%
828 \unless\ifnofamiliar@%
829   \newhookcommand@series{onlysideX}%
830 \fi
831
832
833 %

```

## 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`.

```

834 \newcommand{\init@series@par}{%
835   \def\do##1{\newseries@par{##1}}%
836   \dolistloop{\@series}%
837 }%
838 \init@series@par%
839 %

```

## 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. 48).

```

\footnote@reading 40 \newcount\footnote@reading%
\footnote@typeset 41 \newcounter{footnote@typeset}%
842 %

```

## VI Pstart numbers dumping and restoration

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

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

```

\list@pstartL@pc  \list@create{\list@pstartL@pc}%
\list@pstartR@pc  \list@create{\list@pstartR@pc}%
845 %

```

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

```

\dump@pstartL@pc  \def\dump@pstartL@pc{%
\dump@pstartR@pc  \xright@appenditem{\the\c@pstartL}\to\list@pstartL@pc%
848 \global\cslet{numberpstartL\the\l@dnumpstartsL}{\ifnumberpstart}%
849 }%
850
851 \def\dump@pstartR@pc{%
852 \xright@appenditem{\the\c@pstartR}\to\list@pstartR@pc%
853 \global\cslet{numberpstartR\the\l@dnumpstartsR}{\ifnumberpstart}%
854 }%
855
856 %

```

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

```

\restore@pstartR@pc
857 \def\restore@pstartL@pc{%
858 \ifx\list@pstartL@pc\empty\else%
859 \gl@p\list@pstartL@pc\to\@temp%
860 \global\c@pstartL=\@temp%
861 \fi%
862 }%
863 \def\restore@pstartR@pc{%
864 \ifx\list@pstartR@pc\empty\else%
865 \gl@p\list@pstartR@pc\to\@temp%
866 \global\c@pstartR=\@temp%
867 \fi%
868 }%
869 %

```

## 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.

```

870 \newenvironment{pairs}{%}
871 \l@dpairingtrue
872 \l@dpagingfalse
873 \initnumbering@quote

```

```

874 \save@familiarfootnote@number%
875 \if@ledgroup%
876 \get@familiarfootnote@number%
877 \fi%
878 \save@section@number%
879 \at@begin@pairs%
880 }{%
881 \l@dpairingfalse
882 }
883
884 %

```

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

```

885 \newcommand{\AtBeginPairs}[1]{\gdef\at@begin@pairs{#1}}%
886 \def\at@begin@pairs{}%
887
888 %

```

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.

```

889 \newenvironment{pages}{%
890 \l@dpairingtrue
891 \l@dpagingtrue
892 \initnumbering@quote
893 \save@familiarfootnote@number%
894 \if@ledgroup%
895 \get@familiarfootnote@number%
896 \fi%
897 \save@section@number%
898 \setlength{\Lcolwidth}{\textwidth}%
899 \setlength{\Rcolwidth}{\textwidth}%
900 }{%
901 \l@dpairingfalse
902 \l@dpagingfalse
903 }
904
905 %

```

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

```

906 \newif\ifinstanzaL
907 \newif\ifinstanzaR
908 %

```

**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.

```

909 \newenvironment{Leftside}{%
910   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname\else%
911     \led@err@Leftside@PreviousNotPrinted%
912   \fi%
913   \ledRcolfalse
914   \setcounter{pstartL}{1}
915   \let\pstart\pstartL
916   \let\thepstart\thepstartL
917   \let\pend\pendL
918   \let\memorydump\memorydumpL
919   \Leftsidehook
920   \let\old@startstanza\@startstanza
921   \def\@startstanza[##1]{\global\instanzaLtrue\old@startstanza[##1]}
922 }{
923   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname%
924     \led@error@missing@numbering{Leftside}%
925   \fi%
926   \Leftsidehookend}
927 %

```

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

```

\Leftsidehookend
\Rightsidehook
\Rightsidehookend
928 \newcommand*{\Leftsidehook}{}
929 \newcommand*{\Leftsidehookend}{}
930 \newcommand*{\Rightsidehook}{}
931 \newcommand*{\Rightsidehookend}{}
932
933 %

```

**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.

```

934 \newenvironment{Rightside}{%
935   \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else%
936     \led@err@Rightside@PreviousNotPrinted%
937   \fi%
938   \ledRcoltrue
939   \let\beginnumbering\beginnumberingR
940   \let\endnumbering\endnumberingR
941   \let\pausenumbering\pausenumberingR
942   \let\resumenumbering\resumenumberingR
943   \let\memorydump\memorydumpR
944   \let\thepstart\thepstartR

```

```

945 \let\pstart\pstartR
946 \let\pend\pendR
947 \let\ledpb\ledpbR
948 \let\lednopb\lednopbR
949 \let\lineation\lineationR
950 \Rightsidehook
951 \let\old@startstanza\@startstanza
952 \def\@startstanza[##1]{\global\instanzaRtrue\old@startstanza[##1]}
953 }{
954 \ledRcolfalse
955 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
956 \lederror@missing@numbering{Rightside}%
957 \fi%
958 \Rightsidehookend
959 }
960
961 %

```

## 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** 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. **\par@lineR** \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.

```

962 \newcount\num@linesR
963 \newbox\one@lineR
964 \newcount\par@lineR
965 %

```

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

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

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

```

966 \newcounter{pstartL}
967 \renewcommand{\thepstartL}{\bfseries\@arabic\c@pstartL}. }
968 \newcounter{pstartR}
969 \renewcommand{\thepstartR}{\bfseries\@arabic\c@pstartR}. }
970
971 \newcommandx*{\pstartL}[1][1]{%
972   \ifluatex%
973     \xdef\l@luatextextdir@L{\the\textdir}%
974     \xdef\l@luatexpardir@L{\the\pardir}%
975     \xdef\l@luatexbodydir@L{\the\bodydir}%
976   \fi%
977   \ifnumbering \else%
978     \led@err@PstartNotNumbered%
979     \beginnumbering%
980   \fi%
981   \ifnumberedpar@%
982     \led@err@PstartInPstart%
983     \pend%
984   \fi%
985 }%
986

```

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

```

987 \ifpst@rtedL\else%
988   \list@clear{\inserts@list}%
989   \global\let\next@insert=\empty%
990   \global\pst@rtedLtrue%
991 \fi%
992 \begingroup\everypar{}%
993

```

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.

```

994 \global\advance\l@dnumpstartsL \@one%
995 \ifnum\l@dnumpstartsL>\l@dc@maxchunks%
996   \led@err@TooManyPstarts%
997   \global\l@dnumpstartsL=\l@dc@maxchunks%
998 \fi%
999 \global\setnamebox{\l@dLcolrawbox\the\l@dnumpstartsL}=\vbox\bgroup%
1000

```

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.

```

1001 \l@dzeroopenalties%
1002 \ifautopar\else%
1003   \ifnumberpstart%
1004     \ifsidepstartnum%
1005       \else%
1006         \thepstartL%
1007       \fi%
1008     \fi%
1009   \fi%
1010 \hsize=\Lcolwidth%
1011 \numberedpar@true%
1012 \iflabelpstart\protected@edef\@currentlabel%
1013   {\p@pstartL\thepstartL}\fi%
1014 %

```

Dump the optional arguments

```

1015 \ifstrempy{#1}%
1016   {\csgdef{before@pstartL@the\l@dnumpstartsL}{\at@every@pstart}}%
1017   {\csgdef{before@pstartL@the\l@dnumpstartsL}{\noindent#1}}%
1018   \at@every@pstart@call%
1019 %

```

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

```

1020 \ignorespaces%
1021 %
1022 }
1023 %

```

The same for right side.

```

1024 \newcommandx*{\pstartR}[1][1]{%
1025   \ifluatex%
1026     \xdef\l@luatextextdir@R{\the\textdir}%
1027     \xdef\l@luatexpardir@R{\the\pardir}%
1028     \xdef\l@luatexbodydir@R{\the\bodydir}%
1029   \fi%
1030   \ifnumberingR \else%
1031     \led@err@PstartNotNumbered%
1032     \beginnumberingR%
1033   \fi%
1034   \ifnumberedpar@%
1035     \led@err@PstartInPstart%
1036     \pendR%
1037   \fi%
1038   \ifpst@rtedR\else%
1039     \list@clear{\inserts@listR}%
1040     \global\let\next@insertR=\empty%
1041     \global\pst@rtedRtrue%

```

```

1042 \fi%
1043 \begingroup\everypar{}%
1044 \global\advance\l@dnumpstartsR \@ne%
1045 \ifnum\l@dnumpstartsR>\l@dc@maxchunks%
1046   \led@err@TooManyPstarts%
1047   \global\l@dnumpstartsR=\l@dc@maxchunks%
1048 \fi%
1049 \global\setnamebox{\l@dRcolrawbox\the\l@dnumpstartsR}=\vbox\bgroup%
1050   \l@dzeropenalties%
1051   \ifautopar\else%
1052     \ifnumberpstart%
1053       \ifsidepstartnum\else%
1054         \thepstartR%
1055       \fi%
1056     \fi%
1057   \fi%
1058   \hsize=\Rcolwidth%
1059   \numberedpar@true%
1060   \iflabelpstart\protected@edef\@currentlabel%
1061     {\p@pstartR\thepstartR}\fi%
1062   \ifstrempy{#1}%
1063     {\csgdef{before@pstartR@the\l@dnumpstartsR}{\at@every@pstart}}%
1064     {\csgdef{before@pstartR@the\l@dnumpstartsR}{\noindent#1}}%
1065     \at@every@pstart@call%
1066     \ignorespaces%
1067   }
1068 %

```

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

```

1069 \newcommandx*\pendL}[1][1]{%
1070   \ifnumbering \else%
1071     \led@err@PendNotNumbered%
1072   \fi%
1073   \ifnumberedpar@ \else%
1074     \led@err@PendNoPstart%
1075   \fi%
1076 %

```

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.

```

1077   \endgraf\global\num@lines=\prevgraf\egroup%
1078   \global\par@line=0%
1079 %

```

End the group that was begun in the \pstart.

```

1080   \endgroup%
1081   \ignorespaces%
1082   \dump@pstartL@pc%

```

```

1083 \ifnumberpstart%
1084   \addtocounter{pstartL}{1}%
1085 \fi
1086 \parledgroup@beforenotes@save{L}%
1087 %
    Dump content of the optional argument.
1088 \ifstrempy{#1}%
1089   {\csgdef{after@pendL@the\l@dnumpstartsL}{\at@every@pend}}%
1090   {\csgdef{after@pendL@the\l@dnumpstartsL}{\noindent#1}}%
1091 }
1092 %

```

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

```

1093 \newcommandx*{\pendR}[1][1]{%
1094   \ifnumberingR \else%
1095     \led@err@PendNotNumbered%
1096   \fi%
1097   \ifnumberedpar@ \else%
1098     \led@err@PendNoPstart%
1099   \fi%
1100   \endgraf\global\num@linesR=\prevgraf\egroup%
1101   \global\par@lineR=0%
1102   \endgroup%
1103   \ignorespaces%
1104   \dump@pstartR@pc%
1105   \ifnumberpstart%
1106     \addtocounter{pstartR}{1}%
1107   \fi%
1108   \parledgroup@beforenotes@save{R}%
1109   \ifstrempy{#1}%
1110     {\csgdef{after@pendR@the\l@dnumpstartsR}{\at@every@pend}}%
1111     {\csgdef{after@pendR@the\l@dnumpstartsR}{\noindent#1}}%
1112 }
1113 %
1114 %

```

**\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.

```

1115 \newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
1116 \gdef\at@every@pstart@call{}%
1117 %

```

**\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
1118 \newif\ifprint@last@after@pendL%
1119 \newif\ifprint@last@after@pendR%
1120 %

```

## 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`

```
1121 \newbox\l@dleftbox
1122 \newbox\l@drightbox
1123
1124 %
```

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

```
\countRline
1125 \newcount\countLline
1126 \countLline \z@
1127 \newcount\countRline
1128 \countRline \z@
1129
1130 %
```

`\@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`

```
1131 \newcount\@donereallinesL
1132 \newcount\@donetotallinesL
1133 \newcount\@donereallinesR
1134 \newcount\@donetotallinesR
1135
1136 %
```

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

```
1137 \newcommand*{\do@lineL}{%
1138 \letcs{\ifnumberpstart}{numberpstart@L\the\l@dpscl}%
1139 \advance\countLline \@ne%
1140 \ifvbox\namebox{\l@dLcolrawbox\the\l@dpscl}%
1141 {\vbadness=10000%
1142 \splittopskip=\z@%
1143 \do@lineLhook%
1144 \l@demtyd@ta%
1145 \global\setbox\one@line=\vsplit\namebox{\l@dLcolrawbox\the\l@dpscl}%
1146 to\baselineskip}%
1147 \IfStrEq{\splitfirstmarks\parledgroup@}{\begin}{\
parledgroup@notes@startL}{}%
1148 \unvbox\one@line \global\setbox\one@line=\lastbox%
1149 \@writepageofparL%
```

```

1150 \getline@numL%
1151 \ifnum\@lock>\@ne%
1152   \inserthangingsymboltrue%
1153 \else%
1154   \inserthangingsymbolfalse%
1155 \fi%
1156 \setbox\l@dleftbox%
1157 \hb@xt@ \Lcolwidth{%
1158   \ifl@dhiddenumber%
1159     \global\l@dhiddenumberfalse%
1160     \f@x@l@cks%
1161   \else%
1162     \affixline@num%
1163   \fi%
1164   \xifinlist{\the\l@dpscL}{\eled@sections@@}%
1165   {%
1166     \if@firstlineofpage%
1167       \set@Xtxtbeforenotes%
1168       \global\@firstlineofpagefalse%
1169     \fi%
1170     \insert@msdata%
1171     \add@inserts%
1172     \add@Xgroupbyline%
1173     \affixside@note%
1174   }%
1175   {\print@lineL}%
1176 }%
1177 \add@penaltiesL%
1178 \global\advance\@donereallinesL\@ne%
1179 \global\advance\@donetotallinesL\@ne%
1180 \else%
1181   \iflinenumberLevenifblank
1182     \new@lineL%
1183     \l@emptyd@ta%
1184     \getline@numL%
1185     \affixline@num%
1186     \setbox\l@dleftbox \hb@xt@ \Lcolwidth{%
1187       \l@dld@ta%
1188       \hspace*{\Lcolwidth}%
1189       \ledrlfill\l@drd@ta%
1190     }%
1191   \else%
1192     \setbox\l@dleftbox \hb@xt@ \Lcolwidth{\hspace*{\Lcolwidth}}%
1193   \fi%
1194   \global\advance\@donetotallinesL\@ne%
1195 \fi%
1196 }%
1197
1198
1199 %

```

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

```

1200 \def\print@lineL{%
1201   \affixpstart@numL%
1202   \l@dld@ta%
1203   \if@firstlineofpage%
1204     \set@Xtxtbeforenotes%
1205     \global\@firstlineofpagefalse%
1206   \fi%
1207   \insert@msdata%
1208   \add@inserts%
1209   \add@Xgroupbyline%
1210   \affixside@note%
1211   \l@dlsn@te%
1212   \hb@xt@ \Lcolwidth{\ledllfill\hb@xt@ \wd\one@line{%
1213     \do@insidelineLhook%
1214     \ifluatex%
1215       \textdir\l@luatextextdir@L%
1216     \fi%
1217     \new@lineL%
1218     \inserthangingsymbolL%
1219     \l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta%
1220   \l@drsn@te}}%
1221
1222 %

```

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

```

1223 \def\print@eledsectionL{%%
1224   \addtocounter{pstartL}{-1}%
1225   \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{%
1226     \ifdefstring{\@eledsectmark}{L}{\ledsectnomark}%
1227     \numdef{\temp@}{\l@dpscL-1}%
1228     \@eled@sectioningtrue%
1229   \bgroup%
1230     \ifluatex%
1231       \textdir\l@luatextextdir@L%
1232       \pardir\l@luatexpardir@L%
1233       \bodydir\l@luatexbodydir@L%
1234       \ifdefstring{\l@luatextextdir@L}{TRT}{\@RTLtrue}{%
1235     \fi%
1236     \csuse{eled@sectioning@the\l@dpscL}%
1237   \egroup%
1238   \@eled@sectioningfalse%
1239   \global\csundef{eled@sectioning@the\l@dpscL}%
1240   \if@RTL%
1241     \hspace{-3\paperwidth}%
1242   {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1243   \else%

```

```

1244 \hspace{3\paperwidth}%
1245 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1246 \fi%
1247 \vskip\eledsection@correcting@skip%
1248 }
1249
1250 %

```

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

```

\doinssidelineLhook \newcommand*{\dolineLhook}[1]{\gdef\dolineLhook{#1}}%
\doinssidelineRhook \newcommand*{\dolineRhook}[1]{\gdef\dolineRhook{#1}}%
1251 \newcommand*{\doinssidelineLhook}[1]{\gdef\doinssidelineLhook{#1}}%
1252 \newcommand*{\doinssidelineRhook}[1]{\gdef\doinssidelineRhook{#1}}%
1253
1254 %
1255
1256 %

```

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

```

\dolineRhook \newcommand*{\do@lineLhook}{}
\doinssidelineLhook \newcommand*{\do@lineRhook}{}
\doinssidelineRhook \newcommand*{\do@insidelineLhook}{}
1257 \newcommand*{\do@insidelineRhook}{}
1258
1259 %
1260
1261
1262 %

```

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

```

1263 \newcommand*{\do@lineR}{%
1264 \let\linenumrepL\linenumrep%
1265 \let\sublinenumrepL\sublinenumrep%
1266 \let\linenumrepR\linenumrepR%
1267 \let\sublinenumrepR\sublinenumrepR%
1268 \letcs{ifnumberpstart}{numberpstart@R\the\l@dpscr}%
1269 \ledRcol@true%
1270 \advance\countRline \@ne%
1271 \ifvbox\namebox{\l@dRcolrawbox\the\l@dpscr}%
1272 {\vbadness=10000%
1273 \splittopskip=\z@%
1274 \do@lineRhook%
1275 \l@demtyd@ta%
1276 \global\setbox\one@lineR=\vsplit\namebox{\l@dRcolrawbox\the\l@dpscr}%
1277 to\baselineskip}%
1278 \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
\parledgroup@notes@startR}{}%
1279 \unvbox\one@lineR \global\setbox\one@lineR=\lastbox%
1280 \writepageofparR%
1281 \getline@numR%

```

```

1282 \ifnum\@lockR>\@ne%
1283   \inserthangingsymbolRtrue%
1284 \else%
1285   \inserthangingsymbolRfalse%
1286 \fi%
1287 \setbox\l@drightbox%
1288 \hb@xt@ \Rcolwidth{%
1289   \ifl@dhiddenumber%
1290     \global\l@dhiddenumberfalse%
1291     \f@x@l@cksR%
1292   \else%
1293     \affixline@numR%
1294   \fi%
1295 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
1296   {%
1297     \if@firstlineofpageR%
1298       \set@Xtxtbeforenotes%
1299       \global\@firstlineofpageRfalse%
1300     \fi%
1301     \insert@msdata%
1302     \add@insertsR%
1303     \add@Xgroupbyline%
1304     \affixside@noter%
1305   }%
1306   {\print@lineR}%
1307 }%
1308 \add@penaltiesR%
1309 \global\advance\@donereallinesR\@ne%
1310 \global\advance\@donetotallinesR\@ne%
1311 \else%
1312 \iflinenumberRevenifblank%
1313   \new@lineR
1314   \l@emptyd@ta%
1315   \getline@numR%
1316   \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1317     \affixline@numR%
1318     \l@dld@ta%
1319     \hspace*{\Rcolwidth}%
1320     \ledrlfill\l@drd@ta%
1321   }%
1322 \else%
1323   \setbox\l@drightbox \hb@xt@ \Rcolwidth{\hspace*{\Rcolwidth}}%
1324 \fi%
1325 \global\advance\@donetotallinesR\@ne%
1326 \fi%
1327 \ledRcol@false%
1328 \let\linenumrep\linenumrepL%
1329 \let\sublinenumrep\sublinenumrepL%
1330 }
1331

```



```
1332
1333 %
```

```
\print@lineR
\print@eledsectionR
```

### VIII.3 Line and page number computation

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

```
1334 \newcommand*{\getline@numR}{%
1335   \global\advance\absline@numR \@ne
1336   \do@actionsR
1337   \do@ballastR
1338   \ifledgroupnotesR@else
1339     \ifnumberline
1340       \ifsublines@
1341         \ifnum\sub@lockR<\tw@
1342           \global\advance\subline@numR \@ne
1343         \fi
1344       \else
1345         \ifnum\@lockR<\tw@
1346           \global\advance\line@numR \@ne
1347           \global\subline@numR \z@
1348         \fi
1349       \fi
1350     \fi
1351   \fi
1352 }
1353 \newcommand*{\getline@numL}{%
1354   \global\advance\absline@num \@ne
1355   \do@actions
1356   \do@ballast
1357   \ifledgroupnotesL@else
1358     \ifnumberline
1359       \ifsublines@
1360         \ifnum\sub@lock<\tw@
1361           \global\advance\subline@num \@ne
1362         \fi
1363       \else
1364         \ifnum\@lock<\tw@
1365           \global\advance\line@num \@ne
1366           \global\subline@num \z@
1367         \fi
1368       \fi
1369     \fi
1370   \fi
1371 }
1372
1373
```

```
1374 %
```

`\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.

```
1375 \newcommand*{\do@ballastR}{\global\ballast@count=\z@
1376   \begingroup
1377     \advance\absline@numR \@ne
1378     \ifnum\next@actionlineR=\absline@numR
1379       \ifnum\next@actionR>-1001
1380         \global\advance\ballast@count by -\c@ballast
1381       \fi
1382     \fi
1383   \endgroup}
1384 %
```

`\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  
`\do@actions@nextR` this.

```
1385
1386 \newif\ifl@dskipversenumberR
1387 \newcommand*{\do@actions@fixedcodeR}{%
1388   \ifcase\@l@dttempcnta%
1389     \or% % 1001 = starting sublineation
1390       \global\sublines@true
1391     \or% % 1002 = ending sublineation
1392       \global\sublines@false
1393     \or% % 1003 = starting locking number
1394       \global\@lockR=\@ne
1395     \or% % 1004 = ending locking number
1396       \ifnum\@lockR=\tw@
1397         \global\@lockR=\thr@@
1398       \else
1399         \global\@lockR=\z@
1400       \fi
1401     \or% % 1005 = starting locking subnumber
1402       \global\sub@lockR=\@ne
1403     \or% % 1006 = ending locking subnumber
1404       \ifnum\sub@lockR=\tw@
1405         \global\sub@lockR=\thr@@
1406       \else
1407         \global\sub@lockR=\z@
1408       \fi
1409     \or% % 1007 = skipping numbering
1410       \l@dskipnumbertrue
1411     \or% % 1008 = skipping numbering in stanza
1412       \l@dskipversenumberRtrue%
1413     \or% % 1009 = hiding number
```

```

1414 \l@dhidenumbertrue%
1415 \or% % 1010 = inserting msdata
1416 \add@msdata%
1417 \else%
1418 \led@warn@BadAction
1419 \fi%
1420 }
1421
1422
1423 \newcommand*{\do@actionsR}{%
1424 \global\let\do@actions@nextR=\relax
1425 \@l@dttempcntb=\absline@numR
1426 \ifnum\@l@dttempcntb<\next@actionlineR\else
1427 \ifnum\next@actionR>-1001\relax
1428 \@firstlineofpageRtrue%
1429 \global\page@numR=\next@actionR
1430 \ifbypage@R
1431 \global\line@numR \z@ \global\subline@numR \z@
1432 \fi
1433 \add@msdata@firstlineofpage%
1434 \else
1435 \ifnum\next@actionR<-4999\relax % 9/05 added relax here
1436 \@l@dttempcnta=-\next@actionR
1437 \advance\@l@dttempcnta by -5001\relax
1438 \ifsublines@
1439 \global\subline@numR=\@l@dttempcnta
1440 \else
1441 \global\line@numR=\@l@dttempcnta
1442 \fi
1443 \else
1444 \@l@dttempcnta=-\next@actionR
1445 \advance\@l@dttempcnta by -1000\relax
1446 \do@actions@fixedcodeR
1447 \fi
1448 \fi
1449 \ifx\actionlines@listR\empty
1450 \gdef\next@actionlineR{1000000}%
1451 \else
1452 \gl@p\actionlines@listR\to\next@actionlineR
1453 \gl@p\actions@listR\to\next@actionR
1454 \global\let\do@actions@nextR=\do@actionsR
1455 \fi
1456 \fi
1457 \do@actions@nextR}
1458
1459 %

```

### VIII.4 Line number printing

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

```

\ch@cksub@l@ckR
\ch@ck@l@ckR
\fx@l@ckR
\affixline@numR
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505

\newcommand*{\l@dcalcnum}[3]{%
  \ifnum #1 > #2\relax
    \@l@tempcnta = #1\relax
    \advance\@l@tempcnta by -#2\relax
    \divide\@l@tempcnta by #3\relax
    \multiply\@l@tempcnta by #3\relax
    \advance\@l@tempcnta by #2\relax
  \else
    \@l@tempcnta=#2\relax
  \fi}

\newcommand*{\ch@cksub@l@ckR}{%
  \ifcase\sub@lockR
  \or
    \ifnum\sublock@disp=\@ne
      \@l@tempcntb \z@ \@l@tempcnta \@ne
    \fi
  \or
    \ifnum\sublock@disp=\tw@
    \else
      \@l@tempcntb \z@ \@l@tempcnta \@ne
    \fi
  \or
    \ifnum\sublock@disp=\z@
      \@l@tempcntb \z@ \@l@tempcnta \@ne
    \fi
  \fi}

\newcommand*{\ch@ck@l@ckR}{%
  \ifcase\@lockR
  \or
    \ifnum\lock@disp=\@ne
      \@l@tempcntb \z@ \@l@tempcnta \@ne
    \fi
  \or
    \ifnum\lock@disp=\tw@
    \else
      \@l@tempcntb \z@ \@l@tempcnta \@ne
    \fi
  \or
    \ifnum\lock@disp=\z@
      \@l@tempcntb \z@ \@l@tempcnta \@ne
    \fi
  \fi}

```

```

1506 \newcommand*{\f@x@l@cksR}{%
1507   \ifcase\@lockR
1508   \or
1509     \global\@lockR \tw@
1510   \or \or
1511     \global\@lockR \z@
1512   \fi
1513   \ifcase\sub@lockR
1514   \or
1515     \global\sub@lockR \tw@
1516   \or \or
1517     \global\sub@lockR \z@
1518   \fi}
1519
1520
1521 \let\linenumberlistR\empty%
1522 \newcommand*{\affixline@numR}{%
1523   \ifledgroupnotesR\else\ifnumberline
1524   \ifl@dskipnumber
1525     \global\l@dskipnumberfalse
1526   \else
1527     \ifsublines@
1528       \@l@tempcntb=\subline@numR
1529       \l@dcalcnm{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR
}%
1530     \ch@cksub@lockR
1531   \else
1532     \@l@tempcntb=\line@numR
1533     \ifx\linenumberlistR\empty%
1534       \l@dcalcnm{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%
1535     \else
1536       \@l@tempcnta=\line@numR
1537       \edef\rem@inderR{\linenumberlistR,\number\line@numR,}%
1538       \edef\sc@n@list{\def\noexpand\sc@n@list
1539         ###1,\number\@l@tempcnta,###2|\def\noexpand\rem@inderR{###2}}
%
1540       \sc@n@list\expandafter\sc@n@list\rem@inderR|
1541       \ifx\rem@inderR\empty\advance\@l@tempcnta\@ne\fi%
1542     \fi
1543     \ch@ck@l@ckR
1544   \fi
1545   \ifnum\@l@tempcnta=\@l@tempcntb
1546   \ifl@dskipversenumberR\else
1547     \if@twocolumn
1548       \if@firstcolumn
1549         \gdef\l@dld@ta{\llap{\leftlinenumR}}%
1550       \else
1551         \gdef\l@drd@ta{\rlap{\rightlinenumR}}%
1552       \fi
1553     \else

```

```

1554 \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columnsR}{\m@ne}}}%
1555   {\@l@tempcntb=\line@margin@columnsR}%
1556   {\@l@tempcntb=\line@marginR}%
1557   \ifnum\@l@tempcntb>\@ne
1558     \advance\@l@tempcntb by\page@numR
1559   \fi
1560   \ifodd\@l@tempcntb
1561     \gdef\l@drd@ta{\rlap{{\rightlinenumR}}}%
1562   \else
1563     \gdef\l@dld@ta{\llap{{\leftlinenumR}}}%
1564   \fi
1565 \fi
1566 \fi
1567 \fi
1568 \f@x@l@cksR
1569 \fi
1570 \fi
1571 \fi}
1572 %

```

### 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@numL73
\affixpstart@numR74 \newcommand*{\affixpstart@numL}{%
\leftpstartnum75 \ifsidepstartnum
\rightpstartnumR76 \if@twocolumn
\leftpstartnumL77 \if@firstcolumn
\rightpstartnumL78 \gdef\l@dld@ta{\llap{{\leftpstartnumL}}}%
\ifpstartnumR79 \else
1580 \gdef\l@drd@ta{\rlap{{\rightpstartnumL}}}%
1581 \fi
1582 \else
1583 \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columns}{\m@ne}}}%
1584   {\@l@tempcntb=\line@margin@columns}%
1585   {\@l@tempcntb=\line@margin}%
1586   \ifnum\@l@tempcntb>\@ne
1587     \advance\@l@tempcntb \page@num
1588   \fi
1589   \ifodd\@l@tempcntb

```

```

1590     \gdef\l@drd@ta{\rlap{\rightpstartnumL}}}%
1591     \else
1592     \gdef\l@dld@ta{\llap{\leftpstartnumL}}}%
1593     \fi
1594 \fi
1595 \fi
1596 }
1597 \newcommand*{\affixpstart@numR}{%
1598 \ifsidepstartnum
1599 \if@twocolumn
1600     \if@firstcolumn
1601     \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
1602     \else
1603     \gdef\l@drd@ta{\rlap{\rightpstartnumR}}}%
1604     \fi
1605     \else
1606     \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\line@margin@columnsR}{\m@ne}}}%
1607     {\@l@tempcntb=\line@margin@columnsR}%
1608     {\@l@tempcntb=\line@marginR}%
1609     \ifnum\@l@tempcntb>\@ne
1610     \advance\@l@tempcntb \page@numR
1611     \fi
1612     \ifodd\@l@tempcntb
1613     \gdef\l@drd@ta{\rlap{\rightpstartnumR}}}%
1614     \else
1615     \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
1616     \fi
1617 \fi
1618 \fi
1619 }
1620
1621 \newcommand*{\leftpstartnumL}{
1622 \ifpstartnum
1623 \thepstartL
1624 \kern\linenumsep\global\pstartnumfalse\fi
1625 }
1626 \newcommand*{\rightpstartnumL}{
1627 \ifpstartnum\kern\linenumsep
1628 \thepstartL
1629 \global\pstartnumfalse\fi
1630 }
1631 \newif\ifpstartnumR
1632 \pstartnumRtrue
1633 \newcommand*{\leftpstartnumR}{
1634 \ifpstartnumR
1635 \thepstartR
1636 \kern\linenumsep\global\pstartnumRfalse\fi
1637 }
1638 \newcommand*{\rightpstartnumR}{

```

```

1639 \ifpstartnumR\kern\linenumsep
1640 \thepstartR
1641 \global\pstartnumRfalse\fi
1642 }
1643 %

```

## 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.

```

1644 \list@create{\inserts@listR}
1645 %

```

`\add@insertsR` The right text version.

```

\add@inserts@nextR
1646 \newcommand*{\add@insertsR}{%
1647   \global\let\add@inserts@nextR=\relax
1648   \ifx\inserts@listR\empty \else
1649     \ifx\next@insertR\empty
1650       \ifx\insertlines@listR\empty
1651         \global\noteschanged@true
1652         \gdef\next@insertR{100000}%
1653       \else
1654         \gl@p\insertlines@listR\to\next@insertR
1655       \fi
1656     \fi
1657     \ifnum\next@insertR=\absline@numR
1658       \gl@p\inserts@listR\to\@insertR
1659       \@insertR
1660       \global\let\@insertR=\undefined
1661       \global\let\next@insertR=\empty
1662       \global\let\add@inserts@nextR=\add@insertsR
1663     \fi
1664   \fi
1665   \add@inserts@nextR}
1666 %
1667 %

```

## VIII.7 Penalties

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

In the code below, which is a virtual copy of the original `\add@penalties`, `\num@lines` is the number of lines in the whole paragraph, and `\par@line` is the line



we are working on at the moment. The count `\@l@dttempcnta` is used to calculate and accumulate the penalty; it is initially set to the value of `\ballast@count`, which has been worked out in `\do@ballast`. Finally, the penalty is checked to see that it does not go below  $-10000$ .

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

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

```
1668 \newcommand*{\add@penaltiesL}{%
1669 \newcommand*{\add@penaltiesR}{%
1670
1671 %
```

## 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.

```
1672 \newcommand*{\flush@notesR}{%
1673 \xloop
1674 \ifx\inserts@listR\empty \else
1675 \glp\inserts@listR\to\@insertR
```

```

1676 \insertR
1677 \global\let\insertR=\undefined
1678 \repeat}
1679
1680 %

```

## IX Footnotes

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

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

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

```

1681 \newcommand\print@Xnotes@forpages[1]{%
1682 %

```

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

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

```

1683 \ifcempty{Xonlyside@#1}{%
1684   \csuse{#1footstart}{#1}%
1685   \csuse{#1footgroup}{#1}%
1686 }%
1687 %

```

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

```

1688 {%
1689   \ifboolexpr{%
1690     ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page
1691   }})%
1692     or%
1693     (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}}))%
1694   }%
1695 %

```

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

```

1695 {%
1696   \csuse{#1footstart}{#1}%
1697   \csuse{#1footgroup}{#1}%
1698 %

```

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.

```

1699 \global\count\csuse{#1footins}=\z@%
1700 \global\skip\csuse{#1footins}=\z@%
1701 \global\dimen\csuse{#1footins}=\maxdimen%
1702 }%
1703 %

```

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

```

1704 {%
1705   \global\count\csuse{#1footins}=\csuse{default@#1footins}%
1706   \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
1707   \bgroup%
1708     \csuse{Xnotefontsize@#1}%
1709     \global\dimen\csuse{#1footins}=\csuse{Xmaxhnotes@#1}%
1710   \egroup%
1711 %

```

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

```

1712 \global\setnamebox{#1footins@kept}=\box\namebox{#1footins}%
1713 }%
1714 %

```

End of \print@Xnotes@forpages.

```

1715 }%
1716 }%
1717 %

```

And now, the same for familiar footnotes.

```

1718 \newcommand\print@notesX@forpages[1]{%
1719   \ifcempty{onlysideX@#1}{%
1720     \csuse{footstart#1}{#1}%
1721     \csuse{footgroup#1}{#1}%
1722   }%
1723   {%
1724     \ifboolexpr{%
1725       ((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page
1726       }%
1727       (test {\ifcsstring{onlysideX@#1}{R}} and test{\ifnumodd{\c@page}}))%
1728       }%
1729       {%
1730         \csuse{footstart#1}{#1}%
1731         \csuse{footgroup#1}{#1}%
1732         \global\count\csuse{footins#1}=\z@%
1733         \global\skip\csuse{footins#1}=\z@%
1734         \global\dimen\csuse{footins#1}=\maxdimen%
1735       }%
1736       {%
1737         \global\count\csuse{footins#1}=\csuse{default@footins#1}%
1738         \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
1739         \bgroup%
1740           \csuse{Xnotefontsize@#1}%
1741           \global\dimen\csuse{footins#1}=\csuse{maxhnotesX@#1}%
1742         \egroup%
1743         \global\setnamebox{footins#1@kept}=\box\namebox{footins#1}%
1744       }%
1745     }%
1746   }%
1747 }%
1748 %

```

`\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.

```

1749 \newcommand{\insert@notes@for@onlyside}{%
1750   \def\do##1{%
1751     \unless\ifnocritical@%
1752       \ifvoid\csuse{##1footins@kept}\else%
1753         \expandafter\insert\csname ##1footins\endcsname%
1754       \bgroup%
1755         \unvnamebox{##1footins@kept}%
1756       \egroup%

```

```

1757 \fi%
1758 \fi%
1759 \unless\ifnofamiliar@%
1760 \ifvoid\csuse{footins##1@kept}\else%
1761 \expandafter\insert\csname footins##1\endcsname%
1762 \bgroup%
1763 \unvnamebox{footins##1@kept}%
1764 \egroup%
1765 \fi%
1766 \fi%
1767 }%
1768 \dolistloop{\@series}%
1769 }%
1770 %

```

## IX.2 Critical footnote printed in right side

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

```

\edtextnow
1771 \newcount\edtext@now%
1772 \newcount\edtext@later%
1773 %
1774 \newcommand{\edtextlater}[2]{%#1 lemma, #2 critical notes
1775 %

```

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

```

1776 \global\advance\edtext@later by \@ne%
1777 %
1778 % 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}.
1779 \advance\@edtext@level by \@ne%
1780 \ifcsvoid{sw@list@edtext@\the\@edtext@level}%
1781 {\global\let\sw@inthisedtext\empty}%
1782 {\expandafter\gl@p\csname sw@list@edtext@\the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1783 %

```

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

```

1784 %
1785 \csxdef{edtext@later@\the\edtext@later}{%
1786 %

```

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

```

1787 \noexpand\edtext{%
1788 }%
1789 %

```

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

```

1790 {%
1791 \unexpanded{%
1792 \lemma{\no@expands #1}%
1793 \xxref{start:edtext:later:\the\edtext@later}{end:edtext:later:\the\
edtext@later}%
1794 }%
1795 \noexpand\linenum{||||\edfont@info}%
1796 \unexpanded{\def\sw@inthisedtext}{\expandonce{\sw@inthisedtext}}}%
1797 %

```

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.

```

1798 \ifledRcol%
1799 \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@true}}%
1800 \else%
1801 \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@false}}%
1802 \fi%
1803 %

```

And the footnote command of this `\edtextlater`.

```

1804 \unexpanded{#2}%
1805 }%
1806 }%
1807 %

```

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`

```

1808 \edlabel{start:edtext:later:\the\edtext@later}%
1809 \flag@start@later%
1810 \bgroup%
1811 \def\lemma##1{%
1812 \ifledRcol%
1813 \write\linenum@outR{\string\@lemma}%
1814 \else%
1815 \write\linenum@out{\string\@lemma}%
1816 \fi%
1817 }%
1818 \renewcommand{\do}[1]{\expandafter\renewcommandx\csname ##1footnote\
endcsname[2][1,usedefault]{}}%
1819 \dolistloop{\@series}%
1820 #2%
1821 \egroup%

```

```

1822 \showlemma{#1}%
1823 \edlabel{end:edtext:later:\the\edtext@later}%
1824 \flag@end@later%
1825 %

```

We decrease the counter increased at the beginning.

```

1826 \advance\edtext@level by -\@ne%
1827 }%
1828 %

```

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

```

1829 \newcommand{\edtextnow}[0]{%
1830 \global\advance\edtext@now by \@ne
1831 \csuse{edtext@later@\the\edtext@now}%
1832 }%
1833 %

```

## 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.

```

1834 \list@create{\labelref@listR}
1835
1836 %

```

**\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.

```

1837 \def\l@dmake@labelsR#1|#2|#3|#4|#5{%
1838 \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
else
1839 \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
1840 \fi
1841 \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
{#1|#2|#3|#4|\@Rlineflag}%
1842 \ignorespaces}
1843 \AtBeginDocument{%
1844 \def\l@dmake@labelsR#1|#2|#3|#4|#5{%
1845 }
1846
1847 %

```

**\@lab** The \@lab command, which appears in the \linenum@out file, appends the current values of page, line and sub-line to the \labelref@list. These values are defined by the earlier \@page, \@nl, and the \sub@on and \sub@off commands appearing in the \linenum@out file.

It is defined on reledmac.

## XI Side notes

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

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

`\sidenotemargin*`

```

1848 \WithSuffix\newcommand\sidenotemargin*[1]{%
1849   \l@dotsidesidenote@margin{#1}
1850   \global\sidenote@marginR=\@l@tempcntb
1851   \global\sidenote@margin=\@l@tempcntb
1852 }
1853 \newcount\sidenote@marginR
1854 \global\sidenote@margin=\@ne
1855
1856 %

```

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

```

1857 \newcommand*{\affixside@noteR}{%
1858   \prepare@edindex@fornote{\the\page@numR|\the\line@numR|\the\
subline@numR|\the\page@numR|\the\line@numR|\the\subline@numR|}%
1859   \def\sidenotecontent@{}%
1860   \numgdef{\itemcount@}{0}%
1861   \def\do##1{%
1862     \ifnumequal{\itemcount@}{0}%
1863       {%
1864         \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
1865       {\appto\sidenotecontent@{\sidenotessep ##1}%
1866       }%
1867     \numgdef{\itemcount@}{\itemcount@+1}%
1868   }%
1869   \dolistloop{\l@dcnotesetext}%
1870   \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
1871   \gdef\@templ@d{%
1872     \gdef\@templ@n{\l@dcnotesetext\l@dcnotesetext@1\l@dcnotesetext@r}%
1873     \ifx\@templ@d\@templ@n \else%
1874       \if@twocolumn%
1875         \if@firstcolumn%
1876           \setl@dlp@rbox{##1}{\sidenotecontent@}%
1877         \else%
1878           \setl@drp@rbox{\sidenotecontent@}%
1879         \fi%
1880       \else%
1881         \@l@tempcntb=\sidenote@marginR%
1882         \ifnum\@l@tempcntb>\@ne%
1883           \advance\@l@tempcntb by\page@numR%
1884         \fi%
1885         \ifodd\@l@tempcntb%

```



```

1886 \setl@drp@rbox{\sidenotecontent@}%
1887 \gdef\sidenotecontent@{}%
1888 \numdef{\itemcount@}{0}%
1889 \dolistloop{\l@dcsnotetext@l}%
1890 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
1891 \setl@dlp@rbox{\sidenotecontent@}%
1892 \else%
1893 \setl@dlp@rbox{\sidenotecontent@}%
1894 \gdef\sidenotecontent@{}%
1895 \numdef{\itemcount@}{0}%
1896 \dolistloop{\l@dcsnotetext@r}%
1897 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
1898 \setl@drp@rbox{\sidenotecontent@}%
1899 \fi%
1900 \fi%
1901 \fi%
1902 \advance\@edindex@fornote@m@ne%
1903 }
1904
1905 %

```

## 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.

```

\inserthangingsymbolL906 \newif\ifinserthangingsymbolR
\inserthangingsymbolR907 \newcommand{\inserthangingsymbolL}{%
1908 \ifinserthangingsymbol%
1909 \ifinstanzaL%
1910 \hskip \@ifundefined{sza@00}{0}{\expandafter%
1911 \noexpand\csname sza@00\endcsname}\stanzaindentbase%
1912 \@hangingsymbol%
1913 \fi%
1914 \fi%
1915 }%
1916 \newcommand{\inserthangingsymbolR}{%
1917 \ifinserthangingsymbolR%
1918 \ifinstanzaR%
1919 \hskip \@ifundefined{sza@00}{0}{\expandafter%
1920 \noexpand\csname sza@00\endcsname}\stanzaindentbase%
1921 \@hangingsymbol%
1922 \fi%
1923 \fi%
1924 }%
1925 %

```

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.

```

1926 \chardef\next=\catcode`\&
1927 \catcode`\&=\active
1928
1929 %

```

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

```

1930 \newenvironment{astanza}[1][\%
1931 \catcode`\&=\active
1932 \global\stanza@count\@ne\stanza@modulo\@ne
1933 \ifnum\usernamecount{sza@0@}=\z@
1934 \let\stanza@hang\relax
1935 \let\endlock\relax
1936 \else
1937 \rightskip\z@ plus 1fil\relax
1938 \fi
1939 \ifnum\usernamecount{szp@0@}=\z@
1940 \let\sza@penalty\relax
1941 \fi
1942 \def&{%
1943 \endlock\mbox{}%
1944 \sza@penalty
1945 \global\advance\stanza@count\@ne
1946 \@astanza@line}%
1947 \def\&{\@stopastanza}%
1948 \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}} and test{\ifstrempy
1949 {#1}}}%
1950 {\pstart[\at@every@stanza]}%
1951 {\pstart[#1]}%
1952 \@astanza@line
1953 \let\par\relax\ignorespaces%No paragraph in verses
1954 }{}
1955 %

```

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

```

1956 \newcommandx{\@stopastanza}[1][1,usedefault]{%
1957 \endlock\mbox{}%
1958 \ifboolexpr{not test{\ifdefvoid{\at@every@stop@stanza}} and test{\
1959 ifstrempy{#1}}}%
1960 {\pend[\at@every@stop@stanza]}%
1961 {\pend[#1]}%
1962 }%

```

`\@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.

```

1963 \newcommand*{\@astanza@line}{%
1964   \ifnum\value{stanzaindentrepetition}=0
1965     \ifcsdef{sza@\number\stanza@count @}{%
1966       {%
1967         \parindent=\csname sza@\number\stanza@count @\endcsname\
stanzaindentbase%
1968       }{%
1969         \led@err@StanzaIndentNotDefined%
1970       }%
1971     \else
1972       \ifcsdef{sza@\number\stanza@modulo @}{%
1973         \parindent=\csname sza@\number\stanza@modulo @\endcsname\
stanzaindentbase%
1974         \managestanza@modulo%
1975       }%
1976       {%
1977         \led@err@StanzaIndentNotDefined%
1978       }%
1979     \fi
1980   \endgraf
1981   \stanza@hang%
1982   \ignorespaces}
1983 %
1984 %

```

Lastly reset the modified category codes.

```

1985 \catcode`\&=\next
1986 %
1987 %

```

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

```

\thestanzaR
1988 \newcounter{stanzaL}
1989 \newcounter{stanzaR}
1990 \renewcommand{\thestanzaL}{%
1991   \textbf{\arabic{stanzaL}}%
1992 }
1993 \renewcommand{\thestanzaR}{%
1994   \textbf{\arabic{stanzaR}}%
1995 }
1996 %
1997 %

```

## XIII Fixing babel and polyglossia

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

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

```
\ifl@dusedbabel A flag for checking if babel has been used as a package.
\l@dusedbabelfalse
\l@dusedbabeltrue1998 \newif\ifl@dusedbabel
1999 %
```

```
\l@dchecklang
```

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

```
2000 \patchcmd{\bbl@set@language}%
2001 {\select@language{\language}}}%
2002 {\edef\language{#1}\select@language{\language}}}%
2003 {}%
2004 {}%
2005
2006 %
```

The rest of the setup has to be postponed until the end of the preamble when we know if babel or polyglossia have been used or not. However, for now assume that it has not been used.

```
\selectlanguage \selectlanguage is a babel command. \theledlanguageL and \theledlanguageR
\l@duselanguage are the names of the languages of the left and right texts. \l@duselanguage is similar
\theledlanguageL to \selectlanguage.
\theledlanguageR
2007 \newcommand*{\l@duselanguage}[1]{ }
2008 \gdef\theledlanguageL{ }
2009 \gdef\theledlanguageR{ }
2010
2011 %
```

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

```

2012 \AtBeginDocument{%
2013   \ifundefined{xpg@main@language}{%
2014     \ifundefined{bbl@main@language}{%
2015 %

```

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

```

2016   \l@dusedbabelfalse
2017 }{%
2018 %

```

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.

```

2019   \l@dusedbabeltrue
2020   \let\l@doldselectlanguage\selectlanguage
2021   \let\l@doldbbl@set@language\bbl@set@language
2022   \renewcommand{\selectlanguage}[1]{%
2023     \l@doldselectlanguage{#1}%
2024     \ifledRcol \gdef\theledlanguageR{#1}%
2025     \else      \gdef\theledlanguageL{#1}%
2026     \fi}
2027 %

```

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

```

2028   \renewcommand*{\l@duselanguage}[1]{%
2029     \expandafter\l@doldselectlanguage\expandafter{#1}}
2030 %

```

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

```

2031   \gdef\theledlanguageL{\bbl@main@language}%
2032   \gdef\theledlanguageR{\bbl@main@language}%
2033   }%
2034 }
2035 %

```

If use polyglossia

```

2036 { \let\old@otherlanguage\otherlanguage%
2037   \renewcommand{\otherlanguage}[2] []{%
2038     \selectlanguage{#1}{#2}%
2039     \ifledRcol \gdef\theledlanguageR{#2}%
2040     \else      \gdef\theledlanguageL{#2}%
2041     \fi}%
2042   \renewcommand{\l@duselanguage}[1]{%
2043     \csuse{no\language@name @numbers}\select@language{#1}%
2044   }%
2045   \gdef\theledlanguageL{\xpg@main@language}%
2046   \gdef\theledlanguageR{\xpg@main@language}%
2047 %

```

That is it.

```
2048 }}
2049 %
```

## 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`

```
2050 \newcount\l@dc@maxchunks
2051 \newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}
2052 \maxchunks{5120}
2053
2054 %
```

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

`\l@dnumpstartsR`  

```
2055 \newcount\l@dnumpstartsR
2056
2057 %
```

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

`\l@pscR`  

```
2058 \newcount\l@dpscL
2059 \newcount\l@dpscR
2060
2061 %
```

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

```
2062 \newcommand*{\l@dsetuprawboxes}{%
2063   \l@dttempcntb=\l@dc@maxchunks
2064   \loop\ifnum\l@dttempcntb>\z@
2065     \newnamebox{\l@dLcolrawbox\the\l@dttempcntb}
2066     \newnamebox{\l@dRcolrawbox\the\l@dttempcntb}
2067     \advance\l@dttempcntb \m@ne
2068   \repeat}
2069
2070 %
```

`\l@dssetupmaxlinecounts` 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@dssetupmaxlinecounts` creates `\maxchunks` new counts called `\l@dmmaxlinesinpar1`, etc., and `\l@dzeromaxlinecounts` zeroes all of them.

```

2071 \newcommand*{\l@dssetupmaxlinecounts}{%
2072   \l@l@tempcntb=\l@dc@maxchunks
2073   \loop\ifnum\l@l@tempcntb>\z@
2074     \newnamecount{\l@dmmaxlinesinpar\the\l@l@tempcntb}
2075     \advance\l@l@tempcntb \m@ne
2076   \repeat}
2077 \newcommand*{\l@dzeromaxlinecounts}{%
2078   \begin{group}
2079   \l@l@tempcntb=\l@dc@maxchunks
2080   \loop\ifnum\l@l@tempcntb>\z@
2081     \global\usenamecount{\l@dmmaxlinesinpar\the\l@l@tempcntb}=\z@
2082     \advance\l@l@tempcntb \m@ne
2083   \repeat
2084   \end{group}}
2085
2086 %

```

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

```

2087 \AtBeginDocument{%
2088   \l@dssetuprawboxes
2089   \l@dssetupmaxlinecounts
2090   \l@dzeromaxlinecounts
2091   \l@dnumpstartsL=\z@
2092   \l@dnumpstartsR=\z@
2093   \l@dpscL=\z@
2094   \l@dpscR=\z@}
2095
2096 %

```

## XV Checking text to be processed

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

`\@pstartstrue` `\newif\if@pstarts`

`\@pstartsfalse` `\newcommand*{\check@pstarts}{%`

`\check@pstarts` `\@pstartsfalse`

```

2097 \ifnum\l@dnumpstartsL>\l@dpscL
2098   \@pstartstrue
2099 \else
2100   \ifnum\l@dnumpstartsR>\l@dpscR
2101     \@pstartstrue
2102   \fi
2103
2104 \fi
2105

```

```

2106 \fi
2107 }
2108
2109 %

```

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

```

\checkraw@text
2110 \newif\ifaraw@text
2111 \newcommand*{\checkraw@text}{%
2112   \araw@textfalse
2113   \ifvbox\namebox{1@dLcolrawbox\the\l@dpscL}
2114     \araw@texttrue
2115   \else
2116     \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}
2117       \araw@texttrue
2118   \fi
2119 \fi
2120 }
2121
2122 %

```

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

```

2123 \newcommand*{\@writelinesinparL}{%
2124   \edef\next{%
2125     \write\linenum@out{\string\@pend[\the\@donereallinesL]}}%
2126   \next
2127   \global\@donereallinesL \z@}
2128 \newcommand*{\@writelinesinparR}{%
2129   \edef\next{%
2130     \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}%
2131   \next
2132   \global\@donereallinesR \z@}
2133
2134 %

```

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

```

\@writepageofparR
2135 \newcommand*{\@writepageofparL}[0]{%
2136   \ifnum\@donereallinesL=\z@%
2137     \edef\next{%
2138       \write\linenum@out{\string\@pstart{\the\l@dpscL}{\the\c@page}{\the\
2139       numpagelinesL}}%
2140     }%
2141     \next%
2142   \fi%
}

```



```

2143 \newcommand*{\@writepageofparR}[0]{%
2144   \ifnum\@donereallinesR=\z@%
2145     \edef\next{%
2146       \write\linenum@outR{\string\@pstartR{\the\l@dpscR}{\the\c@page}{\the\
numpagelinesR}}}%
2147   }%
2148   \next%
2149   \fi%
2150 }%
2151 %

```

## XVI Parallel columns

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

```

\@eledsectionR
2152 \newsavebox{\@eledsectionL}%
2153 \newsavebox{\@eledsectionR}%
2154 %

```

**\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.

```

2155 \newcommand*{\Columns}{%
2156   \ifl@dpairing%
2157     \led@err@Columns@InsideEnv%
2158   \fi%
2159   \ifbool@expr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{
l@dLcolrawbox1}}}{%
2160     \ifcsboxvoid{l@dRcolrawbox1}{%
2161       \ifcsboxvoid{l@dLcolrawbox1}%
2162       {\led@err@Columns@WithoutEnv}%
2163       {\led@err@Columns@WithoutRightside}%
2164     }%
2165     {\led@err@Columns@WithoutLeftside}%
2166   }{%
2167     \global\l@dprintingcolumnstrue%
2168     \eledsection@correcting@skip=-\baselineskip% Correction for sections'
titles
2169     \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else
2170       \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
2171     \fi
2172   }%

```

Start a group and zero counters, etc.

```

2173 \begingroup
2174   \l@dzeropenalties
2175   \endgraf\global\num@lines=\prevgraf
2176   \global\num@linesR=\prevgraf
2177   \global\par@line=\z@

```

```

2178 \global\par@lineR=\z@
2179 \global\l@dpscL=\z@
2180 \global\l@dpscR=\z@
2181 \get@familiarfootnote@number%
2182 %

```

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

```

2183 \check@pstarts
2184 \loop\if@pstarts
2185 \global\pstartnumtrue
2186 \global\pstartnumRtrue
2187 %

```

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

```

2188 \global\advance\l@dpscL \@ne
2189 \global\advance\l@dpscR \@ne
2190 \restore@pstartL@pc%
2191 \restore@pstartR@pc%
2192 %

```

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

```

2193 \Columns@print@before@pstart%
2194 %

```

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

```

2195 \checkraw@text
2196 { \loop\ifaraw@text
2197 %

```

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

```

2198 \l@duselanguage{\theledlanguageL}%
2199 \do@lineL
2200 \xifinlist{\the\l@dpscL}{\eled@sections@@}
2201 {%
2202 \ifdefstring{\@eledsectmark}{L}%
2203 {\csuse{eled@sectmark@the\l@dpscL}%
2204 }}%
2205 \global\csundef{eled@sectmark@the\l@dpscL}%
2206 \savebox{\@eledsectionL}{\parbox[t]{}[t]{\Lcolwidth}{\vbox
{\print@eledsectionL}}}%\vbox{}-> prevent alignment troubles with RTL
language
2207 }%
2208 {}%
2209 \l@duselanguage{\theledlanguageR}%
2210 \do@lineR

```

```

2211 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}
2212 {%
2213 \ifdefstring{\@eledsectmark}{R}%
2214 {\csuse{eled@sectmark@the\l@dpscR R}%
2215 }{}%
2216 \global\csundef{eled@sectmark@the\l@dpscR R}%
2217 \savebox{\@eledsectionR}{\parbox[t]{}[t]{\Rcolwidth}{\vbox
{} \print@eledsectionR}}}% \vbox{} -> prevent alignment troubles with RTL
language
2218 {}%
2219 \hb@xt@ \hsize{%
2220 \ifdefstring{\columns@position}{L}{-}{\hfill }%
2221 \unhbox\l@dleftbox%
2222 \ifhbox\@eledsectionL%
2223 \usebox{\@eledsectionL}%
2224 \fi%
2225 \print@columnseparator%
2226 \unhbox\l@drightbox%
2227 \ifhbox\@eledsectionR%
2228 \usebox{\@eledsectionR}%
2229 \fi%
2230 \ifdefstring{\columns@position}{R}{-}{\hfill}%
2231 }%
2232 \checkraw@text
2233 \checkverseL
2234 \checkverseR
2235 \checkpb@columns
2236 \repeat}
2237 %

```

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.

```

2238 \@writelinesinparL
2239 \@writelinesinparR
2240 \check@pstarts
2241 \ifbypstart@%
2242 \write\linenum@out{\string\@set[1]}
2243 \resetprevline@
2244 \fi
2245 \ifbypstart@R
2246 \write\linenum@outR{\string\@set[1]}
2247 \resetprevline@
2248 \fi
2249 \Columns@print@after@pend%
2250 \repeat
2251 %

```

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.

```

2252 \flush@notes
2253 \flush@notesR
2254 \endgroup
2255 %

2256 \global\l@dpscL=\z@
2257 \global\l@dpscR=\z@
2258 \global\l@dnpstartsL=\z@
2259 \global\l@dnpstartsR=\z@
2260 \global\l@dprintingcolumnsfalse%
2261 \ignorespaces
2262 \global\instanzaLfalse%
2263 \global\instanzaRfalse%
2264 }%
2265 }%
2266
2267 %

```

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

```

2268 \def\print@columnseparator{%
2269 \ifdim\beforecolumnseparator<0pt%
2270 \hfill%
2271 \else%
2272 \hspace{\beforecolumnseparator}%
2273 \fi%
2274 \columnseparator%
2275 \ifdim\aftercolumnseparator<0pt%
2276 \hfill%
2277 \else%
2278 \hspace{\aftercolumnseparator}%
2279 \fi%
2280 }%
2281 %

```

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

```

2282
2283 \newcommand{\checkpb@columns}{%
2284 \newif\if@pb
2285 \newif\if@nopb
2286 \IfStrEq{\led@pb@setting}{before}{
2287 \numdef{\next@absline}{\the\absline@num+1}%
2288 \numdef{\next@abslineR}{\the\absline@numR+1}%
2289 \xifinlistcs{\next@absline}{l@prev@pb}{\@pbtrue}{}%
2290 \xifinlistcs{\next@abslineR}{l@prev@pbR}{\@pbtrue}{%
2291 \xifinlistcs{\next@absline}{l@prev@nopb}{\@nopbtrue}{}%

```

```

2292 \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\@nopbtrue}{}
2293 }{}
2294 \IfStrEq{\led@pb@setting}{after}{
2295 \xifinlistcs{\the\absline@num}{l@prev@pb}{\@pbtrue}{}%
2296 \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\@pbtrue}{}
2297 \xifinlistcs{\the\absline@num}{l@prev@nopb}{\@nopbtrue}{}%
2298 \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\@nopbtrue}{}
2299 }{}
2300 \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
2301 \if@pb\pagebreak[4]\fi
2302 }
2303 %

```

**\columnseparator** The separator between line pairs in parallel columns is in the form of a vertical rule extending a little below the baseline and with a height slightly greater than the **\baselineskip**. The width of the rule is **\columnrulewidth** (initially 0pt so the rule is invisible).

```

2304 \newcommand*{\columnseparator}{%
2305 \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}}
2306 \newdimen\columnrulewidth
2307 \columnrulewidth=\z@
2308
2309 %

```

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

```

2310 \newcommand*{\columnspan}[1]{%
2311 \xdef\columns@position{#1}%
2312 }%
2313 \xdef\columns@position{R}%
2314 %

```

**\beforecolumnseparator** **\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**.

```

2315 \newlength{\beforecolumnseparator}%
2316 \setlength{\beforecolumnseparator}{-2pt}%
2317
2318 \newlength{\aftercolumnseparator}%
2319 \setlength{\aftercolumnseparator}{-2pt}%
2320
2321 %

```

**\setwidthliketwocolumns@L** **\tpositionliketwocolumns@L** **\epositionliketwocolumns@L** **\setwidthliketwocolumns@C** **\tpositionliketwocolumns@C** **\epositionliketwocolumns@C** **\setwidthliketwocolumns@R** **\tpositionliketwocolumns@R** **\epositionliketwocolumns@R**

The **\setwidth...** macros are called in **\beginnumbering** in a **non-parallel** typesetting context, to fix the width of the lines to be vertically aligned with parallel columns. They are also called at the beginning of a note's group, if some options are enabled. The **\setposition...** macros are called in **\beginnumbering** in a **non- parallel** typesetting

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

```

2322 \newcommand{\setwidthliketwocolumns@L}{%
2323 % Temporary dimension, initially equal to the standard hsize, i.e. text
      width
2324 %   \begin{macrocode}
2325   \newdimen\temp%
2326   \temp=\hsize%
2327 %

```

Hsize : Left + Right width

```

2328   \hsize=\Lcolwidth%
2329   \advance\hsize\Rcolwidth%
2330 %

```

Now, calculating the remaining space

```

2331   \advance\temp-\hsize%
2332 %

```

And multiply the hsize by 2/3 of this space

```

2333   \multiply\temp by 2%
2334   \divide\temp by 3%
2335   \advance\hsize\temp%
2336 }%
2337
2338 \newcommand{\setpositionliketwocolumns@L}{%
2339   \renewcommand{\ledrlfill}{\hfill}%
2340 }%
2341
2342 \newcommand{\setnotespositionliketwocolumns@L}{%
2343 }%
2344
2345
2346 %

```

```

2347 \newcommand{\setwidthliketwocolumns@C}{%
2348 % Temporary dimension, initially equal to the standard hsize, i.e. text
      width
2349 %

```

```

2350   \newdimen\temp%
2351   \temp=\hsize%
2352 % Hsize : Left + Right width
2353 %

```

```

2354   \hsize=\Lcolwidth%
2355   \advance\hsize\Rcolwidth%
2356 % Now, calculating the remaining space
2357 %

```

```

2358 \advance\temp-\hsize%
2359 %

```

And multiply the hsize by 1/2 of this space

```

2360 \divide\temp by 2%
2361 \advance\hsize\temp%
2362 }%
2363
2364 \newcommand{\setpositionliketwocolumns@C}{%
2365 \doinsidelinehook{\hfill}%
2366 \renewcommand{\ledrlfill}{\hfill}%
2367 }%
2368
2369 \newcommand{\setnotespositionliketwocolumns@C}{%
2370 \newdimen\temp%
2371 \newdimen\tempa%
2372 \temp=\hsize%
2373 \tempa=\Lcolwidth%
2374 \advance\tempa\Rcolwidth%
2375 \advance\temp-\tempa%
2376 \divide\temp by 2%
2377 \leftskip=\temp%
2378 \rightskip=-\temp%
2379 }%
2380
2381 \newcommand{\setwidthliketwocolumns@R}{%
2382 %

```

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

```

2383 \newdimen\temp%
2384 \temp=\hsize%
2385 %

```

Hsize : Left + Right width

```

2386 \hsize=\Lcolwidth%
2387 \advance\hsize\Rcolwidth%
2388 %

```

Now, calculating the remaining space

```

2389 \advance\temp-\hsize%
2390 %

```

And multiply the hsize by 2/3 of this space

```

2391 \multiply\temp by 2%
2392 \divide\temp by 3%
2393 \advance\hsize\temp%
2394 }%
2395
2396 \newcommand{\setpositionliketwocolumns@R}{%

```

```

2397 \doinsidelinehook{\hfill}%
2398 }%
2399
2400 \newcommand{\setnotespositionliketwocolumns@R}{%
2401 \newdimen\temp%
2402 \newdimen\tempa%
2403 \temp=\hsize%
2404 \tempa=\Lcolwidth%
2405 \advance\tempa\Rcolwidth%
2406 \advance\temp-\tempa%
2407 \divide\temp by 2%
2408 \leftskip=\temp%
2409 \rightskip=-\temp%
2410 }%
2411
2412 %

```

`\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.

```

2413 \newcommand{\Columns@print@before@pstart}{%
2414 \ifboolexpr{%
2415 test{\ifcsstring{before@pstartL@the\l@dpscl}{\at@every@pstart}}%
2416 and test {\ifcsstring{before@pstartR@the\l@dpscl}{\at@every@pstart}}%
2417 and test {\ifdefempty{\at@every@pstart}}}%
2418 {}%
2419 {%
2420 \hb@xt@ \hsize{%
2421 \ifdefstring{\columns@position}{L}{\hfill }%
2422 \par\parbox[t] [] [t]{\Lcolwidth}{%
2423 \csuse{before@pstartL@the\l@dpscl}%
2424 }%
2425 \print@columnseparator%
2426 \parbox[t] [] [t]{\Rcolwidth}{%
2427 \set@sectcountR%
2428 \csuse{before@pstartR@the\l@dpscl}%
2429 }%
2430 \ifdefstring{\columns@position}{R}{\hfill}%
2431 }%
2432 }%
2433 \global\csundef{before@pstartL@the\l@dpscl}%
2434 \global\csundef{before@pstartR@the\l@dpscl}%
2435 }%
2436 \newcommand{\Columns@print@after@pend}{%
2437 \ifboolexpr{%
2438 test{\ifcsstring{after@pendL@the\l@dpscl}{\at@every@pend}}%
2439 and test {\ifcsstring{after@pendR@the\l@dpscl}{\at@every@pend}}%
2440 and test {\ifdefempty{\at@every@pend}}}%
2441 {}%

```



```

2442 {%
2443   \hb@xt@ \hsize{%
2444     \ifdefstring{\columns@position}{L}{\hfill }%
2445     \parbox[t] [] [t]{\Lcolwidth}{%
2446       \csuse{after@pendL@the\l@dpscl}%
2447     }%
2448     \print@columnseparator%
2449     \parbox[t] [] [t]{\Rcolwidth}{%
2450       \set@sectcountR%
2451       \csuse{after@pendR@the\l@dpscr}%
2452     }%
2453     \ifdefstring{\columns@position}{R}{\hfill}%
2454   }%
2455 }%
2456 \global\csundef{after@pendL@the\l@dpscl}%
2457 \global\csundef{after@pendR@the\l@dpscr}%
2458 }%
2459 %

```

## 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`

```

2460 \newcount\numpagelinesL
2461 \newcount\numpagelinesR
2462 \newcount\l@dminpagelines
2463
2464 %

```

### 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.

```

2465 \newcommand*{\Pages}[1][1,usedefault]{%
2466   \ifl@dpairing%
2467   \led@err@Pages@InsideEnv%
2468   \fi%
2469   \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{
l@dLcolrawbox1}}}{%
2470     \ifcsboxvoid{l@dRcolrawbox1}{%
2471       \ifcsboxvoid{l@dLcolrawbox1}%

```

```

2472         {\led@err@Pages@WithoutEnv}%
2473         {\led@err@Pages@WithoutRightside}%
2474     }%
2475     {\led@err@Pages@WithoutLeftside}%
2476 }{%
2477 \ifstrequal{#1}{mainmatter}{\Pages@mainmattertrue}{\Pages@mainmatterfalse}%
2478 }%
2479 \eledsection@correcting@skip=-2\baselineskip% line correcting for section
titles.
2480 \parledgroup@notespacing@set@correction%
2481 \typeout{}%
2482 \typeout{***** PAGES *****}%
2483 \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else%
2484     \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
2485 \fi%
%
```

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

```

2486 \cleartol@devenpage%
2487 \global\l@dprintingpagetrue%
2488 \begingroup%
2489 %

```

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

```

2490 \setlength{\Lcolwidth}{\textwidth}%
2491 \setlength{\Rcolwidth}{\textwidth}%
2492 %

```

```

2493 \l@dzeropenalties%
2494 \endgraf\global\num@lines=\prevgraf%
2495     \global\num@linesR=\prevgraf%
2496 \global\par@line=\z@%
2497 \global\par@lineR=\z@%
2498 \global\l@dpscL=\z@%
2499 \global\l@dpscR=\z@%
2500 \writtenlinesLfalse%
2501 \writtenlinesRfalse%
2502 \get@familiarfootnote@number%
2503 %

```

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

```

2504 \let\print@Xnotes\print@Xnotes@forpages%
2505 \let\print@notesX\print@notesX@forpages%
2506 %

```

Check if there are chunks to be processed.

```

2507 \check@pstarts%

```

```
2508 \loop\if@pstarts%
2509 %
```

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

```
2510 \global\advance\l@dpscL \@ne%
2511 \global\advance\l@dpscR \@ne%
2512 %
```

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

```
2513 \getlinesfromparlistL%
2514 \getlinesfromparlistR%
2515 \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}%
2516 {\usernamecount{\l@dmaxlinesinpar\the\l@dpscL}}%
2517 \check@pstarts%
2518 \repeat%
2519 %
```

Zero the counts again, ready for the next bit.

```
2520 \global\l@dpscL=\z@%
2521 \global\l@dpscR=\z@%
2522 %
```

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

```
2523 \getlinesfrompagelistL%
2524 \getlinesfrompagelistR%
2525 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
2526 {\l@dminpagelines}%
2527 %
```

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.

```
2528 \check@pstarts%
2529 \if@pstarts%
2530 %
```

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

```
2531 \global\advance\l@dpscL \@ne%
2532 \global\advance\l@dpscR \@ne%
2533 \restore@pstartL@pc%
2534 \restore@pstartR@pc%
2535 %
```

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

```

2536 \global\@donereallinesL=\z@%
2537 \global\@donetotallinesL=\z@%
2538 \global\@donereallinesR=\z@%
2539 \global\@donetotallinesR=\z@%
2540 %

```

Start a loop over the boxes (chunks).

```

2541 \checkraw@text%
2542 %
2543 % \begingroup
2544 { \loop\ifaraw@text%
2545 %

```

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

```

2546 \checkpageL%
2547 \l@duselanguage{\theledlanguageL}%
2548 { \loop\ifl@dsamepage%
2549 %

```

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

```

2550 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}%
2551 \csuse{before@pstartL@the\l@dpscl}%
2552 \global\csundef{before@pstartL@the\l@dpscl}%
2553 \do@lineL%
2554 \xifinlist{\the\l@dpscl}{\eled@sections@@}
2555 {\print@eledsectionL}%
2556 {}%
2557 \advance\numpagelinesL \@ne%
2558 %

```

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`

```

2559 \ifshiftedpstarts%
2560 \ifdim\ht\l@dleftbox>0pt%
2561 \parledgroup@correction@notespacing{L}%
2562 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2563 \else%
2564 \unless\ifadvancedshiftedpstarts%
2565 \dimen0=\pagetotal%
2566 \advance\dimen0 by \baselineskip%
2567 \global\pagetotal=\dimen0%
2568 \else%

```

```

2569         \ifnomaxlines%
2570             \numdef{\@tmp}{\the\l@dpscL+1}%
2571             \ifcsdef{minpage@pstart@\@tmp}{%
2572                 \ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}}%
2573                 {\dimen0=\pagetotal%
2574                 \advance\dimen0 by \baselineskip%
2575                 \global\pagetotal=\dimen0%
2576                 }%
2577                 {}%
2578             }{}%
2579         \fi%
2580     \fi%
2581 \fi%
2582 \else%
2583     \parledgroup@correction@notespacing{L}%
2584     \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2585 \fi%
2586 %

```

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

```

2587     \get@nextboxL%
2588     \global\l@dskipversenumberfalse%
2589     \ifprint@last@after@pendL%
2590         \csuse{after@pendL@\the\l@dpscL}%
2591         \global\csundef{after@pendL@\the\l@dpscL}%
2592     \fi%
2593     \checkpageL%
2594     \checkverseL%
2595     \checkpbL%
2596     \repeat%
2597 %

```

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.

```

2598     \ifl@dpagefull%
2599         \@writelinesonpageL{\the\numpagelinesL}%
2600     \else%
2601         \@writelinesonpageL{1000}%
2602     \fi%
2603 %

```

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.

```

2604 \numpagelinesL \z%
2605 \parledgroup@correction@notespacing@init%
2606 \clearl@dleftpage }%
2607 %

```

Now do the same for the right text.

```

2608 \checkpageR%
2609 \l@duselanguage{\theledlanguageR}%
2610 {
2611   \loop\ifl@dsamepage%
2612     \set@sectcountR%
2613     \ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{}%
2614     \csuse{before@pstartR@the\l@dpscR}%
2615     \global\csundef{before@pstartR@the\l@dpscR}%
2616     \do@lineR%
2617     \xifinlist{the\l@dpscR}{\eled@sectionsR@@}%
2618     {\print@eledsectionR}%
2619     {}%
2620   \advance\numpagelinesR \@ne%
2621   \ifshiftedpstarts%
2622     \ifdim\ht\l@drightbox>0pt%
2623       \parledgroup@correction@notespacing{R}%
2624       \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2625     \else%
2626       \unless\ifadvancedshiftedpstarts%
2627         \dimen0=\pagetotal%
2628         \advance\dimen0 by \baselineskip%
2629         \global\pagetotal=\dimen0%
2630       \else%
2631         \ifnomaxlines%
2632           \numdef{\@tmp}{\the\l@dpscR+1}%
2633           \ifcsdef{minpage@pstart@\@tmp}{%
2634             \ifnumless{the\c@page}{\csuse{
minpage@pstart@\@tmp}}%
2635               {\dimen0=\pagetotal%
2636                 \advance\dimen0 by \baselineskip%
2637                 \global\pagetotal=\dimen0%
2638                 }%
2639               {}%
2640             }%
2641           \fi%
2642         \fi%
2643       \else%
2644         \parledgroup@correction@notespacing{R}%
2645         \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2646       \fi%
2647     \get@nextboxR%
2648     \global\l@dskipversenumberRfalse%
2649     \ifprint@last@after@pendR%
2650       \csuse{after@pendR@the\l@dpscR}%

```

```

2651         \global\csundef{after@pendR@the\l@dpscR}%
2652         \fi%
2653         \checkpageR%
2654         \checkverseR%
2655         \checkpbR%
2656         \repeat%
2657         \ifl@pagefull%
2658         \@writelinesonpageR{\the\numpagelinesR}%
2659         \else%
2660         \@writelinesonpageR{1000}%
2661         \fi%
2662         \numpagelinesR=\z@%
2663         \parledgroup@correction@notespacing@init%
2664 %

```

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

```

2665         \clearl@drighthpage}%
2666 %

```

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.

```

2667         \checkraw@text%
2668         \ifaraw@text%
2669         \getlinesfrompagelistL%
2670         \getlinesfrompagelistR%
2671         \l@dcalcm@minoftwo{\cs@linesonpageL}{\cs@linesonpageR}%
2672         {\l@dminpagelines}%
2673         \fi%
2674         \repeat}%
2675 %

```

We have now output the text from all the chunks.

```

2676         \fi%
2677 %

```

Make sure that there are no inserts hanging around.

```

2678         \flush@notes%
2679         \flush@notesR%
2680         \endgroup%
2681 %

```

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

```

2682         \global\l@dpscL=\z@%
2683         \global\l@dpscR=\z@%
2684         \global\l@dnumpstasL=\z@%
2685         \global\l@dnumpstasR=\z@%
2686         \global\instanzaLfalse%
2687         \global\instanzaRfalse%

```

```

2688 \global\l@dprintingpagesfalse%
2689 %

```

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

```

2690 \ifnum\edtext@later=\edtext@now%
2691 \else%
2692 \led@error@edtext@later@now%
2693 \fi%
2694 \global\edtext@later=\z@%
2695 \global\edtext@now=\z@%
2696 %

```

Prevent final notes from overlapping the line number

```

2697 \finish@Pages@notes%
2698 \ignorespaces}}}%
2699
2700
2701 %

```

### 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.

```

2702 \newcommand{\finish@Pages@notes}{%
2703 \def\do##1{%
2704 %

```

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

```

2705 \ifnocritical@%
2706 \global\newnamebox{##1footins}%
2707 \fi
2708 \ifnofamiliar@%
2709 \global\newnamebox{footins##1}%
2710 \fi
2711 %

```

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

```

2712 \ifvoid\csuse{##1footins}%
2713 \ifvoid\csuse{footins##1}\else%
2714 \newpage\null%
2715 \listbreak%
2716 \fi%
2717 \else%
2718 \newpage\null%
2719 \listbreak%
2720 \fi%
2721 }%

```



```

2722 \dolistloop{\@series}%
2723 }%
2724 %

```

## XVII.4 Struts

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

```

\ledstrutR
2725 \newcommand*\ledstrutL{}
2726 \newcommand*\ledstrutR{}
2727
2728 %

```

## XVII.5 Page clearing

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

```

2729 \providecommand{\cleartoevenpage}[1][\@empty]{%
2730 \clearpage
2731 \ifodd\c@page\hbox{ }#1\clearpage\fi}
2732
2733 \newcommand*\cleartol@evenpage{%
2734 \ifdim\pagetotal<\topskip% on an empty page
2735 \else
2736 \clearpage
2737 \Pages@mainmatter%
2738 \fi
2739 \ifodd\c@page%
2740 \ifprevpgnotnumbered%
2741 \addtocounter{par@page}{-1}%
2742 \ifdef\prevpgstyle{\thispagestyle{\prevpgstyle}}}%
2743 \fi%
2744 \hbox{ }\clearpage%
2745 \fi%
2746 }%
2747 %

```

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

```

2748 \newcommand*\clearl@leftpage{%
2749 \ifdim\pagetotal=0pt\hbox{ }\fi%
2750 \newpage%

```

```

2751 \insert@notes@for@onlyside%
2752 \ifodd\c@page\else
2753 \led@err@LeftOnRightPage
2754 \hbox{}%
2755 \cleardoublepage
2756 \fi}
2757
2758 \newcommand*{\clearl@drighthouse}{%
2759 \ifdim\pagetotal=0pt\hbox{}\fi%
2760 \newpage%
2761 \insert@notes@for@onlyside%
2762 \ifodd\c@page
2763 \led@err@RightOnLeftPage
2764 \hbox{}%
2765 \cleartoevenpage
2766 \fi}
2767
2768 %

```

## XVII.6 Lines managing

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

`\@cs@linesinparR`

```

2769 \newcommand*{\getlinesfromparlistL}{%
2770 \ifx\linesinpar@listL\empty
2771 \gdef\@cs@linesinparL{0}%
2772 \else
2773 \gl@p\linesinpar@listL\to\@cs@linesinparL
2774 \fi}
2775 \newcommand*{\getlinesfromparlistR}{%
2776 \ifx\linesinpar@listR\empty
2777 \gdef\@cs@linesinparR{0}%
2778 \else
2779 \gl@p\linesinpar@listR\to\@cs@linesinparR
2780 \fi}
2781
2782 %

```

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

`\@cs@linesonpageR`

```

2783 \newcommand*{\getlinesfrompagelistL}{%
2784 \ifx\linesonpage@listL\empty
2785 \gdef\@cs@linesonpageL{1000}%
2786 \else
2787 \gl@p\linesonpage@listL\to\@cs@linesonpageL

```

```

2788 \fi}
2789 \newcommand*\getlinesfrompagelistR{%
2790 \ifx\linesonpage@listR\empty
2791 \gdef\@cs@linesonpageR{1000}%
2792 \else
2793 \gl@p\linesonpage@listR\to\@cs@linesonpageR
2794 \fi}
2795
2796 %

```

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

```

2797 \newcommand*\@writelinesonpageL}[1]{%
2798 \edef\next{\write\linenum@out{\string\@lopL{#1}}}%
2799 \next}
2800 \newcommand*\@writelinesonpageR}[1]{%
2801 \edef\next{\write\linenum@outR{\string\@lopR{#1}}}%
2802 \next}
2803
2804 %

```

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

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

```

2805 \newcommand*\l@dcalc@maxoftwo}[3]{%
2806 \ifnum #2>#1\relax
2807 #3=#2\relax
2808 \else
2809 #3=#1\relax
2810 \fi}
2811 \newcommand*\l@dcalc@minoftwo}[3]{%
2812 \ifnum #2<#1\relax
2813 #3=#2\relax
2814 \else
2815 #3=#1\relax
2816 \fi}
2817
2818 %

```

## XVII.7 Page break managing

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

number of lines have been output then both `\ifl@dpagfull` and `\ifl@dsamepage` are set FALSE.

```

2819 \newif\ifl@dsamepage
2820 \l@dsamepagetrue
2821 \newif\ifl@dpagfull
2822
2823 \newcommand*{\checkpageL}{%
2824 \l@dpagfulltrue
2825 \l@dsamepagetrue
2826 \check@goal
2827 \ifdim\pagetotal<\ledthegoal
2828 \ifnum\numpagelinesL<\l@dmnpagelines
2829 \else
2830 \ifnomaxlines%
2831 \else%
2832 \l@dsamepagefalse%
2833 \l@dpagfullfalse%
2834 \fi%
2835 \fi
2836 \else
2837 \l@dsamepagefalse
2838 \l@dpagfulltrue
2839 \fi%
2840 \ifprint@last@after@pendL%
2841 \l@dpagfullfalse%
2842 \l@dsamepagefalse%
2843 \print@last@after@pendLfalse%
2844 \fi%
2845 }%
2846
2847 \newcommand*{\checkpageR}{%
2848 \l@dpagfulltrue
2849 \l@dsamepagetrue
2850 \check@goal
2851 \ifdim\pagetotal<\ledthegoal
2852 \ifnum\numpagelinesR<\l@dmnpagelines
2853 \else
2854 \ifnomaxlines%
2855 \else%
2856 \l@dsamepagefalse%
2857 \l@dpagfullfalse%
2858 \fi%
2859 \fi
2860 \else
2861 \l@dsamepagefalse
2862 \l@dpagfulltrue
2863 \fi%
2864 \ifprint@last@after@pendR%
2865 \l@dpagfullfalse%
```

```

2866 \l@dsamepagefalse%
2867 \print@last@after@pendRfalse%
2868 \fi%
2869 }%
2870
2871 %

```

**\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.

```

2872 \newcommand{\checkpbL}{
2873   \IfStrEq{\led@pb@setting}{after}{
2874     \xifinlistcs{the\absline@num}{l@prev@pb}{\l@dpagetrue\
l@dsamepagefalse}{
2875       \xifinlistcs{the\absline@num}{l@prev@nopb}{\l@dpagetrue\
l@dsamepagetrue}{
2876         }{}
2877       \IfStrEq{\led@pb@setting}{before}{
2878         \numdef{next@absline}{the\absline@num+1}
2879         \xifinlistcs{next@absline}{l@prev@pb}{\l@dpagetrue\
l@dsamepagefalse}{
2880           \xifinlistcs{next@absline}{l@prev@nopb}{\l@dpagetrue\
l@dsamepagetrue}{
2881             }{}
2882           }
2883         }
2884       \newcommand{\checkpbR}{
2885         \IfStrEq{\led@pb@setting}{after}{
2886           \xifinlistcs{the\absline@numR}{l@prev@pbR}{\l@dpagetrue\
l@dsamepagefalse}{
2887             \xifinlistcs{the\absline@numR}{l@prev@nopbR}{\l@dpagetrue\
l@dsamepagetrue}{
2888               }{}
2889             \IfStrEq{\led@pb@setting}{before}{
2890               \numdef{next@abslineR}{the\absline@numR+1}
2891               \xifinlistcs{next@abslineR}{l@prev@pbR}{\l@dpagetrue\
l@dsamepagefalse}{
2892                 \xifinlistcs{next@abslineR}{l@prev@nopbR}{\l@dpagetrue\
l@dsamepagetrue}{
2893                   }{}
2894                 }
2895               }

```

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

```

2896 \newcommand{\checkverseL}{
2897   \ifinstanzaL
2898     \iflednopbinverse
2899     \ifinserthangingsymbol

```

```

2900     \numgdef{\prev@abslineverse}{\the\absline@num-1}
2901     \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{}
2902     \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesL<3\ledpbnum{\
prev@abslineverse}\fi}{}
2903     \fi
2904     \fi
2905 \fi
2906 }
2907 \newcommand{\checkverseR}{
2908 \ifinstanzaR
2909     \iflednopbinverse
2910     \ifinserthangingsymbolR
2911     \numgdef{\prev@abslineverse}{\the\absline@numR-1}
2912     \IfStrEq{\led@pb@setting}{after}{\lednopbnumR{\prev@abslineverse}}{}
2913     \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesR<3\ledpbnumR{\
prev@abslineverse}\fi}{}
2914     \fi
2915     \fi
2916 \fi
2917 }
2918 %

```

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

```

\check@goal
2919 \newdimen\ledthegoal
2920 \ifshiftedpstarts
2921     \newcommand*{\@goalfraction}{0.95}
2922 \else
2923     \newcommand*{\@goalfraction}{0.9}
2924 \fi
2925
2926 \newcommand*{\check@goal}{%
2927     \ledthegoal=\@goalfraction\pagegoal}
2928 \newcommand{\setgoalfraction}[1]{%
2929     \xdef\@goalfraction{#1}%
2930 }
2931 %

```

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

```

\ifwrittenlinesL
2932 \newif\ifwrittenlinesL
2933 \newif\ifwrittenlinesR
2934
2935 %

```

## XVII.8 Getting boxes content

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

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

```
2936 \newif\if@getnextbox%
2937 %
```

`\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.

```
2938 \newcommand*{\get@nextboxL}{%
2939   \ifvbox\namebox{1@dLcolrawbox\the\l@dpscl}% box is not empty
2940 %
```

The current box is not empty; do nothing.

```
2941   \else%                               box is empty
2942 %
```

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

```
2943   \@getnextboxtrue%Should be local, but be cautious
2944 %
```

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

```
2945   \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscl}>\@donetotallinesL
2946   \parledgroup@notes@endL%
2947   \unless\ifnosyncpstarts%
2948   \getnextboxfalse%
2949 %
```

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.

```
2950   \ifnomaxlines%
2951   \ifdim\pagetotal<\ledthegoal%
2952   \numdef{\@tmp}{\l@dpscl+1}%
2953   \ifcsdef{afterlines@pstart@\@tmp R}{%
2954   \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp R}}
2955   %
2956   {}%
2957   {\ifcsdef{minpage@pstart@\@tmp}%
    {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
    %
```

```

2958         {\ifnum\numpagelinesL=\l@minpagelines%
2959             \@getnextboxtrue%
2960             \fi%
2961         }%
2962         {\@getnextboxtrue}}%
2963         {\@getnextboxtrue}%
2964     }%
2965 }%
2966 {}%
2967 \fi%
2968 \fi%
2969 \fi%
2970 \else%
2971     \ifnomaxlines%
2972         \numdef{\@tmp}{\the\l@dpscl+1}%
2973         \ifcsdef{minpage@pstart@\@tmp}{%
2974             \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2975                 {\ifdimgreater{\pagetotal}{\ledthegoal}%
2976                     {\@getnextboxtrue}%
2977                     {\@getnextboxfalse}%
2978                 }%
2979                 {\@getnextboxtrue}%
2980             }{}%
2981         \fi%
2982     \fi%
2983 %

```

Sufficient lines have been output.

```

2984 \if@getnextbox%
2985     \ifnum\usenamecount{\l@maxlinesinpar\the\l@dpscl}=\@donetotallinesL
2986         \parledgroup@notes@endL
2987     \fi
2988     \ifwrittenlinesL\else
2989 %

```

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

```

2990         \@writelinesinparL
2991         \writtenlinesLtrue
2992     \fi
2993     \ifnum\l@dnumstartsL>\l@dpscl
2994 %

```

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.

```

2995         \writtenlinesLfalse
2996         \ifbypstart@
2997             \global\line@num=0%
2998             \resetprevline@%

```



```

2999         \fi
3000 % Add the content of the optional argument of the previous \protect\cs{pend
    }.
3001 % \begin{macrocode}
3002         \csuse{after@pendL@the\l@dpscl}%
3003         \global\csundef{after@pendL@the\l@dpscl}%
3004 %

Check the number of lines

3005         \l@dcalc@maxoftwo{\the\usernamecount{l@dmaxlinesinpar\the\l@dpscl}}%
3006                             {\the\@donetotallinesL}%
3007                             {\usernamecount{l@dmaxlinesinpar\the\l@dpscl}}}%
3008         \global\@donetotallinesL \z@
3009 %

Go to the next pstart

3010         \global\advance\l@dpscl \@ne
3011         \global\pstartnumtrue%
3012         \restore@pstartL@pc%
3013 %

Add notes of parallel ledgroup.

3014         \parledgroup@notes@endL
3015         \parledgroup@correction@notes@spacing@final{L}
3016     \else
3017 %

3018         \print@last@after@pendLtrue%
3019     \fi
3020 \fi
3021 \fi}
3022 %

3023 \newcommand*{\get@nextboxR}{%
3024     \ifvbox\namebox{l@dRcolrawbox\the\l@dpscl}% box is not empty
3025 \else% box is empty
3026     \@getnextboxtrue%
3027     \ifnum\usernamecount{l@dmaxlinesinpar\the\l@dpscl}>\@donetotallinesR
3028         \parledgroup@notes@endR
3029     \unless\ifnosyncpstarts%
3030         \@getnextboxfalse%
3031     \ifnomaxlines%
3032         \ifdim\pagetotal<\ledthegoal%
3033             \numdef{\@tmp}{\l@dpscl+1}%
3034             \ifcsdef{afterlines@pstart@\@tmp L}{%
3035                 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp L}}
3036             }%
3037             {\ifcsdef{minpage@pstart@\@tmp}%

```

```

3038         {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
3039         {\ifnum\numpagelinesR=\l@dminpagelines%
3040           \getnextboxtrue%
3041           \fi%
3042         }%
3043         {\@getnextboxtrue}}%
3044         {\@getnextboxtrue}%
3045       }%%
3046     }%
3047   {}%
3048   \fi%
3049 \fi%
3050 \fi%
3051 \else%
3052   \ifnomaxlines%
3053     \numdef{\@tmp}{\the\l@dpscR+1}%
3054     \ifcsdef{minpage@pstart@\@tmp}{%
3055       \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3056       {\ifdimgreater{\pagetotal}{\ledthegoal}%
3057         {\@getnextboxtrue}%
3058         {\@getnextboxfalse}%
3059       }%
3060       {\@getnextboxtrue}%
3061     }{}
3062   \fi%
3063 \fi%
3064 \if@getnextbox%
3065   \ifnum\usenamecount{l@dmaxlinesinpar\the\l@dpscR}=\@donetotallinesR
3066     \parledgroup@notes@endR
3067   \fi
3068   \ifwrittenlinesR\else
3069     \@writelinesinparR
3070     \writtenlinesRtrue
3071   \fi
3072   \ifnum\l@dnumpstartsR>\l@dpscR
3073     \writtenlinesRfalse
3074     \ifbypstart@R
3075       \global\line@numR=0%
3076       \resetprevline@%
3077     \fi
3078     \csuse{after@pendR@\the\l@dpscR}%
3079     \global\csundef{after@pendR@\the\l@dpscR}%
3080     \l@dcalc@maxoftwo{\the\usenamecount{l@dmaxlinesinpar\the\l@dpscR}}%
3081       {\the\@donetotallinesR}%
3082       {\usenamecount{l@dmaxlinesinpar\the\l@dpscR}}%
3083     \global\@donetotallinesR \z@
3084     \global\advance\l@dpscR \@ne
3085     \global\pstartnumRtrue%
3086     \restore@pstartR@pc%
3087     \parledgroup@notes@endR

```

```

3088     \parledgroup@correction@notespacing@final{R}
3089     \else
3090         \print@last@after@pendRtrue%
3091     \fi
3092 \fi
3093 \fi}
3094
3095 %

```

## XVIII Page numbering

### XVIII.1 Global options

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

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

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

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

```

3096
3097 \newcommand{\par@patch@thepage}{%
3098     \ifboolexpr{%
3099         bool{sameparallelpagewater}%
3100         or bool{prevpgnotnumbered}%
3101     }%
3102     {%
3103         \patchcmd{\thepage}%
3104             {page}{par@page}%
3105             {}{}%
3106         {\led@error@fail@patch@thepage}%
3107     }{}%
3108 }%

```

```

3109 \newcommand{\par@patch@pagenumbering}{%
3110   \ifboolexpr{%
3111     bool{sameparallelpagnumber}%
3112     or bool{prevpgnotnumbered}%
3113   }%
3114   {%
3115     \setcounter{par@page}{1}%
3116   }%
3117   {%
3118     \par@patch@thepage%
3119   }%
3120 }%
3121
3122 \ifl@dmemoir%
3123   \apptocmd{\@mempnum}%
3124     {\par@patch@pagenumbering}%
3125     {}%
3126     {\led@error@fail@patch@@mempnum}%
3127
3128 \else%
3129   \apptocmd{\pagenumbering}%
3130     {\par@patch@pagenumbering}%
3131     {}%
3132     {\led@error@fail@patch@pagenumbering}%
3133 \fi%
3134
3135 \AtBeginDocument{\par@patch@thepage}%
3136 %

```

**\@outputpage** As its name says, \@outputpage is a L<sup>A</sup>T<sub>E</sub>X's macro called in the output routine. It is this macro which increases the page counter.. We patch it in order to increase, conditionally, the par@page counter.

```

3137 \AtBeginDocument{%
3138   \apptocmd{\@outputpage}{%
3139     \ifsameparallelpagnumber%
3140       \ifl@dprintingpages%
3141         \ifodd\c@page\else%
3142           \stepcounter{par@page}%
3143         \fi%
3144       \else%
3145         \stepcounter{par@page}%
3146       \fi%
3147     \else%
3148       \stepcounter{par@page}%
3149     \fi%
3150   }%
3151   {}%
3152   {\led@error@fail@patch@@outputpage}%
3153 }

```

```
3154 %
```

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

```
3155 \newcounter{par@page}%
3156 \setcounter{par@page}{1}%
3157 %
```

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

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

```
\ifPages@mainmatter58 \newif\ifPages@mainmatter
\Pages@mainmatter59 \newcommand{\Pages@mainmatter}{%
3160     \ifPages@mainmatter%
3161     \pagenumbering{arabic}%
3162     \addtocounter{page}{1}%
3163     \addtocounter{par@page}{-1}%
3164     \patchcmd{\thepage}{page}{par@page}{}{}%
3165     \fi%
3166 }
3167 %
```

## 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.

```
\chapter68 \pretocmd{\chapter}{%
3169     \ifl@drprintingpages%
3170     \vbox{}%
3171     \fi%
3172 }%
3173 {}%
3174 {}%
3175 %
```

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

```
3176 \newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}}
3177 \eledsectnotoc{R}
3178 %
```

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

```
3179 \newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}}
3180 \eledsectmark{L}
3181 %
```

`\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`.

```
3182 \newskip\eledsection@correcting@skip
3183 %
```

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

```
3184 \newwrite\eled@sectioningR@out
3185 %
```

## 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`.

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

```
3186 \def\l@prev@pbR{}
3187 \def\l@prev@nopbR{}
3188 %
```

`\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.

```
3189 \newcommand{\ledpbR}{\write\linenum@outR{\string\led@pbR}}
3190 \newcommand{\ledpbnumR}[1]{\write\linenum@outR{\string\led@pbnumR{#1}}}
3191 \newcommand{\lednopbR}{\write\linenum@outR{\string\led@nopbR}}
3192 \newcommand{\lednopbnumR}[1]{\write\linenum@outR{\string\led@nopbnumR{#1}}}
3193 %
```

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

```

3194 \newcommand{\led@pbR}{\listxadd{\l@prev@pbR}{\the\absline@numR}}
3195 \newcommand{\led@pbnumR}[1]{\listxadd{\l@prev@pbR}{#1}}
3196 \newcommand{\led@nopbR}{\listxadd{\l@prev@nopbR}{\the\absline@numR}}
3197 \newcommand{\led@nopbnumR}[1]{\listxadd{\l@prev@nopbR}{#1}}
3198 %

```

## XXI Parallel ledgroup

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

```

3199 \newmarks\parledgroup@
3200 \newmarks\parledgroup@series
3201 \newmarks\parledgroup@type
3202 %

```

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

```

3203 \newcommand{\parledgroup@notes@startL}{%
3204   \ifnum\usernamecount{1@dmxlinesinpar\the\l@dpscL}>0%
3205     \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
3206       bhooknoteX@\splitfirstmarks\parledgroup@series}}{}%
3207     \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
3208       bhookXnote@\splitfirstmarks\parledgroup@series}}{}%
3209     \fi%
3210     \global\ledgroupnotesL@true%
3211     \insert@noterule@ledgroup{L}%
3212 }
3213 \newcommand{\parledgroup@notes@startR}{%
3214   \ifnum\usernamecount{1@dmxlinesinpar\the\l@dpscR}>0%
3215     \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
3216       bhooknoteX@\splitfirstmarks\parledgroup@series}}{}%
3217     \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
3218       bhookXnote@\splitfirstmarks\parledgroup@series}}{}%
3219     \fi%
3220     \global\ledgroupnotesR@true%
3221     \insert@noterule@ledgroup{R}%
3222 }
3223 %

```

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

```

3220 \newcommand{\parledgroup@notes@endL}{%
3221   \global\ledgroupnotesL@false%

```

```

3222 }
3223 \newcommand{\parledgroup@notes@endR}{\%
3224   \global\ledgroupnotesR@false%
3225 }
3226 \%

```

`\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`

```

3227 \newcommand{\insert@noterule@ledgroup}[1]{
3228   \IfStrEq{\splitbotmarks\parledgroup@}{begin}{\%
3229     \IfStrEq{\splitbotmarks\parledgroup@type}{Xfootnote}{
3230       \csuse{ifledgroupnotes#1@}
3231       \vskip\skip\csuse{mp\splitbotmarks\parledgroup@series footins}
3232       \csuse{\splitbotmarks\parledgroup@series footnoterule}
3233     \fi
3234   }
3235   {}
3236   \IfStrEq{\splitbotmarks\parledgroup@type}{footnoteX}{
3237     \csuse{ifledgroupnotes#1@}
3238     \vskip\skip\csuse{mpfootins\splitbotmarks\parledgroup@series}
3239     \csuse{footnoterule\splitbotmarks\parledgroup@series}
3240     \fi
3241   }{}
3242 }
3243 {}
3244 }
3245 \%

```

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

```

3246 \newcommand{\setparledgroupnotespadding}[1]{\gdef\@parledgroupnotespadding
3247   {#1}}
3247 \newcommand{\@parledgroupnotespadding}{}
3248 \%

```

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

```

3249 \dimdef{\parledgroup@notespadding@correction}{0pt}
3250 \newcommand{\parledgroup@notespadding@set@correction}{\%
3251   {\@getfirstseries\csuse{Xnotefontsize@\@firstseries}}%We suppose all the
3252   series has the same footnote size setup
3253   \parledgroup@notespadding\dimgdef{\temp@spacing}{\baselineskip}}%
3254   \dimgdef{\parledgroup@notespadding@correction}{\baselineskip-\temp@spacing}
3255 }%
3254 }
3255 \%

```



`\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.

```

3256 \newcommand{\parledgroup@correction@notespacing@init}{
3257   \dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}
3258   \dimdef{\parledgroup@notespacing@correction@modulo}{0pt}
3259 }
3260 \parledgroup@correction@notespacing@init
3261 %

```

`\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.

```

3262 \newcommand{\parledgroup@correction@notespacing@final}[1]{
3263   \ifparledgroup
3264   \vspace{\parledgroup@notespacing@correction@accumulated}
3265   \parledgroup@correction@notespacing@init%
3266   \ifstrequal{#1}{L}{
3267     \numdef{\@checking}{\the\l@dpscl-1}
3268   }{
3269     \numdef{\@checking}{\the\l@dpscr-1}
3270   }
3271   \dimdef{\@beforenotes@current@diff}{\csuse{@parledgroup@beforenotes@
3272     \@checking L}-\csuse{@parledgroup@beforenotes@\@checking R}}%
3273   \ifstrequal{#1}{L}{
3274     {% Left
3275       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{-\
3276         \@beforenotes@current@diff}}%
3277     }%
3278     {% Right
3279       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{\
3280         \@beforenotes@current@diff}}%
3281     }%
3282   }
3283   \fi
3284 }
3285 %

```

`\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` 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.

```

3282 \newcommand{\parledgroup@correction@notespacing}[1]{%
3283   \csuse{ifledgroupnotes#1@}%
3284   \vspace{-\parledgroup@notespacing@correction}%
3285   \dimdef{\parledgroup@notespacing@correction@accumulated}{\
parledgroup@notespacing@correction@accumulated+
parledgroup@notespacing@correction}%
3286   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo+
parledgroup@notespacing@correction}%
3287   \ifdimless{\parledgroup@notespacing@correction@modulo}{\baselineskip
}{\advance\numpagelinesL -\@ne%
3288   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo-\baselineskip}%
3289   }% mean greater than equal
3290   \fi%
3291 }
3292 %

```

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

```

3293 \dimdef\parledgroup@beforenotesL{0pt}
3294 \dimdef\parledgroup@beforenotesR{0pt}
3295 %

```

`\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.

```

3296 \newcommand{\parledgroup@beforenotes@save}[1]{
3297   \ifparledgroup
3298     \csdimgdef{@parledgroup@beforenotes@\the\csuse{lednumpstarts#1}#1}{\
csuse{parledgroup@beforenotes#1}}
3299     \csdimgdef{parledgroup@beforenotes#1}{0pt}
3300   \fi
3301 }
3302 %

```

## XXII Compatibility with eledmac

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

```

3303 \ifeledmaccompat%
3304
3305
3306   \unless\ifnocritical@

```

```
3307 \let\onlyXside\Xonlyside
3308 \fi
3309 \fi
3310 %
```

## **XXIII The End**

</code>

## Appendix A Some things to do when changing version

### Appendix A.1 Migration to eledpar 1.4.3

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

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

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

See the following two examples:

With standard `\hangingsymbol`:

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

With the modification of the `hangingsymbol`:

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

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

As for migration from eledmac to reledmac:

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

#### Appendix A.2.1 Deprecated options

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

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

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

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

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

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

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

## Appendix A.3 Migration to reledpar 2.2.0

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

## Appendix A.4 Migration to reledpar 2.3.0

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

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

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

## Appendix A.5 Migration to reledpar 2.4.0

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

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

## Appendix A.6 Migration to reledpar 2.5.0

If you use either `\stanza` or *astanza* environment, please read Appendix A.12 p. 353.

## Appendix A.7 Migration to reledpar 2.6.0

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

## Appendix A.8 Migration to reledpar 2.6.1

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

## Appendix 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`.

## References

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

## Index

### Symbols

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

\@pstartstrue	1
\@ref	1
\@ref@reg@parseR	1
\@ref@regR	1
\@set	1
\@stopastanza	1
\@writelinesinparL	1
\@writelinesinparR	1
\@writelinesonpageL	1
\@writelinesonpageR	1
\@writepageofparL	1
\@writepageofparR	1
CLASSmemoir	115
COMMAND\+	50
COMMAND\@Rlineflag	79, 152
COMMAND\@adv	41, 151
COMMAND\@cs@linesinparL	106
COMMAND\@cs@linesonpageL	106
COMMAND\@eledsectionL	89
COMMAND\@eledsectionR	89
COMMAND\@eledsectmark	118
COMMAND\@eledsectnotoc	117
COMMAND\@goalfraction	12, 110
COMMAND\@l@dtmptcnta	73
COMMAND\@lab	40, 79, 151
COMMAND\@lopL	45, 100, 107
COMMAND\@lopR	45, 107
COMMAND\@mempnum	115
COMMAND\@namedef	28
COMMAND\@namuse	28
COMMAND\@nl	40, 47, 79, 151
COMMAND\@nl@regR	40
COMMAND\@outputpage	115, 116
COMMAND\@page	79
COMMAND\@par@sync@option	39
COMMAND\@parledgroupnotespacing	120
COMMAND\@pend	45
COMMAND\@pendR	45
COMMAND\@pstart	45
COMMAND\@pstartstrue	87
COMMAND\@ref	43, 44, 47, 151
COMMAND\@ref@later	44
COMMAND\@ref@reg@parseargR	44
COMMAND\@ref@regR	43
COMMAND\@set	41, 151
COMMAND\@sw	44
COMMAND\AtBeginPairs	8, 53, 149, 153
COMMAND\AtEveryPend	149–151
COMMAND\AtEveryPstart	2, 18, 59, 90, 149–151
COMMAND\AtEveryPstartCall	2, 18, 59, 150

COMMAND\AtEveryStanza	153
COMMAND\AtEveryStopStanza	153
COMMAND\Clear the right lines for \read@linelist	39
COMMAND\Columns	7, 8, 17, 18, 51, 89, 93, 146–148, 150, 152
COMMAND\Columns@print@after@pend	96
COMMAND\Columns@print@before@pstart	96
COMMAND\Lcolwidth	8, 10, 98
COMMAND\Leftsidehook	146
COMMAND\Leftsidehookend	146
COMMAND\MakePerPage	13
COMMAND\Pages	4, 7, 10, 12, 18, 50, 51, 70, 74, 76, 97, 98, 104, 115, 117, 120, 146, 148, 150–152
COMMAND\Pages@mainmatter	117
COMMAND\Pairs	50
COMMAND\Rcolwidth	8, 10, 98
COMMAND\Rightsidehook	146
COMMAND\Rightsidehookend	146
COMMAND\Rlineflag	124
COMMAND\Xendlineflag	15, 152
COMMAND\Xlineflag	15, 125, 152
COMMAND\Xmaxhnotes	13
COMMAND\Xnoteswidthliketwocolumns	9, 149
COMMAND\Xonlyside	14, 74, 125, 153
COMMAND\Xtxtbeforenotes	153
COMMAND\&	19
COMMAND\absline@numR	38
COMMAND\add@penalties	72
COMMAND\add@penaltiesL	72
COMMAND\advanceline	41, 47, 151
COMMAND\affixline@num	68
COMMAND\affixline@numR	68, 146, 147
COMMAND\affixpstart@num	70
COMMAND\affixpstart@numR	70
COMMAND\affixside@note	80
COMMAND\aftercolumnseparator	9, 93, 148
COMMAND\araw@textfalse	88
COMMAND\araw@texttrue	88
COMMAND\at@begin@pairs	53
COMMAND\autopar	18
COMMAND\ballast@count	73
COMMAND\baselineskip	93, 121
COMMAND\bbl@set@language	84, 85, 151
COMMAND\beforecolumnseparator	9, 93, 148
COMMAND\begin	19
COMMAND\beginnumbering	15, 16, 18, 29, 40, 93, 147, 148, 150
COMMAND\beginnumberingR	33, 46
COMMAND\bf	147
COMMAND\bfseries	147
COMMAND\brokenpenalty	72
COMMAND\chapter	117, 146
COMMAND\check@goal	110



COMMAND\check@pstarts	87
COMMAND\checkpageL	100, 107
COMMAND\checkpb@columns	92
COMMAND\checkpbL	109
COMMAND\checkpbR	109
COMMAND\checkraw@text	88
COMMAND\checkverseL	109
COMMAND\checkverseR	109
COMMAND\clear(double)page	105
COMMAND\clearl@dleftpage	105
COMMAND\clearl@drighpage	105
COMMAND\clearpage	105, 150
COMMAND\cleartoevenpage	105
COMMAND\cleartol@devenpage	105, 117
COMMAND\columnrulewidth	8, 93
COMMAND\columns@position	93
COMMAND\columnseparator	8, 9
COMMAND\columnspoint	9, 148
COMMAND\correct@Xfootins@box	150
COMMAND\correct@footinsX@box	150
COMMAND\critext	149
COMMAND\curname	52
COMMAND\displaywidowpenalty	72
COMMAND\do@actions	66
COMMAND\do@actions@fixedcode	146
COMMAND\do@actions@nextR	66
COMMAND\do@actionsR	66, 146
COMMAND\do@ballast	73
COMMAND\do@ballastR	66
COMMAND\do@insidelineLhook	148
COMMAND\do@insidelineRhook	148
COMMAND\do@line	60
COMMAND\do@line(L/R)	63
COMMAND\do@lineL	60, 72, 146, 147
COMMAND\do@lineLhook	146
COMMAND\do@lineR	63, 146–148
COMMAND\do@lineRhook	146
COMMAND\do@lockoff	151
COMMAND\do@lockoffR	42
COMMAND\do@lockon	151
COMMAND\do@lockonR	41
COMMAND\doinsidelineLhook	149
COMMAND\doinsidelineRhook	149
COMMAND\dolineLhook	149
COMMAND\dolineRhook	149
COMMAND\edindex	149
COMMAND\edlabel	147, 149
COMMAND\edtext	15, 43, 47, 77, 149, 153
COMMAND\edtext@later	77, 79, 104
COMMAND\edtext@now	104

COMMAND\edtextlater	15, 77, 78, 153
COMMAND\edtextnow	15, 77–79, 153
COMMAND\eled@sectioningR@out	118
COMMAND\eledchapter	150
COMMAND\eledsection	149–151
COMMAND\eledsection@correcting@skip	118
COMMAND\eledsectmark	21, 118
COMMAND\eledsectnotoc	21, 117
COMMAND\eledxxx	149
COMMAND\end	19
COMMAND\endgraf	58
COMMAND\endlock	47, 151
COMMAND\endnumbering	9, 15, 18, 30, 150
COMMAND\endsub	47, 151
COMMAND\endumbering	16
COMMAND\expandafter	49
COMMAND\extensionchars	29
COMMAND\firstlinenum	16, 148, 151, 152
COMMAND\firstsublinenum	148, 151, 152
COMMAND\fix@page	41, 151
COMMAND\flag@end	47, 146, 149
COMMAND\flag@start	47, 149
COMMAND\flush@notesR	73
COMMAND\footnote	51
COMMAND\footnoteX	48, 49
COMMAND\footnoteX@reading	152
COMMAND\footnoteXmk	14
COMMAND\footnoteXnomk	14, 49
COMMAND\frontmatter	12, 21
COMMAND\get@nextboxL	147
COMMAND\get@nextboxR	147
COMMAND\getline@numL	65
COMMAND\getline@numR	65
COMMAND\getlinesfrompagelistL	106
COMMAND\getlinesfrompagelistR	106
COMMAND\getlinesfromparlistL	106
COMMAND\getlinesfromparlistR	106
COMMAND\gl@p	49
COMMAND\goalfraction	124
COMMAND\hangingsymbol	124, 146
COMMAND\hfill	92, 93
COMMAND\hidenumbering	16, 151
COMMAND\if@getnextbox	111
COMMAND\ifPages@mainmatter	117
COMMAND\ifbypage@	151
COMMAND\ifbypstart@R	151
COMMAND\ifcsboxvoid	28
COMMAND\ifdim	92
COMMAND\ifinserthangingsymbol	81
COMMAND\ifinserthangingsymbolR	81

COMMAND\ifl@dpagfull	107, 108
COMMAND\ifl@dpaging	25, 149
COMMAND\ifl@dpairing	25, 146
COMMAND\ifl@dsamelang	149
COMMAND\ifl@dsamepage	107, 108
COMMAND\ifl@pagefull	100
COMMAND\ifledRcol	25
COMMAND\iflinenumberLevenifblank	37
COMMAND\iflinenumberRevenifblank	37
COMMAND\iflledRcol	147
COMMAND\ifnumberedpar@	55
COMMAND\ifnumberingR	147
COMMAND\ifnumberpstart	51
COMMAND\ifpst@rtedL	29, 30, 56, 146
COMMAND\ifpst@rtedR	29
COMMAND\ifsublines@	41
COMMAND\insert@countR	43
COMMAND\insert@noterule@ledgroup	120
COMMAND\insert@notes@for@onlyside	76
COMMAND\insertlines@list	43
COMMAND\insertlines@listR	43
COMMAND\inserts@list	55
COMMAND\inserts@listR	72
COMMAND\l@d@nums	47
COMMAND\l@d@set	41, 47, 151
COMMAND\l@dLcolrawbox	55
COMMAND\l@dLcolrawbox1	86
COMMAND\l@dLcolrawbox2	86
COMMAND\l@dRcolrawbox	55
COMMAND\l@dcalc@maxoftwo	107
COMMAND\l@dcalc@minoftwo	107
COMMAND\l@dchecklang	146, 148
COMMAND\l@dcsnote	148
COMMAND\l@dleftbox	60, 100, 150
COMMAND\l@dlinenumR	37, 146
COMMAND\l@dlsnote	148
COMMAND\l@dmake@labels	79
COMMAND\l@dmaxlinesinpar	99
COMMAND\l@dmaxlinesinpar1	87
COMMAND\l@dminpagelines	99, 146
COMMAND\l@dnumpstartsL	86, 146
COMMAND\l@dprintingcolumnstrue	149
COMMAND\l@dprintingpagestrue	149
COMMAND\l@dpscL	90, 99, 112
COMMAND\l@dpscR	90, 99
COMMAND\l@drsnote	148
COMMAND\l@dsetupmaxlinecounts	87
COMMAND\l@duselanguage	84, 85, 146
COMMAND\l@dzeromaxlinecounts	87
COMMAND\l@prev@nopbR	118

COMMAND\l@prev@pbR	118
COMMAND\labelstarttrue	147
COMMAND\labelref@list	79
COMMAND\labelref@listR	79
COMMAND\lang	84
COMMAND\last@page@numR	41
COMMAND\led	147
COMMAND\led@nopbR	118
COMMAND\led@nopbnumR	118
COMMAND\led@pbR	118
COMMAND\led@pbnumR	118
COMMAND\ledinnerrote	19
COMMAND\ledleftnote	19
COMMAND\lednopb	19, 92, 109
COMMAND\lednopbR	118
COMMAND\lednopbnumR	118
COMMAND\ledouterote	19
COMMAND\ledpb	92, 109
COMMAND\ledpbR	118
COMMAND\ledpbnumR	118
COMMAND\ledrightnote	19
COMMAND\ledsidenote	19
COMMAND\ledstrutL	146
COMMAND\ledstrutR	146, 152
COMMAND\ledthegoal	110
COMMAND\ledtrutL	146, 152
COMMAND\leftlinenumR	37, 146
COMMAND\lemma	78
COMMAND\let	49
COMMAND\line@list@R	43
COMMAND\line@list@stuff	40, 46
COMMAND\line@margin	34
COMMAND\line@marginR	34, 146
COMMAND\line@numR	38
COMMAND\lineation	17, 150
COMMAND\lineation*	17, 34, 149
COMMAND\lineationR	17, 33, 150, 153
COMMAND\linenum	78
COMMAND\linenum@out	79
COMMAND\linenum@outR	46
COMMAND\linenumberLevenifblanktrue	17, 153
COMMAND\linenumberRevenifblank	153
COMMAND\linenumberRevenifblanktrue	17
COMMAND\linenumberlist	17, 126, 153
COMMAND\linenumberlistR	17, 126, 153
COMMAND\linenumberstyle	17, 125
COMMAND\linenumberstyle*	125
COMMAND\linenumberstyleR	17, 125
COMMAND\linenumincrement	16, 148, 151, 152
COMMAND\linenummargin	17, 34, 146, 151–153

COMMAND\linenummargin*	17, 34, 152
COMMAND\linenummarginColumns	17, 35
COMMAND\linenummarginColumns*	17
COMMAND\linenummarginColumnsR	17
COMMAND\linenummarginR	17, 34, 152
COMMAND\linenumrepR	37, 146
COMMAND\linesinpar@listL	45, 106
COMMAND\linesonpage@listL	45, 106
COMMAND\lock@off	42
COMMAND\lock@on	41
COMMAND\mainmatter	2, 12, 21, 152
COMMAND\makeatletter	63
COMMAND\maxchunks	7, 18, 86, 87
COMMAND\maxdimen	74, 75
COMMAND\maxhnotesX	13
COMMAND\memorydump	16, 32
COMMAND\n@num	150
COMMAND\new@lineL	47
COMMAND\new@lineR	47
COMMAND\newhookcommand@series	51
COMMAND\newif	150
COMMAND\newpage	104, 105, 150
COMMAND\newseries	51
COMMAND\newseries@	48
COMMAND\newseries@par	48, 50, 51
COMMAND\noeledxxx	149
COMMAND\nomark@	49
COMMAND\nomaxlines	45
COMMAND\notesXwidthliketwocolumns	9, 149
COMMAND\num@lines	72
COMMAND\num@lines(R)	55
COMMAND\numberingR	31
COMMAND\numberlinefalse	7
COMMAND\numberonlyfirstinline	147
COMMAND\numberpstartfalse	16
COMMAND\numberpstarttrue	16, 147, 151
COMMAND\one@line	55, 81
COMMAND\one@lineR	55
COMMAND\onlyXside	125
COMMAND\onlysideX	14, 74, 76, 150, 153
COMMAND\otherlanguage	151
COMMAND\page@action	41, 151
COMMAND\pagenumbering	115, 151
COMMAND\pages	12
COMMAND\pagetotal	100, 150
COMMAND\par@line	72
COMMAND\par@line(R)	55
COMMAND\par@patch@pagenumbering	115
COMMAND\par@patch@thepage	115
COMMAND\par@sync@option	23

COMMAND\parledgroup@	119
COMMAND\parledgroup@beforenotes@save	122
COMMAND\parledgroup@beforenotesL	122
COMMAND\parledgroup@beforenotesR	122
COMMAND\parledgroup@correction@notespacing	121
COMMAND\parledgroup@correction@notespacing@final	121
COMMAND\parledgroup@correction@notespacing@init	121
COMMAND\parledgroup@notes@endL	119
COMMAND\parledgroup@notes@endR	119
COMMAND\parledgroup@notes@startL	119
COMMAND\parledgroup@notes@startR	119
COMMAND\parledgroup@notespacing@correction	120, 121
COMMAND\parledgroup@notespacing@correction@accumulated	121
COMMAND\parledgroup@notespacing@correction@modulo	121
COMMAND\parledgroup@notespacing@set@correction	120
COMMAND\parledgroup@series	119
COMMAND\parledgroup@type	119
COMMAND\parledgroupnotespacing	124
COMMAND\parledgrouptrue	20
COMMAND\patchcmd	151
COMMAND\pausenumbering	9, 31, 32, 153
COMMAND\pend	3, 7, 10, 17–20, 51, 55, 56, 58, 59, 86, 96, 148, 149, 151, 152
COMMAND\pendL	148
COMMAND\pendR	148
COMMAND\pends	18
COMMAND\perpage	13
COMMAND\prev@nopbR	118
COMMAND\prev@pbR	118
COMMAND\prevpgstyle	24
COMMAND\print@Xnotes	74
COMMAND\print@Xnotes@forpages	74, 75, 150
COMMAND\print@columnseparator	92, 148
COMMAND\print@eledsectionL	62
COMMAND\print@line	62
COMMAND\print@lineL	62
COMMAND\print@notesX@forpages	150
COMMAND\printlinenumR	125
COMMAND\printlinesR	152
COMMAND\pstart	3, 7, 10, 16–20, 32, 47, 51, 55, 56, 58, 59, 86, 90, 96, 147, 148, 150–152
COMMAND\pstartL	59, 148
COMMAND\pstartR	59, 147, 148
COMMAND\pstartinfootnote	150
COMMAND\raw@text	86
COMMAND\read@linelist	39, 40, 128, 151
COMMAND\ref@reg	43
COMMAND\ref@regR	43, 151
COMMAND\relax	125
COMMAND\reledmac	151
COMMAND\renewcommand	124
COMMAND\resumenumbering	9, 31, 32, 148, 153

COMMAND\resumenumberingR	33, 149
COMMAND\rightlinenumR	37, 146
COMMAND\sameword	78
COMMAND\section	146
COMMAND\section@num	29
COMMAND\selectlanguage	18, 84, 85
COMMAND\set@continuousnumberingforR	33
COMMAND\set@line	47, 151
COMMAND\set@line@action	41, 151
COMMAND\setRlineflag	17, 124
COMMAND\setgoalfraction	12, 124
COMMAND\sethangingsymbol	19
COMMAND\setline	41, 47, 151
COMMAND\setlinenum	41, 47, 151
COMMAND\setnoteposition...	94
COMMAND\setparledgroupnotespacing	124, 151
COMMAND\setposition...	93
COMMAND\setstanzaindents	9, 19, 124
COMMAND\setwidth...	93
COMMAND\sidenotemargin	19, 148
COMMAND\sidenotemargin*	19, 148
COMMAND\skipnumbering	16, 150
COMMAND\sloppy	8
COMMAND\stanza	7, 9, 16, 18, 19, 53, 82, 125, 147
COMMAND\stanzanumtrue	19
COMMAND\startlock	47, 151
COMMAND\startsub	47, 151
COMMAND\sub@action	41, 151
COMMAND\sub@off	79
COMMAND\sub@on	79
COMMAND\subline@numR	38
COMMAND\sublinenumberstyle	17, 125
COMMAND\sublinenumberstyle*	125
COMMAND\sublinenumberstyleR	17, 125
COMMAND\sublinenumincrement	148, 151, 152
COMMAND\sublinenumrepR	37, 146
COMMAND\syntaxonly	152
COMMAND\sza@0@	19
COMMAND\textheight	13
COMMAND\textwidth	53
COMMAND\the@labelX	152
COMMAND\theledlanguageL	84, 85
COMMAND\theledlanguageR	84, 85
COMMAND\thepage	21, 115
COMMAND\thepstartL	16, 147
COMMAND\thepstartR	16, 147
COMMAND\thestanzaL	19
COMMAND\thestanzaR	19
COMMAND\vbox	57
COMMAND\vskip	120

COMMAND\vsplit	72
COMMAND\widthliketwocolumns	9
COMMAND\widthliketwocolumnsfalse	9
COMMAND\widthliketwocolumnstrue	9
COMMAND\xflagref	152
COMMAND\xright@appenditem	49
COMMAND\xspace	22
COMMAND\xxxfootstart	94
COMMAND\xxxmatter	115
ENVIRONMENTLeftside	54
ENVIRONMENTRightside	54
ENVIRONMENTastanza	19, 82, 125, 152
ENVIRONMENTcolumns	31, 152
ENVIRONMENTledgroup	6, 152
ENVIRONMENTleft	17
ENVIRONMENTpages	31, 52, 152
ENVIRONMENTpairs	52, 152
PACKAGEEDMAC	126
PACKAGEEDSTANZA	126
PACKAGEEledmac	150
PACKAGEEledpar	150
PACKAGETABMAC	126
PACKAGEbabel	18, 84, 85, 152
PACKAGEedmac	126
PACKAGEeledmac	4, 86, 122, 124, 126, 148, 151
PACKAGEeledpar	5, 6, 13, 36, 124, 125, 148, 149
PACKAGEetoolbox	28, 92, 118
PACKAGEledmac	6
PACKAGEledpar	1, 6
PACKAGEMemoir	126
PACKAGEMusixtex	148
PACKAGEperpage	2, 13, 125, 152
PACKAGEpolyglossia	18, 84, 85, 153
PACKAGEreledmac	1, 3, 5–7, 9, 13, 17–22, 24, 25, 29, 30, 34–36, 38, 40–43, 45–49, 51, 62, 70, 79, 81, 98, 104, 124, 125, 151–153
PACKAGEreledpar	1, 3, 5–7, 9–12, 19–24, 32, 38, 39, 45, 48, 51, 79, 124–126, 151
PACKAGEsetspace	2, 20
PACKAGESyntonly	152
PACKAGExkeyval	22
PACKAGEX	153

## A

\absline@numR	1
\actionlines@listR	1
\actions@listR	1
\add@inserts@nextR	1
\add@insertsR	1
\add@penaltiesL	1
\add@penaltiesR	1
\advanceline	1



\affixline@numR	1
\affixpstart@numL	1
\affixpstart@numR	1
\affixside@noteR	1
\aftercolumnseparator	1, 9
\araw@textfalse	1
\araw@texttrue	1
astanza (environment)	19
\AtBeginPairs	1, 8
\AtEveryPstartCall	1
\autopar	18

**B**

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

**C**

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

<code>\countLline</code> .....	1
<code>\countRline</code> .....	1
<code>\critext</code> .....	1

**D**

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

**E**

<code>\edlabel</code> .....	1
<code>\edtext</code> .....	1
<code>\edtext@later</code> .....	1
<code>\edtext@now</code> .....	1
<code>\edtextlater</code> .....	1, 15
<code>\edtextnow</code> .....	1, 15
<code>\eled@sectioningR@out</code> .....	1
<code>\eledsection@correcting@skip</code> .....	1
<code>\eledsectmark</code> .....	1, 21
<code>\eledsectnotoc</code> .....	1, 21
<code>\endlock</code> .....	1
<code>\endnumbering</code> .....	1, 15
<code>\endnumberingR</code> .....	1
<code>\endsub</code> .....	1
environments:	
<code>astanza</code> .....	19
<code>Leftside</code> .....	15
<code>pages</code> .....	9
<code>pairs</code> .....	8
<code>Rightside</code> .....	15

**F**

<code>\f@x@l@cksR</code> .....	1
--------------------------------	---

<code>\finish@Pages@notes</code> .....	1
<code>\first@linenum@out@Rfalse</code> .....	1
<code>\first@linenum@out@Rtrue</code> .....	1
<code>\firstlinenum</code> .....	1, 16
<code>\firstlinenum*</code> .....	1, 17
<code>\firstlinenumR</code> .....	1, 17
<code>\firstsublinenum</code> .....	1, 16
<code>\firstsublinenum*</code> .....	1, 17
<code>\firstsublinenumR</code> .....	1, 17
<code>\fix@page</code> .....	1
<code>\flag@end</code> .....	1
<code>\flag@start</code> .....	1
<code>\flush@notesR</code> .....	1
<code>\footnote@reading</code> .....	1
<code>\footnote@typeset</code> .....	1
<code>\footnoteXmk</code> .....	14
<code>\footnoteXnomk</code> .....	14

## G

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

## H

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

## I

<code>\if@getnextbox</code> .....	1
<code>\if@pstarts</code> .....	1
<code>\ifaraw@text</code> .....	1
<code>\ifcsboxvoid</code> .....	1
<code>\iffirst@linenum@out@R</code> .....	1
<code>\ifinstanzaL</code> .....	1
<code>\ifinstanzaR</code> .....	1
<code>\ifl@dpagfull</code> .....	1
<code>\ifl@dpaging</code> .....	1
<code>\ifl@dpairing</code> .....	1
<code>\ifl@dsamepage</code> .....	1
<code>\ifl@dusedbabel</code> .....	1
<code>\ifledRcol</code> .....	1
<code>\iflinenumberLevenifblank</code> .....	1
<code>\iflinenumberRevenifblank</code> .....	1
<code>\ifnomaxlines</code> .....	1
<code>\ifnosyncpstarts</code> .....	1

<code>\ifPages@mainmatter</code> .....	1
<code>\ifprevpgnotnumbered</code> .....	1
<code>\ifprint@last@after@pendL</code> .....	1
<code>\ifprint@last@after@pendR</code> .....	1
<code>\ifpst@rtedL</code> .....	1
<code>\ifpst@rtedR</code> .....	1
<code>\ifpstartnumR</code> .....	1
<code>\ifsameparallelpagenunder</code> .....	1
<code>\ifshiftedpstarts</code> .....	1
<code>\ifwidthliketwocolumns</code> .....	1
<code>\ifwrittenlinesL</code> .....	1
<code>\init@series@par</code> .....	1
<code>\initnumbering@sectcountR</code> .....	1
<code>\insert@countR</code> .....	1
<code>\insert@noterule@ledgroup</code> .....	1
<code>\insert@notes@for@onlyside</code> .....	1
<code>\inserthangingsymbolL</code> .....	1
<code>\inserthangingsymbolR</code> .....	1
<code>\insertlines@listR</code> .....	1
<code>\inserts@listR</code> .....	1

## L

<code>\l@d@set</code> .....	1
<code>\l@dc@maxchunks</code> .....	1
<code>\l@dcalc@maxoftwo</code> .....	1
<code>\l@dcalc@minoftwo</code> .....	1
<code>\l@dcalcnun</code> .....	1
<code>\l@dchecklang</code> .....	1
<code>\l@dleftbox</code> .....	1
<code>\l@dlinenumR</code> .....	1
<code>\l@dmake@labelsR</code> .....	1
<code>\l@dminpagelines</code> .....	1
<code>\l@dnumpstartsL</code> .....	1
<code>\l@dnumpstartsR</code> .....	1
<code>\l@dpagefullfalse</code> .....	1
<code>\l@dpagefulltrue</code> .....	1
<code>\l@drightbox</code> .....	1
<code>\l@dsamepagefalse</code> .....	1
<code>\l@dsamepagetrue</code> .....	1
<code>\l@dsetupmaxlinecounts</code> .....	1
<code>\l@dsetuprawboxes</code> .....	1
<code>\l@dskipversenumberR</code> .....	1
<code>\l@dusedbabelfalse</code> .....	1
<code>\l@dusedbabeltrue</code> .....	1
<code>\l@duselanguage</code> .....	1
<code>\l@dzeromaxlinecounts</code> .....	1
<code>\l@pscl</code> .....	1
<code>\l@pscr</code> .....	1
<code>\labelref@listR</code> .....	1
<code>\last@page@numR</code> .....	1

\Lcolwidth .....	1, 8, 10
\led@err@BadLeftRightPstarts .....	1
\led@err@Columns@InsideEnv .....	1
\led@err@Columns@WithoutEnv .....	1
\led@err@Columns@WithoutLeftside .....	1
\led@err@Columns@WithoutRightside .....	1
\led@err@LeftOnRightPage .....	1
\led@err@Leftside@PreviousNotPrinted .....	1
\led@err@Pages@InsideEnv .....	1
\led@err@Pages@WithoutEnv .....	1
\led@err@Pages@WithoutLeftside .....	1
\led@err@Pages@WithoutRightside .....	1
\led@err@RightOnLeftPage .....	1
\led@err@Rightside@PreviousNotPrinted .....	1
\led@err@TooManyPstarts .....	1
\led@error@edtext@later@now .....	1
\led@error@fail@patch@mempnum .....	1
\led@error@fail@patch@outputpage .....	1
\led@error@fail@patch@pagenumbering .....	1
\led@error@fail@patch@thepage .....	1
\led@error@missing@numbering .....	1
\led@error@note@called@onleftside .....	1
\led@error@note@called@onrightside .....	1
\led@nopbnumR .....	1
\led@nopbR .....	1
\led@pbnumR .....	1
\led@pbR .....	1
\led@warn@ChangeSyncOption .....	1
\led@warn@setting@in@rightside .....	1
\lednopbnum .....	1
\lednopbnumR .....	1
\ledpbnumR .....	1
\ledpbR .....	1
\ledstrutL .....	1
\ledstrutR .....	1
\ledthegoal .....	1
\leftlinenumR .....	1
\leftpstartnumL .....	1
\leftpstartnumR .....	1
Leftside (environment) .....	15
\Leftsidehook .....	1
\Leftsidehookend .....	1
\line@list@stuffR .....	1
\line@listR .....	1
\line@marginR .....	1
\line@numR .....	1
\lineation* .....	1, 17
\lineationR .....	1, 17
\linenum@outR .....	1
\linenumberLevenifblanktrue .....	17

<code>\linenumberRevenifblanktrue</code> .....	17
<code>\linenumberstyle*</code> .....	<u>1</u> , 17
<code>\linenumberstyleR</code> .....	<u>1</u> , 17
<code>\linenumincrement</code> .....	<u>1</u> , 16
<code>\linenumincrement*</code> .....	<u>1</u> , 17
<code>\linenumincrementR</code> .....	<u>1</u> , 17
<code>\linenummargin</code> .....	<u>1</u>
<code>\linenummargin*</code> .....	<u>1</u> , 17
<code>\linenummarginColumns</code> .....	<u>1</u> , 17
<code>\linenummarginColumns*</code> .....	<u>1</u> , 17
<code>\linenummarginColumnsR</code> .....	<u>1</u> , 17
<code>\linenummarginR</code> .....	<u>1</u> , 17
<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> , 7
<code>\maxlinesinpar@list</code> .....	<u>1</u>
<code>\memorydump</code> .....	16
<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>\notesXwidthliketwocolumns</code> .....	9
<code>\num@linesR</code> .....	<u>1</u>
<code>\numberpstartfalse</code> .....	16
<code>\numberpstarttrue</code> .....	16
<code>\numpagelinesL</code> .....	<u>1</u>
<code>\numpagelinesR</code> .....	<u>1</u>

## O

<code>\one@lineR</code> .....	<u>1</u>
<code>\onlysideX</code> .....	14
<code>optionadvancedshiftedpstarts</code> .....	10, 11
<code>optionauxdir</code> .....	153
<code>optioncontinuousnumberingwithcolumns</code> .....	9, 153
<code>optionnomaxlines</code> .....	10, 11, 23
<code>optionnosyncpstarts</code> .....	12, 23, 111

optionshiftedpstarts	6, 11, 23
optionwidthliketwocolumns	9

**P**

\page@action	1
\page@numR	1
\Pages	1, 10
pages (environment)	9
\Pages@mainmatter	1
pairs (environment)	8
\par@lineR	1
\par@patch@pagenumbering	1
\par@patch@thepage	1
\parledgroup@	1
\parledgroup@beforenotes@save	1
\parledgroup@beforenotesL	1
\parledgroup@beforenotesR	1
\parledgroup@correction@notespacing	1
\parledgroup@correction@notespacing@final	1
\parledgroup@correction@notespacing@init	1
\parledgroup@notes@startL	1
\parledgroup@notes@startR	1
\parledgroup@notespacing@correction	1
\parledgroup@notespacing@set@correction	1
\parledgroupseries@	1
\parledgrouptype@	1
\pausenumberingR	1
\pend	17
\pendL	1
\pendR	1
\prev@nopbR	1
\prev@pbR	1
\prevpgstyle	1
\print@columnseparator	1
\print@eledsectionL	1
\print@eledsectionR	1
\print@lineL	1
\print@lineR	1
\print@notesX@forpages	1
\print@Xnotes@forpages	1
\pstart	17
\pstartL	1
\pstartR	1

**R**

\Rcolwidth	1, 8, 10
\read@linelist	1
\reledpar@error	1
\reledpar@warning	1
\restore@pstartL@pc	1

<code>\restore@pstartR@pc</code> .....	1
<code>\resumenumberingR</code> .....	1
<code>\rightlinenumR</code> .....	1
<code>\rightpstartnumL</code> .....	1
<code>\rightpstartnumR</code> .....	1
<code>Rightside (environment)</code> .....	15
<code>\Rightsidehook</code> .....	1
<code>\Rightsidehookend</code> .....	1
<code>\Rlineflag</code> .....	1

## S

<code>\save@familiarfootnote@number</code> .....	1
<code>\save@section@number</code> .....	1
<code>\section@numR</code> .....	1
<code>\selectlanguage</code> .....	1
<code>\set@continuousnumberingforR</code> .....	1
<code>\set@line</code> .....	1
<code>\set@line@action</code> .....	1
<code>\set@sectcountR</code> .....	1
<code>\setgoalfraction</code> .....	1, 12
<code>\sethangingsymbol</code> .....	19
<code>\setline</code> .....	1
<code>\setlinenum</code> .....	1
<code>\setnamebox</code> .....	1
<code>\setnotepositionliketwocolumns@C</code> .....	1
<code>\setnotepositionliketwocolumns@L</code> .....	1
<code>\setnotepositionliketwocolumns@R</code> .....	1
<code>\setpositionliketwocolumns@C</code> .....	1
<code>\setpositionliketwocolumns@L</code> .....	1
<code>\setpositionliketwocolumns@R</code> .....	1
<code>\setRlineflag</code> .....	17
<code>\setwidthliketwocolumns@C</code> .....	1
<code>\setwidthliketwocolumns@L</code> .....	1
<code>\setwidthliketwocolumns@R</code> .....	1
<code>\sidenote@marginR</code> .....	1
<code>\sidenotemargin*</code> .....	1
<code>\skip@lockoff</code> .....	1
<code>\skipnumbering</code> .....	1, 16
<code>\startlock</code> .....	1
<code>\startsub</code> .....	1
<code>\sub@action</code> .....	1
<code>\subline@numR</code> .....	1
<code>\sublinenumberstyle*</code> .....	1, 17
<code>\sublinenumberstyleR</code> .....	1, 17
<code>\sublinenumincrement</code> .....	1, 16
<code>\sublinenumincrement*</code> .....	1, 17
<code>\sublinenumincrementR</code> .....	1, 17
<code>\sublinenumrepR</code> .....	1



## T

<code>\theledlanguageL</code> .....	<u>1</u>
<code>\theledlanguageR</code> .....	<u>1</u>
<code>\thepar@page</code> .....	<u>1</u>
<code>\thepstartL</code> .....	16
<code>\thepstartR</code> .....	16
<code>\thestanzaL</code> .....	<u>1</u> , 19
<code>\thestanzaR</code> .....	<u>1</u> , 19

## U

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

## W

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

## X

<code>\Xendlineflag</code> .....	15
<code>\Xlineflag</code> .....	15
<code>\Xnoteswidthliketwocolumns</code> .....	9
<code>\Xonlyside</code> .....	14

## Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
\Columns: Added \l@dchecklang and \l@duselanguage to \Columns	90
\Pages: Added \l@duselanguage to \Pages	100
General: Added section of babel related code	84
Fix babel problems	1
v0.3.0.	
\Pages: Added \ledstrutL to \Pages	100
Added \ledstrutR to \Pages	102
\Rightsidehookend: Added \Leftsidehook, \Leftsidehookend, \Rightsidehook and \Rightsidehookend	54
\affixline@numR: Changed \affixline@numR to match neweledmac	68
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR	66
\do@lineL: Added \do@lineLhook to \do@lineL	60
Simplified \do@lineL by using macros for some common code	60
\do@lineR: Changed \do@lineR similarly to \do@lineL	63
\flag@end: Removed extraneous spaces from \flag@end	47
\ifledRcol: Moved \ifl@dpairing toeledmac	25
\ifpst@rtedR: Moved \ifpst@rtedL toeledmac	29
\l@dlinenumR: Simplified \leftlinenumR and \rightlinenumR by introducing \l@dlinenumR	37
\l@dnumpstartsR: Moved \l@dnumpstartsL toeledmac	86
\ledstrutR: Added \ledtrutL and \ledstrutR	105
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR	37
General: Added \do@lineLhook and \do@lineRhook	63
Added hooks into Leftside environment	54
Reorganize for ledarab	1
v0.3.a.	
\line@marginR: Do not just set \line@marginR in \linenummargin	34
General: Minor \linenummargin fix	1
v0.3.b.	
\Pages: Added \l@dminpagelines calculation for succeeding page pairs	103
General: Improved parallel page balancing	1
v0.3.c.	
General: Compatibilty with Polyglossia	1
v0.4.0.	
General: No more ledparpatch. All patches are now in the main file.	1
v0.5.0.	
General: Corrections about \section and other titles in numbered sections	1
v0.6.0.	
General: Be able to use \chapter in parallel pages.	1
v0.7.0.	
General: Option ‘shiftedverses’ which make there is no blank between two parallel verses with unequal length.	1
v0.8.0.	
General: Possibility to have a symbol on each hanging of verses, like in the french typography. Redefine the commande \hangingsymbol to define the character.	1

v0.9.0.	
\iflledRcol: Moved \iflledRcol and \ifnumberingR to eledmac	25
General: Possibility to number \pstart.	16
Possibility to number the pstart with the commands \numberpstarttrue.	1
v0.9.1.	
General: The numbering of the pstarts restarts on each \beginnumbering.	1
v0.9.2.	
General: Debug : with \Columns, the hanging indentation now runs on the left columns and the hanging symbol is shown only when \stanza is used.	1
v0.9.3.	
General: \thepstartL and \thepstartR use now \bfseries and not \bf, which is deprecated and makes conflicts with memoir class.	1
v0.10.0.	
General: \edlabel commands on the right side are now correctly indicated.	1
\edlabel commands which start a paragraph are now put in the right place.	1
v0.11.0.	
\Columns: Line numbering by pstart.	91
\affixline@numR: Changed \affixline@numR to allow to disable line numbering (like in eledmac 0.15).	68
\get@nextboxR: Change \get@nextboxL and \get@nextboxR to allow to disable line numbering (like in eledmac 0.15).	111
Pstart number can be printed in side	112
\inserthangingsymbolR: Prevent the column separator for hanging verse from shifting	81
General: Change \do@lineL and \do@lineR to allow line numbering by pstart (like in eledmac 0.15).	60
Lineation can be by pstart (like in eledmac 0.15).	32
New management of hangingsymbol insertion, preventing undesirable insertions.	81
v0.12.0.	
General: New management of hangingsymbol insertion, preventing undesirable insertions.	81
v1.0.0.	
General: Compatibility with eledmac. Change name to eledpar.	1
Debug in lineation by pstart	32
v1.0.1.	
General: Correction on \numberonlyfirstinline with lineation by pstart or by page.	1
v1.1.0.	
\pstartR: Add \labelpstarttrue (from eledmac).	55
General: Shiftedverses becomes shiftedpstarts.	1
v1.1.1.	
\pstartR: Correct \pstartR bug introduced by 1.1.	55
v1.1.2.	
\affixside@noteR: Remove spurious space between line number and line content	80
v1.2.0.	
General: Support for \led<section> commands in parallel texts.	1
v1.2.1.	
\set@sectcountR: For the right section, the counter is defined only once.	31
v1.3.0.	
\edtext: Manage RTL language.	47
v1.3.2.	
General: Debug with some classes.	1

v1.3.3.	
General: Debugging the left notes of the right column. . . . .	80
v1.3.4.	
General: Allow use of commands in sidenotes, as introduced by eledmac 1.0. . . . .	80
v1.4.0.	
General: Added <code>\do@insidelineLhook</code> and <code>\do@insidelineRhook</code> . . . . .	63
v1.4.1.	
General: Enable the use of <code>stanzaindentsrepetition</code> within <code>astanza</code> environment. . . . .	82
v1.4.3.	
<code>\inserthangingsymbolR</code> : Hanging verse is no longer automatically flush right. . . . .	81
<code>\pendL</code> : Spurious spaces in <code>\pendL</code> . . . . .	58
<code>\pendR</code> : Spurious spaces in <code>\pstartR</code> . . . . .	59
<code>\pstartR</code> : Spurious spaces in <code>\pstartL</code> and <code>\pstartR</code> . . . . .	55
General: Corrects a false hanging verse when a verse is exactly the length of a line. . . . .	1
v1.5.0.	
<code>\sublinenumincrement*</code> : Add starred version of <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> to change both Left and Rightside. . . . .	36
General: Add, as in eledmac, features to manage page breaks. . . . .	1
v1.6.0.	
General: Add tool and documentation for parallel ledgroups . . . . .	20
v1.7.0.	
General: Add, as in eledmac, features to make crossrefs with <code>pstart</code> numbers. . . . .	1
v1.8.0.	
<code>\Columns</code> : Modify <code>\Columns</code> to enable to add section's title. . . . .	89
Suppress <code>\l@dchecklang</code> from <code>\Columns</code> . . . . .	90
<code>\Pages</code> : Modify <code>\Pages</code> to enable to add section's title. . . . .	97
<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. . . . .	84
<code>\pendL</code> : As in eledmac, <code>\pendL</code> can have an optional argument. . . . .	58
<code>\pendR</code> : As in eledmac, <code>\pendR</code> can have an optional argument. . . . .	59
<code>\print@columnseparator</code> : Move some code of <code>\Columns</code> to <code>\print@columnseparator</code> . . . . .	92
<code>\pstartR</code> : As in eledmac, <code>\pendL</code> and <code>\pendR</code> can have an optional argument. . . . .	55
<code>\sidenotemargin*</code> : <code>\sidenotemargin</code> is now directly defined in eledmac to be able to manage eledpar. . . . .	80
Add <code>\sidenotemargin*</code> . . . . .	80
<code>\theledlanguageR</code> : Correct left/right language setting with polyglossia. . . . .	85
General: <code>\beginnumbering</code> is defined only on eledmac, not on eledpar. . . . .	29
<code>\l@dlsnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in eledmac. . . . .	80
Add <code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code> . . . . .	9
Add <code>\columnspostion</code> . . . . .	9
Add, as in eledmac, new system of sectioning commands. . . . .	1
Add, as in eledmac, option to insert something after <code>\pends</code> / verses. . . . .	1
Add, as in eledmac, option to insert something between <code>\pstarts</code> / verse. . . . .	1
Change <code>\do@lineR</code> and <code>\do@lineR</code> to allow new sectioning commands. . . . .	60
Compatibility with <code>musixtex</code> . . . . .	1
Debug eledmac sectioning command after using <code>\resumenumbering</code> . . . . .	1
New sectioning commands, as in eledmac. . . . .	21

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.	84
v1.8.1.	
<code>\do@lineL</code> : Fix a bug with critical notes at the beginning of a page, (maybe added by v1.8.0) (?).	60
<code>\do@lineR</code> : Fix a bug with critical notes at the beginning of a page, added by v1.8.0 (?).	63
v1.8.2.	
<code>\flag@end</code> : <code>\flag@start</code> and <code>\flag@end</code> are now defined only one time for <code>eledmac</code> and <code>eledpar</code>	47
<code>\lineation*</code> : Add <code>\lineation*</code>	34
<code>\reledpar@error</code> : Errors specific to <code>eledpar</code> send to <code>eledpar</code> handbook	25
General: Debug <code>\eledxxx</code> with some paper sizes	1
Debug left and side note (bugs added by 1.8.0)	1
v1.8.3.	
<code>\Pages</code> : Debug blank pages when using optional argument in the last <code>\pend</code> .	97
<code>\doinsidelineRhook</code> : Added <code>\dolineLhook</code> , <code>\dolineRhook</code> , <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code>	63
<code>\resumenumberingR</code> : Debug <code>\resumenumberingR</code>	31
General: Add <code>\noeledxxx</code> , as in <code>eledmac</code>	1
v1.9.0.	
<code>\ifwidthliketwocolumns</code> : Added <code>widthliketwocolumns</code> option	24
<code>\theledlanguageR</code> : Debug left/right language switching with <code>polyglossia</code> . Do not write in <code>.aux</code> file when setting left/right lines.	85
General: Add <code>\AtBeginPairs</code> macro.	8
Compatibility with <code>\Xnoteswidthliketwocolumns</code> and <code>\notesXwidthliketwocolumns</code>	1
v1.9.1.	
<code>\ifledRcol</code> : Moved <code>\ifl@dpaging</code> to <code>eledmac</code>	25
v1.10.0.	
<code>\Pages</code> : Debug wrong pages splitting when no optional argument is used in last <code>\pend</code> (bug was added in v1.8.3).	97
Debug wrong parallel pages synchronization when an <code>\edtext</code> falls across two pages.	97
General: Compatibility with <code>\AtEveryPstart</code> and <code>\AtEveryPend</code>	1
Restore critical notes in <code>\eledsection</code> in parallel columns (this bug was added in 1.8.2).	1
v1.10.1.	
<code>\line@list@stuffR</code> : Revert modification of 1.4.2, which makes bugs with numbering. Leave vertical mode to solve spurious space before <code>minipage</code> .	46
v1.11.0.	
<code>\edtext</code> : <code>\critext</code> and <code>\edtext</code> are now defined only in <code>eledmac</code> .	47
General: Compatibility of standard footnotes with some <code>biblatex</code> styles.	1
v1.12.0.	
<code>\Columns</code> : Add <code>\l@dprintingcolumnstrue</code>	89
<code>\Pages</code> : Add <code>\l@dprintingpagestrue</code>	97
<code>\edlabel</code> : <code>\edlabel</code> and <code>\edindex</code> works now with <code>hyperref</code> when using <code>eledpar</code> .	79
<code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code>	79
<code>\print@eledsectionL</code> : Compatibility with <code>Lua<math>\TeX</math></code> RTL languages.	62
<code>\print@eledsectionR</code> : Compatibility with <code>Lua<math>\TeX</math></code> RTL languages.	65
<code>\print@lineL</code> : Compatibility with <code>Lua<math>\TeX</math></code> RTL languages.	62
General: Compatibility with <code>Lua<math>\TeX</math></code> RTL languages.	1

v1.12.1.	
\print@eledsectionL: Fixes bug with Lua <sup>®</sup> TeX RTL \eledsection.	62
v1.13.0.	
\Pages: Prevent false overfull hboxes when using \Pages outside of pages environment.	98
When using shiftedpstarts option, a \l@leftbox with a null height will advance the	
\pagetotal in any case.	97
\clearl@drihtpage: Use \newpage instead of \clearpage.	105
\ifledRcol: Remove false boolean settings which are not needed.	25
General: Enable the use of optional argument of & in astanza environment.	82
Fix bug in shiftedpstarts when size difference between pstarts is very important.	1
With parallel pages, long notes can now flow from the Left to the right side and from	
the Right to the left side.	1
v1.13.1.	
\Pages: Prevent false empty page after \Pages (bug added in 1.13.0)	97
\correct@footinsX@box: Call \correct@footinsX@box and	
\correct@Xfootins@box directly in \print@notesX@forpages and	
\print@Xnotes@forpages.	74
Correct \correct@footinsX@box and \correct@Xfootins@box	74
v1.14.0.	
General: Fix bug with line number position when using \eledsection and similar	
commands for RTL texts with Lua <sup>®</sup> TeX.	1
The \newifs are not followed by boolean values set to false, because it is the TeX	
default setting.	1
v1.15.0.	
\do@actions@nextR: Add actions 1008 and 1009	66
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting	
when a verse is hanging	81
\lineationR: As \lineation, \lineationR automatically set the	
\pstartinfootnote.	33
\n@num: \n@num defined only one time for both Eledmac and Eledpar.	42
\skipnumbering: \skipnumbering defined only one time for both Eledmac and	
Eledpar	47
General: Add \AtEveryPstartCall.	1
Add sameparallelpagenumber option.	12
Fix vertical spurious space before right \eledchapter (bug added in v1.13.0).	1
Prevent vertical space when using \AtEveryPstart or \AtEveryPend with a	
command which prints nothing	1
v1.16.0.	
\newseries@par: Fix bug with \onlysideX.	48
General: Error message when calling \Pages inside ‘pages’ environment and \Columns	
inside ‘pairs’ environment.	1
Error message when starting a Leftside/a Rightside while the previous one has not	
been yet typeset.	1
Error message when using \beginnumbering.. \endnumbering without \pstart.	1
Fix bug with nofamiliar / nocritical option of eledmac.	1
New package option sameparallelpagenumber to have the same page number for both	
left and right side.	1
v1.16.1.	
General: Write information about line-list file version in the correct file.	1

v1.16.2.	
General: Fix bug when adding empty lines before a <code>\pend</code> in combination with some specific penalties setting. . . . .	1
v1.17.0.	
General: Add compatibility of optional argument of <code>\pstart/\pend</code> and <code>\AtEveryPstart/\AtEveryPend</code> with two columns mode. . . . .	1
v1.21.0.	
General: Add <code>\hidenumbering</code> . . . . .	16
v2.0.0.	
<code>\@adv</code> : <code>\@adv</code> defined only in <code>reledmac</code> . . . . .	41
<code>\@lab</code> : <code>\@lab</code> defined only in <code>eledmac</code> . . . . .	79
<code>\@ref@regR</code> : <code>\@ref</code> defined only in <code>reledmac</code> , code specific to right side moved in <code>\ref@regR</code> . . . . .	43
<code>\@set</code> : <code>\@set</code> defined only in <code>reledmac</code> . . . . .	41
<code>\advanceline</code> : <code>\advanceline</code> defined only in <code>reledmac</code> . . . . .	47
<code>\bbl@set@language</code> : Patch <code>\bbl@set@language</code> instead of redefining it . . . . .	84
<code>\do@lockonR</code> : <code>\do@lockon</code> defined only in <code>reledmac</code> . . . . .	41
<code>\endlock</code> : <code>\startlock</code> and <code>\endlock</code> defined only in <code>reledmac</code> . . . . .	47
<code>\endsub</code> : <code>\startsub</code> and <code>\endsub</code> defined only in <code>reledmac</code> . . . . .	47
<code>\fix@page</code> : <code>\fix@page</code> is defined only once in <code>reledmac</code> . . . . .	41
<code>\l@d@set</code> : <code>\l@d@set</code> defined only in <code>reledmac</code> . . . . .	41
<code>\line@marginR</code> : <code>\linenummargin</code> now defined only once time in <code>reledmac</code> . . . . .	34
<code>\page@action</code> : <code>\page@action</code> defined only in <code>reledmac</code> . . . . .	41
<code>\read@linelist</code> : <code>\read@linelist</code> is defined only once time in <code>\reledmac</code> . . . . .	40
<code>\set@line</code> : <code>\set@line</code> defined only in <code>reledmac</code> . . . . .	47
<code>\set@line@action</code> : <code>\set@line@action</code> defined only in <code>reledmac</code> . . . . .	41
<code>\setline</code> : <code>\setline</code> defined only in <code>reledmac</code> . . . . .	47
<code>\setlinenum</code> : <code>\setlinenum</code> defined only in <code>reledmac</code> . . . . .	47
<code>\skip@lockoff</code> : <code>\do@lockoff</code> defined only in <code>reledmac</code> . . . . .	42
<code>\sub@action</code> : <code>\sub@action</code> defined only in <code>reledmac</code> . . . . .	41
<code>\sublinenumincrement*</code> : <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> are now defined only in <code>reledmac</code> . . . . .	36
<code>\theledlanguageR</code> : Patch <code>\otherlanguage</code> instead of redefining it. . . . .	85
General: <code>\@nl</code> is now defined only in <code>reledmac</code> . . . . .	40
<code>\ifbypage@</code> and <code>\ifbypstart@R</code> defined in <code>eledmac</code> . . . . .	32
Fix some bugs with ‘sameparallelpagenumber’ option. . . . .	1
Many code refactored and moved to <code>reledmac</code> . . . . .	1
Package’s name becomes <code>reledpar</code> . . . . .	1
Totally new implementation of ‘sameparallelpagenumber’ option. . . . .	1
<code>chapterinpages</code> : Deleting the old system of managing parallel chapter, keep only the new one with <code>\patchcmd</code> . . . . .	53
v2.1.0.	
General: Fix bug when using <code>\eledsection</code> and related on right pages when page width is short. . . . .	1
Fix bug when using <code>\pagenumbering</code> with memoir (bug added in v2.0.0). . . . .	1
Fix bug with <code>\setparledgroupnotespacing</code> with the <code>shiftedpstarts</code> option. . . . .	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code> . . . . .	1
Options to custom empty right page before <code>\Pages</code> . . . . .	1

v2.2.0.	
General: <code>astanza</code> environment can take an optional argument, which will be the optional argument of <code>\pstart</code> started by this environment. . . . .	1
New tools to number stanza . . . . .	1
v2.2.1.	
General: Fix bug with optional argument of last left <code>\pend</code> . . . . .	1
v2.3.0.	
<code>\Pages</code> : Fix bug when calling <code>\Columns</code> after a <code>\Pages</code> (bug added in v1.13.0). . . . .	98
General: Change some internal codes in order to provide compatibility with $\text{\LaTeX}$ release of october 2015 . . . . .	1
Fix bug with title number in parallel columns . . . . .	1
New line setting command suffixed by R to set only the right side. . . . .	1
v2.4.0.	
<code>\ledstrutR</code> : Deleted <code>\ledstrutL</code> and <code>\ledstrutR</code> . . . . .	105
Fix bug with dotted letter . . . . .	105
General: New way of (not) synchronizing the parallel pages. . . . .	1
Option to switch to <code>\mainmatter</code> when calling <code>\Pages</code> . . . . .	1
v2.5.0.	
General: Disable empty lines as paragraph in <code>astanza</code> . . . . .	1
Fix bug introduced in v1.15.0 which made hanging indentation in verse not work anymore. . . . .	1
New commands <code>\linenummarginR</code> and <code>\linenummargin*</code> . . . . .	1
v2.5.1.	
General: Fix spurious space when using optional argument of <code>astanza</code> environment (introduced in v2.5.0). . . . .	1
v2.5.2.	
General: Fix bug introduced in v2.5.0 with <code>\linenummargin</code> , <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> . . . . .	1
v2.6.0.	
<code>\l@dmake@labelsR</code> : <code>\@Rlineflag</code> is not stored directly after the line number, but as a fifth argument of <code>\the@labelX</code> . Can be retrieved by <code>\xflagref</code> . . . . .	79
General: <code>\Xlineflag</code> and <code>\Xendlineflag</code> added . . . . .	1
<code>\printlinesR</code> deleted . . . . .	1
Error message when calling <code>\Pages</code> or <code>\Columns</code> without previous pages or pairs environment. . . . .	1
Fix bug with footnote numbering when using the same series of familiar footnotes on both sides. . . . .	1
Fix bug with right side title number when using title commands before pages or columns environments. . . . .	1
Fix compatibility with <code>babel</code> (broken in v2.0.0). . . . .	1
No error messages about ends of left / right page when using the <code>\syntonly</code> command of the <code>syntonly</code> package. . . . .	1
v2.6.1.	
General: Fix bug, introduced in v2.6.0, with footnote numbering when using <code>perpage</code> package. . . . .	1
v2.6.2.	
<code>\newseries@par</code> : The $\text{\TeX}$ counter <code>\footnoteX@reading</code> is defined in <code>reledmac</code> . . .	48
General: Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.6.0). . . . .	1



Fix bug (added in v2.6.0) with footnote numbering in parallel typesetting while using <code>polyglossia</code> with specific numbering systems (like Greek). . . . .	1
v2.6.3.	
General: Fix spurious dot when using <code>\linenummargin</code> on right side (introduced in v2.5.0). . . . .	1
v2.7.0.	
General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package. . . . .	1
v2.7.1.	
General: Fix bug added in <code>reledmac</code> 2.8.2, when typesetting parallel text just after a sectioning command . . . . .	1
v2.8.0.	
General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> options. . . . .	1
Add <code>\linenumberLevenifblanktrue</code> and <code>\linenumberRevenifblank</code> commands . . . . .	1
Fix bug when the right line number style is not the same to the left line number style . . . . .	1
v2.9.0.	
General: Add <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> . . . . .	1
More specific error messages. . . . .	1
v2.9.1.	
General: Prevent <code>\Xtxtbeforenotes</code> hook from causing notes to go beyond the bottom margin . . . . .	1
v2.10.0.	
<code>\do@actions@nextR</code> : Add action 1010 . . . . .	66
General: Add new tools to make apparatuses of manuscripts . . . . .	1
v2.11.0.	
<code>\correct@footinsX@box</code> : Clarification in the handbook about the use of <code>\Xonlyside</code> and <code>onlysideX</code> and error message if misuse. . . . .	74
New implementation of <code>\Xonlyside</code> and <code>\onlysideX</code> hooks, prevent trouble with vertical spacing. . . . .	74
General: Compatibility with <code>reledmac</code> 's <code>auxdir</code> option. . . . .	1
v2.12.0.	
General: Add <code>\edtextlater</code> and <code>\edtextnow</code> . . . . .	1, 77
Fix bug with <code>\AtBeginPairs</code> . . . . .	1
Fix bug with <code>\lineationR</code> . . . . .	1
v2.13.0.	
General: Compatibility with the new features of <code>reledmac</code> 2.15.0 . . . . .	1
v2.13.1.	
General: Simplification of the code concerning the sectioning command . . . . .	1
v2.14.0.	
General: Add <code>\linenumberlistR</code> , equivalent to <code>\linenumberlist</code> for the right side. . . . .	1
Error message when missing to define left / right side. . . . .	1
More explicit error message when missing to define stanza indent. . . . .	1
New commands to have specific line margins for text in parallel columns . . . . .	1
When indexing texts in sidenotes with <code>\edtext</code> , referring to the line number where the sidenote is called. . . . .	1