

Parallel typesetting for critical editions: the reledpar package*

Maïeul Rouquette[†] based on the original ledpar by Peter Wilson
Herries Press[‡]

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

1 Introduction	5
1.1 Aim of this package	5
1.2 Historical overview	5
2 Options	6
2.1 Synchronization’s options	6
2.2 Other options	6
3 General	7
4 Parallel columns	8
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

*This file (reledpar.dtx) has version number v2.25.9, last revised 2025/04/13.

[†]maieul at maieul dot net

[‡]herries dot press at earthlink dot net

4.2.4	Mixing two columns and one column texts	9
4.2.5	Schemas of column typesetting	9
4.2.6	Continuous numbering between single and double column texts	9
4.3	Note about <code>\AtEveryPstart*</code> and <code>\AtEveryPend*</code>	10
5	Keeping translation in the outside/inside column	10
6	Facing pages	11
6.1	Basic usage	11
6.2	Setting	11
6.2.1	Text width	11
6.2.2	Way of synchronizing	11
6.2.3	Page number	13
6.2.4	Page breaking	13
6.2.5	Right page before <code>\Pages</code>	13
6.2.6	Notes about <code>\mainmatter</code>	13
6.3	Critical and familiar footnotes	13
6.3.1	Notes height setting	13
6.3.2	About the numbering of familiar footnotes	14
6.3.3	Using <code>perpage</code> package	14
6.3.4	Notes for one side only	14
6.3.5	Familiar notes called on the right side, but to be printed on the left side	15
6.4	Critical notes called on leftside but printed on right side	15
6.5	Using line flag	15
7	Left and right texts	16
7.1	Environments	16
7.2	Numbering text lines and paragraphs	16
7.3	First line number and line number increment	17
7.4	Lineation system	17
7.5	Line number style	17
7.6	Line number margin	18
7.7	Line flag	18
7.8	Chunks	19
7.9	<code>\AtEveryPstart</code> and <code>\AtEveryPstartCall</code>	19
7.10	Language setting	19
7.11	Executing code at each line	20
7.12	Executing code at each page	20
8	Verse	20
9	Side notes	21
10	Parallel ledgroups	21
10.1	General	21
10.2	Parallel ledgroups and <code>setspace</code> package	22
11	Sectioning commands	22
12	Notes about page number	22

<i>Contents</i>	3
I Implementation overview	23
II Preliminaries	23
II.1 Package's meta-data	23
II.2 Package's requirement	23
II.3 Package's options	23
II.4 Package's options	24
II.4.1 Synchronization's options	24
II.4.2 Position options	25
II.4.3 Other options	25
II.5 Determining side and category of parallel processing	26
II.6 Text's width	26
II.7 Messages	26
II.8 Naming macros	29
III Sectioning commands	30
IV Line counting	34
IV.1 Setting lineation reset	34
IV.2 Setting line number margin	35
IV.3 Setting lineation start and step	37
IV.4 Setting line flag	38
IV.5 Setting line number style	38
IV.6 Print marginal line number	39
IV.7 Line-number counters and lists	40
IV.7.1 Correspond to those in <code>reledmac</code> for regular or left text	40
IV.7.2 Specific to <code>reledpar</code>	41
IV.8 Reading the line-list file	41
IV.9 Commands within the line-list file	41
IV.10 Writing to the line-list file	47
V Marking text for notes	49
V.1 Specific hooks and commands for notes	50
V.1.1 Notes to be printed on one side only	50
V.1.2 Tools specific to familiar footnotes	50
V.1.3 Get correct footnote number	52
V.2 Create hooks	52
V.3 Init standards series (A,B,C,D,E,Z)	53
V.4 Tools specific to \LaTeX 's classical footnotes	53
VI Pstart numbers dumping and restoration	53
VII Parallel environments	54
VIII Paragraph decomposition and reassembly	57
VIII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	57
VIII.2 Processing one line	63
VIII.3 Line and page number computation	69
VIII.4 Line number printing	72
VIII.5 Pstart number printing in side	74
VIII.6 Add insertions to the vertical list	76

VIII.7 Penalties	77
VIII.8 Printing leftover notes	77
IX Footnotes	78
IX.1 Footnotes output specific to <code>\Pages</code>	78
IX.2 Critical footnote printed in right side	81
X Cross referencing	83
XI Sidenotes	84
XII Verse	86
XIII Fixing babel and polyglossia	88
XIV Counts and boxes for parallel texts	90
XV Checking text to be processed	92
XVI Parallel columns	93
XVII Parallel pages	104
XVII.1 Specific counters	104
XVII.2 Main macro	104
XVII.3 Ensure all notes are printed at the end of parallel pages	110
XVII.4 Struts	111
XVII.5 Page clearing	111
XVII.6 Lines managing	112
XVII.7 Page break managing	114
XVII.8 Getting boxes content	117
XVIII The mainmatter option of <code>\Pages</code>	120
XIX Sections' titles' commands	121
XX Page break/no page break, depending on the specific line	122
XXI Parallel ledgroup	122
XXII Compatibility with <code>eledmac</code>	126
XXIII The End	126
Appendix A Some things to do when changing version	127
A.1 Migration to <code>eledpar 1.4.3</code>	127
A.2 Migration from <code>eledpar</code> to <code>reledpar</code>	127
A.2.1 Deprecated options	127
A.2.2 <code>\renewcommand</code> replaced with <code>command</code>	127
A.2.3 Commands the names of which have changed	128
A.3 Migration to <code>reledpar 2.2.0</code>	128
A.4 Migration to <code>reledpar 2.3.0</code>	128
A.5 Migration to <code>reledpar 2.4.0</code>	128

A.6 Migration to reledpar 2.5.0	128
A.7 Migration to reledpar 2.6.0	128
A.8 Migration to reledpar 2.6.1	128
A.9 Migration to reledpar 2.14.0	128
A.10 Migration to reledpar 2.14.1	129
A.11 Migration to reledpar v. 2.17.2	129
A.12 Migration to reledpar v. 2.24.3	129
References	129
Index	129
Change History	147

1 Introduction

1.1 Aim of this package

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

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

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

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

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

1.2 Historical overview

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

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

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

2 Options

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

2.1 Synchronization's options

Please read the paragraph on synchronization's option on [6.2.2 p. 11](#) 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`.¹

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.²

continuousnumberingwithcolumns allow you to alternate between normal typesetting and parallel column typesetting without resetting the line number (see [4.2.6 p. 9](#)).

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

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

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

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

¹This option can either be used on `reledmac` or `reledpar`.

²This 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 parallel³. This can be done either as setting the “Leftside” and “Rightside” texts in two columns or on facing pages. In the paired pages case footnotes are placed at the bottom of the page on which they are called out — that is, footnotes belonging to the left are set at the foot of a left (even numbered) page, and those for right texts are at the bottom of the relevant right (odd numbered) page. However, in the columnar case, all footnotes are set at the bottom left of the page on which they are called out — they are not set below the relevant column.

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

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

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

```
\maxchunks{5120}
```

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

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

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

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

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

4 Parallel columns

4.1 Basic use

`pairs` Numbered text that is to be set in columns must be within a `pairs` environment. Within the environment the text for the lefthand and righthand columns is placed within the `Leftside` and `Rightside` environments, respectively; these are described in more detail below in section 7.

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

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

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

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

There is no required pagebreak before or after the columns.

4.2 Setting

4.2.1 Column's width

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

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

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

4.2.2 Column's separator

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

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

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

4.2.3 Column's positions

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

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

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

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

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

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

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

4.2.4 Mixing two columns and one column texts

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

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

4.2.5 Schemas of column typesetting

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

4.2.6 Continuous numbering between single and double column texts

If you want to have continuous line numbers between multiple columns and single columns, use the `continuousnumberingwithcolumns` option when loading `reledmac` or `reledpar`. You will need to use `\pausenumbering... \resumenumbering` in addition to `\beginnumbering... \endnumbering` (see 5.2.7 p. 20).

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

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

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

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

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

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

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

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

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

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

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

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

5 Keeping translation in the outside/inside column

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

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

6 Facing pages

6.1 Basic usage

`pages` Numbered text that is to be set on facing pages must be within a `pages` environment. Within the environment the text for the lefthand and righthand pages is placed within the `Leftside` and `Rightside` environments, respectively.

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

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

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

6.2 Setting

6.2.1 Text width

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

6.2.2 Way of synchronizing⁴

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:

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

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

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

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

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

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

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

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

6.2.3 Page number

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

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

Please also read advising in [12 p. 22](#).

6.2.4 Page breaking

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

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

6.2.5 Right page before `\Pages`

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

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

`prevpgstyle=<style>` in order to set the style of this page. A common value of `<style>` is empty. Use `prevpgstyle=empty` will suppress header and footer in this page.

Please also read advising in [12 p. 22](#).

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

6.2.6 Notes about `\mainmatter`

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

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

6.3 Critical and familiar footnotes

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

6.3.1 Notes height setting

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

However, the `reledmac` default setting for the maximum allotted size to notes is greater than `\textheight`. That makes impossible for long notes to flow across pages.⁵ 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 (7.12.7 p. 53). As an advisable setting:

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

6.3.2 About the numbering of familiar footnotes

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

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

6.3.3 Using `perpage` package

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

So do not set:

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

But set:

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

6.3.4 Notes for one side only

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

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

L for notes to be confined only on the left side and to R for notes to be confined only on the right side.

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

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

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

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

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

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

6.4 Critical notes called on leftside but printed on right side

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

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

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

6.5 Using line flag

`\Xlineflag` Use `\Xlineflag[<s>]` to add right line flag (7.7 p. 18) to right critical footnotes and `\Xendlineflag`

`\Xendlineflag[⟨s⟩]` to add it to right critical endnotes.

7 Left and right texts

7.1 Environments

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

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

7.2 Numbering text lines and paragraphs

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

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

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

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

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



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

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

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

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

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

`\thepstartR`

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

`\hidenumbering` When inserted into a numbered line the macro `\hidenumbering` causes the number for that particular line to be hidden; namely, no line number will print. Note that if you use it in `\stanza`, you must call it at the beginning of the verse.

7.3 First line number and line number increment

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

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

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

```
\firstlinenumR
\linenumincrementR
\firstsublinenumR
```

7.4 Lineation system

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

`\lineationR`

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

`\reledmac` allows you to define a `\linenumberlist` to explicitly define (5.3.2 p. 21) in which line the line number will be printed. In parallel typesetting, this command affects only left side. For right side, you have to define `\linenumberlistR`.

7.5 Line number style

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

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

`\linenumberstyle*`

`\sublinenumberstyle*`

7.6 Line number margin

`\linenummarginR` `\linenummarginR{<margin>}` sets the line margin for right side. `\linenummargin*{<margin>}` sets for both side. `<margin>` can be, as for `reledmac`'s `\linenummargin` one of these values: `left`, `right`, `inner`, `outer`.

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

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

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

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

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

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

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

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

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

7.7 Line flag

`\setRlineflag` A “R” is appended to the line numbers of the right texts. This may be useful for parallel columns but for parallel pages it might be more appropriate to redefine it using `\setRlineflag{<flag>}`. Use `\setRlineflag{}` to empty it.

7.8 Chunks

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

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

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

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

7.9 `\AtEveryPstart` and `\AtEveryPstartCall`

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

Consequently:

- The argument of `\AtEveryPstart` (see 5.2.4 p. 19) is called before every chunk is printed, except if you used an optional argument for the `\pstart`.
- The argument of `\AtEveryPstartCall` is called before every `\pstart`.

7.10 Language setting

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

7.11 Executing code at each line

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

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

About these two hooks, read `reledmac`'s handbook (5.6 p. 27).

7.12 Executing code at each page

`\Rightpagehook` When typesetting parallel pages, the `\Leftpagehook` and `\Rightpagehook` macros are executed each time `reledpar` switches to Left / Right page.

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

8 Verse

If you are typesetting verses with `reledmac` you can use the `\stanza` construct, and you can also use this in right or left parallel texts. In this case each verse line is a chunk which has two implications. (1) you can unexpectedly exceed the `\maxchunks` limit or the overall limit on the number of boxes, and (2) left and right verse lines are matched, which may not be desirable if one side requires more print lines for verse lines than the other does.

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

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

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

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

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

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

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

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

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

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

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

`\thestanzaL` When using `\stanzanumtrue` (9.10 p. 58) in parallel typesetting, stanza counter is replaced by `stanzaL` counter in left side and by `stanzaR` counter in right side. Consequently, you can redefine `\thestanzaL` and `\thestanzaR` to change their aspect.

9 Side notes

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

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

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

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

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

10 Parallel ledgroups

10.1 General

You can also make parallel ledgroups (see the documentation of `reledmac` about ledgroups, 10 p. 59). To do it you have:

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

See the following example:

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

```

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

```

10.2 Parallel ledgroups and setspace package

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

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

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

11 Sectioning commands

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

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

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

12 Notes about page number

If you use `sameparallepagenumber` option (6.2.3 p. 13) or `prevpgnotnumbered` option (6.2.5 p. 13), please read the following paragraph if you want to manipulate page numbers manually.

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

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

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

I Implementation overview

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

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

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

II Preliminaries

II.1 Package’s meta-data

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

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledpar}[2025/04/13 v2.25.9 reledmac extension for
  parallel texts]%
4
5 %

```

II.2 Package’s requirement

Few commands use `\xspace` command.

```

6 \RequirePackage{xspace}%
7 %

```

II.3 Package’s options

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

```

8 \RequirePackage{xkeyval}
9 %

```

II.4 Package's options

II.4.1 Synchronization's options

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

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

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

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

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

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

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

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

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


```

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

```

II.4.2 Position options

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

```

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

```

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

```

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

```

II.4.3 Other options

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

```

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

```

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

```

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

```

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

```

54 \DeclareOptionX{sameparallelpagenumber}{\sameparallelpagenumbertrue}
55 \DeclareOptionX{prevpgnotnumbered}{\prevpgnotnumberedtrue}
56 %

```

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

```

57 \DeclareOptionX{prevpgstyle}{\gdef\prevpgstyle{#1}}%
58 %
59 \ProcessOptionsX%
60 %

```

II.5 Determining side and category of parallel processing

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

```

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

```

II.6 Text’s width

```

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

```

```

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

```

II.7 Messages

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

```

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

```

```

\reledpar@warning 70 \newcommand{\reledpar@warning}[1]{\PackageWarning{reledpar}{#1}}%
71 %

```

```

\led@err@TooManyPstarts 72 \newcommand*{\led@err@TooManyPstarts}{%
73 \reledpar@error{Too many \string\pstart\space without printing.
74 Some text will be lost}{\@ehc}}
75 %

```

```

\led@err@polyglossiaTooOld76 \newcommand{\led@err@polyglossiaTooOld}{%
77   \reledpar@error{You use the polyglossia package.%
78   \MessageBreak However, reledpar requires version 2022/04/10 v1.55 or
later of the polyglossia package.%
79   \MessageBreak Please update your polyglossia version}{\@ehc}%
80 }%
81 %

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

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

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

\led@err@Pages@InsideEnv103 \newcommand*\led@err@Pages@InsideEnv}{%
\led@err@Columns@InsideEnv104   \reledpar@error{\string\Pages\space must be called *outside* of the `
pages` environment}{\@ehc}}
105 \newcommand*\led@err@Columns@InsideEnv}{%
106   \reledpar@error{\string\Columns\space must be called *outside* of the `
pairs` environment}{\@ehc}}
107 %

\led@err@Pages@WithoutEnv108 \newcommand*\led@err@Pages@WithoutEnv}{%
\led@err@Columns@WithoutEnv109   \reledpar@error{\string\Pages\space called without previous `pages`
environment}{\@ehc}}
110 \newcommand*\led@err@Columns@WithoutEnv}{%
111   \reledpar@error{\string\Columns\space called without previous `pairs`
environment}{\@ehc}}
112 %

```

```

\led@err@Pages@WithoutRightside 13 \newcommand*{\led@err@Pages@WithoutRightside}{%
\led@err@Pages@WithoutLeftside 14 \reledpar@error{\string\Pages\space called without any preceding `
\led@err@Columns@WithoutRightside 15 Rightside` environment (or environment without content)}{\@ehc}}%
\led@err@Columns@WithoutLeftside 15 \newcommand*{\led@err@Pages@WithoutLeftside}{%
116 \reledpar@error{\string\Pages\space called without any preceding `
Leftside` environment (or environment without content)}{\@ehc}}%
117 \newcommand*{\led@err@Columns@WithoutRightside}{%
118 \reledpar@error{\string\Columns\space called without any preceding `
Rightside` environment (or environment without content)}{\@ehc}}%
119 \newcommand*{\led@err@Columns@WithoutLeftside}{%
120 \reledpar@error{\string\Columns\space called without any preceding `
Leftside` environment (or environment without content)}{\@ehc}}%
121 %

\led@error@fail@patch@thepage 22 \newcommand{\led@error@fail@patch@thepage}{%
123 \reledpar@error{Fail to patch \string\thepage\space command.}{\@ehc}%
124 }%
125 %

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

\led@error@note@called@onrightside 30 \newcommand{\led@error@note@called@onrightside}[1]{%
\led@error@note@called@onleftside 31 \reledpar@error{#1 called on right side, despite your configuring it to
be for the left side only}{\@ehc}%
132 }%
133 \newcommand{\led@error@note@called@onleftside}[1]{%
134 \reledpar@error{#1 called on left side, despite your configuring it to be
for the right side only}{\@ehc}%
135 }%
136 %

\led@error@fail@patch@mempnum 37 \newcommand{\led@error@fail@patch@mempnum}{%
138 \reledpar@error{Fail to patch \string@mempnum\space command.}{\@ehc}%
139 }%
140 %

\led@error@fail@patch@outputpage 41 \newcommand{\led@error@fail@patch@outputpage}{%
142 \reledpar@error{Fail to patch \string\outputpage\space command.}{\@ehc}%
143 }%
144 %

```

```

\error@edtext@later@now45 \newcommand{\led@error@edtext@later@now}{%
146 \reledpar@error{You call \the\edtext@later\space \string\edtextlater\
space commands, but \the\edtext@now\space \string\edtextnow\space commands
}\@ehc}% (did I follow your intent correctly here?)
147 }%
148 %

```

```

\warn@ChangeSyncOption49 \newcommand*\led@warn@ChangeSyncOption}[1]{%
150 \reledpar@warning{You have changed synchronization's options since the
last run. We have not read line-list file #1. Please run LaTeX again.}%
151 }%
152 %

```

```

\warn@setting@in@rightside53 \newcommand{\led@warn@setting@in@rightside}[1]{%
154 \reledpar@warning{You use #1 inside rightside environment.\MessageBreak%
155 Such behavior is deprecated.\MessageBreak%
156 Use instead #1R or #1* in your preamble.}%
157 }
158 %

```

```

\error@missing@numbering59 \newcommand{\led@error@missing@numbering}[1]{%
160 \reledpar@error{Missing \string\pstart\space ... \space\string\pend\space
inside `#1` environment}{\@ehc}%
161 }%
162 %

```

II.8 Naming macros

The \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
163 \providecommand*\newnamebox}[1]{%
164 \expandafter\newbox\csname #1\endcsname}
165 \providecommand*\setnamebox}[1]{%
166 \expandafter\setbox\csname #1\endcsname}
167 \providecommand*\unhnamebox}[1]{%
168 \expandafter\unhbox\csname #1\endcsname}
169 \providecommand*\unvnamebox}[1]{%
170 \expandafter\unvbox\csname #1\endcsname}
171 \providecommand*\namebox}[1]{%
172 \csname #1\endcsname}
173
174 %

```

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

```

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

```

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

`\usernamecount`

```

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

```

III Sectioning commands

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

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

```

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

```

`\ifpst@rtedL` `\ifpst@rtedL` is set FALSE at the start of left side numbering, and similarly for

`\ifpst@rtedR` `\ifpst@rtedR`. `\ifpst@rtedL` is defined in `reledmac`.

```

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

```

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

```

202 \newcommand*{\beginnumberingR}{%
203   \ifnumberingR
204     \led@err@NumberingStarted
205     \endnumberingR
206   \fi
207   \global\l@dnumpstartsR \z@
208   \global\pst@rtedRfalse
209   \global\numberingRtrue
210   \global\afterendnumberingRfalse%
211   \global\advance\section@numR \@ne
212   \global\absline@numR \z@
213   \gdef\normal@page@breakR{}
214   \gdef\l@prev@pbR{}
215   \gdef\l@prev@nopbR{}
216   \global\line@numR \z@
217   \global\@lockR \z@
218   \global\sub@lockR \z@
219   \global\sublines@Rfalse
220   \global\let\next@page@numR\relax
221   \global\let\this@section@next@page@numR\relax%
222   \global\let\sub@change\relax
223   \global\last@page@numR=-10000%
224   \global\stopmsdata@inserted@true%
225   \global\let\@msdata@list\relax%
226   \global\csundef{\@msdata@\add@msd@cR @dataR}%
227   \ifcontinuousnumberingwithcolumns%
228     \ifnumbering%
229       \global\resumenumberingR@starttrue%
230     \fi%
231     \ifnumberpstart%
232       \addtocounter{pstartR}{1}%
233     \fi%
234   \else%
235     \setcounter{pstartR}{1}%
236   \fi%
237   \message{Section \the\section@numR R }%
238   \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
239   \l@dend@stuff
240   \begingroup
241   \initnumbering@sectcountR
242   \gdef\eled@sectionsR@{ }%
243   \if@noeled@sec\else%
244     \makeatletter%
245     \InputIfFileExists%
246       {\l@auxdir\jobname.eledsec\the\section@numR R}%
247       {}%
248       {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@numR R}}%
249     \makeatother%
250     \immediate\openout\eled@sectioningR@out=\l@auxdir\jobname.eledsec\the\
section@numR R\relax%
251   \fi%
252 }
253 %

```

`\endnumbering` This is the left text version of the regular `\endnumbering` and must follow the last text for a left text numbered section. It sets `\ifpstrtedL` to FALSE. It is fully defined in `reledmac`.

`\endnumberingR` This is the right text equivalent of `\endnumbering` and must follow the last text for a right text numbered section.

```

254 \def\endnumberingR{%
255   \ifnumberingR
256     \global\numberingRfalse
257     \global\afterendnumberingRtrue%
258     \normal@pars
259     \ifnum\l@dnumpstartsR=0%
260       \led@err@NumberingWithoutPstart%
261     \fi%
262     \global\page@numR=\this@section@page@numR%
263     \global\last@page@numR=\this@section@last@page@numR%
264     \global\let\next@page@numR\this@section@next@page@numR%
265     \ifl@dpairing
266       \global\pstrtedRfalse
267     \else
268       \ifx\insertlines@listR\empty\else
269         \global\noteschanged@true
270       \fi
271       \ifx\line@listR\empty\else
272         \global\noteschanged@true
273       \fi
274     \fi
275     \ifnoteschanged@
276       \led@mess@NotesChanged
277     \fi
278   \else
279     \led@err@NumberingNotStarted
280   \fi
281 \endgroup
282 \if@noeled@sec\else%
283   \immediate\closeout\eled@sectioningR@out%
284 \fi%
285 }
286
287 %

```

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

```

288 \newcounter{chapterR}
289 \newcounter{sectionR}
290 \newcounter{subsectionR}
291 \newcounter{subsubsectionR}
292
293 \newcount\old@chapter%
294 \newcount\old@section%
295 \newcount\old@subsection%
296 \newcount\old@subsubsection%

```



```

297 \newcommand{\save@section@number}{%
298   \ifdefined\c@chapter%
299     \global\old@chapter\value{chapter}%
300   \fi%
301   \global\old@section\value{section}%
302   \global\old@subsection\value{subsection}%
303   \global\old@subsubsection\value{subsubsection}%
304 }%
305 \newcommand{\initnumbering@sectcountR}{
306   \ifdefined\c@chapter%
307     \setcounter{chapterR}{\old@chapter}%
308   \fi%
309   \setcounter{sectionR}{\old@section}%
310   \setcounter{subsectionR}{\old@subsection}%
311   \setcounter{subsubsectionR}{\old@subsubsection}%
312   \set@sectcountR%
313 }
314 \newcommand{\set@sectcountR}{%
315   \let\c@chapter\c@chapterR%
316   \let\c@section\c@sectionR%
317   \let\c@subsection\c@subsectionR%
318   \let\c@subsubsection\c@subsubsectionR%
319 }%
320 %

```

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

```

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

```

```

348 %
349 %
\memorydumpL \memorydump is a shorthand for \pausenumbering\resumenumbering. This will clear
\memorydumpR the memorised stuff for the previous chunks while keeping the numbering going.
350 \newcommand*{\memorydumpL}{%
351   \endnumbering
352   \numberingtrue
353   \global\pst@rtedLtrue
354   \global\advance\section@num \@ne
355   \led@mess@SectionContinued{\the\section@num}%
356   \line@list@stuff{\jobname.\extensionchars\the\section@num}%
357   \l@dend@stuff}
358
359 \newcommand*{\memorydumpR}{%
360   \endnumberingR
361   \numberingRtrue
362   \global\pst@rtedRtrue
363   \global\advance\section@numR \@ne
364   \led@mess@SectionContinued{\the\section@numR R}%
365   \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
366   \l@dend@stuff}
367
368 %

```

IV Line counting

IV.1 Setting lineation reset

Sometimes you want line numbers that start at 1 at the top of each page; sometimes you want line numbers that start at 1 at each `\pstart`; other times you want line numbers that start at 1 at the start of each section and increase regardless of page breaks. `reledpar` lets you choose different schemes for the left and right texts.

`\lineationR` `\lineationR{<word>}` is the macro used to select the lineation system for right texts. Its argument is a string: either `page`, `pstart` or `section`.

```

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

```

```

384     \global\bystart@Rtrue
385     \unless\ifnocritical@%
386         \Xpstart%
387     \fi%
388 \else
389     \def\@tempb{section}
390     \ifx\@tempa\@tempb
391         \global\bypage@Rfalse%
392         \global\bystart@Rfalse%
393         \unless\ifnocritical@%
394             \Xpstart [] [false]%
395         \fi%
396     \else
397         \led@warn@BadLineation
398     \fi%
399 \fi
400 \fi
401 \fi}}
402 %

```

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

```

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

```

IV.2 Setting line number margin

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

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

It is defined only once time, in `reledmac`.

```

408 \newcount\line@marginR
409 %

```

By default put right text numbers at the right.

```

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

```

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

```

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

```

```

415 \ifnum\@l@tempcntb>\m@ne%
416 \global\line@marginR=\@l@tempcntb%
417 \fi%
418 }
419 \WithSuffix\newcommand\linenummargin*[1]{%
420 \l@dgetline@margin{#1}%
421 \ifnum\@l@tempcntb>\m@ne%
422 \global\line@marginR=\@l@tempcntb%
423 \global\line@margin=\@l@tempcntb%
424 \fi%
425 }
426 \ifmovecolumnspostiononrightpage%
427 \linenummargin{inner}%
428 \linenummarginR{outer}%
429 \fi%
430 %

```

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

`\linenummarginColumns`

`\linenummarginColumnsR`

```

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

```

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

`\linenumOnlyPagesForColumnsR`

```

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

```

```

461 }%
462
463 \def\linenum@OnlyPages@ForColumnsR{%
464 \newcommand{\linenumOnlyPagesForColumnsR}[1]{%
465 \gdef\linenum@OnlyPages@ForColumnsR{#1}%
466 }%
467 %

```

IV.3 Setting lineation start and step

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

```

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

```

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

```

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

```

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

```

479 \firstsublinenum
480 \sublinenumincrement
481 \firstlinenum*
482 \linenumincrement*
483 \firstsublinenum*
484 \sublinenumincrement*
485
486
487
488
489
490
491
492
493
494
495

```

```

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

```

IV.4 Setting line flag

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

```

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

```

IV.5 Setting line number style

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

```

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

```

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

```

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

```

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

```

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

```

```
531 }%
532 %
533 %
```

IV.6 Print marginal line number

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

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

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

```
537 \newcommand*\leftlinenumR}{%
538   \l@dlinenumR%
539   \kern\linenumsep%
540 }%
541 \newcommand*\rightlinenumR}{%
542   \kern\linenumsep%
543   \l@dlinenumR%
544 }%
545
546
547 \newcommand*\l@dlinenumR}{%
548   \numlabfont%
549   \unless\iflinenumannotationotherside%
550     \ifdefstring{Xlinenumannotationposition@side}{before}{%
551       \l@wrap@ifnotemptybox{Xwraplinenumannotation@side}{%
552         \csuse{annotR@the\absline@numR @the\section@numR}%
553       }%
554     }%
555   \fi%
556   \linenumrepR{\line@numR}\@Rlineflag%
557   \ifsublines@R
558     \ifnum\subline@numR>\z@
559       \unskip\fullstop\sublinenumrepR{\subline@numR}%
560     \fi
561   \fi%
562   \unless\iflinenumannotationotherside%
563     \ifdefstring{Xlinenumannotationposition@side}{after}{%
564       \l@wrap@ifnotemptybox{Xwraplinenumannotation@side}{%
565         \csuse{annotR@the\absline@numR @the\section@numR}%
566       }%
567     }%
568   \fi%
569 }%
570
571 %
```

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

```

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

```

IV.7 Line-number counters and lists

IV.7.1 Correspond to those in `reledmac` for regular or left text

We need another set of counters and lists for the right text, corresponding to those in `reledpar` for regular or left text.

`\line@numR` The count `\line@numR` stores the line number that is used in the right text's marginal line numbering and in notes. The count `\subline@numR` stores a sub-line number that qualifies `\line@numR`. The count `\absline@numR` stores the absolute number of lines since the start of the right text section: that is, the number we have actually printed, no matter what numbers we attached to them.

```

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

```

`\line@listR` Now we can define the list macros that will be created from the line-list file. They are directly analogous to the left text ones. The full list of action codes and their meanings is given in the `reledmac` manual.

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

```

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

```

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

```

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

```


IV.7.2 Specific to reledpar

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

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

IV.8 Reading the line-list file

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

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

`\@par@sync@option` When typesetting parallel pages, `\@par@sync@option` check if we have changed the synchronization's option since the last run. If true, we just not read the numbered file.

```
613 \newcommand{\@par@sync@option}[1]{%
614   \IfStrEq{#1}{\@par@this@sync@option}%
615   {}%
616   {\ifledRcol%
617     \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
618   }%
619   \else%
620     \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
621   }%
622   \fi%
623   \endinput%
624   }%
625 }
```

`\read@linelist` `\read@linelist{<file>}` is the control sequence that is called by `\beginnumbering` (via `\line@list@stuff`) to open and process a line-list file; its argument is the name of the file. . It is defined only once time in `reledmac`.

IV.9 Commands within the line-list file

This section defines the commands that can appear within a line-list file, except for `\@lab` which is in a later section among the cross-referencing commands it is associated with.

The macros with `action` in their names contain all the code that modifies the action-code list.

`\@nl@regR` `\@nl@regR` is called by `\@nl` if we are on a right side. It does everything related to the start of a new line of numbered text on a right side.

```

625 \newcommand{\@nl@regR}{%
626   \ifx\l@dchset@num\relax \else
627     \advance\absline@numR \@ne
628     \csgdef{l@dchset@num@R\the\absline@numR}{}%To remember this line have
been marked by a \setlinenum
629     \set@line@action
630     \let\l@dchset@num\relax
631     \advance\absline@numR \m@ne
632     \advance\line@numR \m@ne%   % do we need this?
633   \fi
634   \reset@current@annot%
635   \advance\absline@numR \@ne
636   \ifx\next@page@numR\relax \else
637     \page@action
638     \let\next@page@numR\relax
639   \fi
640   \ifx\sub@change\relax \else
641     \ifnum\sub@change>\z@
642       \sublines@Rtrue
643     \else
644       \sublines@Rfalse
645     \fi
646     \sub@action
647     \let\sub@change\relax
648   \fi
649   \ifcase\@lockR
650   \or
651     \@lockR \tw@
652   \or\or
653     \@lockR \z@
654   \fi
655   \ifcase\sub@lockR
656   \or
657     \sub@lockR \tw@
658   \or\or
659     \sub@lockR \z@
660   \fi
661   \ifsublines@R
662     \ifnum\sub@lockR<\tw@
663       \advance\subline@numR \@ne
664     \fi
665   \else
666     \ifnum\@lockR<\tw@
667       \advance\line@numR \@ne \subline@numR \z@
668     \fi
669   \fi}
670
671
672 %

```

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

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

`\@adv` The `\@adv{<num>}` macro advances the current visible line number by the amount specified as its argument. This is used to implement `\advance@line`. 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 `\set@line`. 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 `\set@linenum`. It is defined in `reledmac`.

`\page@action` `\page@action` adds an entry to the action-code list to change the page number. It is defined in `reledmac`.

`\set@line@action` `\set@line@action` adds an entry to the action-code list to change the visible line number. It is defined in `reledmac`.

`\sub@action` `\sub@action` adds an entry to the action-code list to turn sub-lineation on or off, according to the current value of the `\ifsublines@` flag. It is defined in `reledmac`.

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

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

```

696 \lockR \one
697 \else
698 \ifnum\lockR=\thr@@
699 \lockR \one
700 \fi
701 \fi
702 \fi}
703
704 %

```

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

```

705
706
707 \newcommand{\do@lockoffR}{%
708 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
709 \ifsublines@R
710 \xright@appenditem{-1006}\to\actions@listR
711 \ifnum\sub@lockR=\tw@@
712 \sub@lockR \thr@@
713 \else
714 \sub@lockR \z@
715 \fi
716 \else
717 \xright@appenditem{-1004}\to\actions@listR
718 \ifnum\lockR=\tw@@
719 \lockR \thr@@
720 \else
721 \lockR \z@
722 \fi
723 \fi}
724
725
726 %

```

`\n@num`

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

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

```

727 \newcount\insert@countR
728 %

```

- #2, a sequence of other line-list-file commands, executed to determine the ending line-number. This may also include other `\@ref` commands, corresponding to uses of `\edtext` within the first argument of another instance of `\edtext`.

`\@ref` itself is defined in `reledmac`. It calls `\ref@reg` or `\ref@regR`, depending whether we are in left or right side. Here, we define only `\ref@regR`, `\ref@reg` is already defined in `reledmac`.

The first thing `\@ref@regR` itself does is to add the specified number of items to the `\insertlines@listR` list.

```

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

```

Next, process the second argument to determine the page and line numbers for the end of this lemma. We temporarily equate `\@ref` to a different macro that just executes its argument, so that nested `\@ref` commands are just skipped this time. Some other macros need to be temporarily redefined to suppress their action.

```

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

```

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

```

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

```

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

```

762 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
763   @edtext@level\endcsname}%
764 %

```

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

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

Decrease edtext level counter.

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

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

Declare and init boolean for lemma in this level.

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

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

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

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

```
788 \renewcommand*{\@pend}[1]{%
789 \ifbypstart@%
790 \unless\ifinstanza%
791 \global\line@num=0%
792 \fi%
793 \fi%
794 \xright@appenditem{#1}\to\linesinpar@listL%
795 }%
796 \renewcommand*{\@pendR}[1]{%
797 \ifbypstart@R%
```

```

798 \unless\ifinstanzaR%
799 \global\line@numR=0%
800 \fi%
801 \fi%
802 \xright@appenditem{#1}\to\linesinpar@listR%
803 }%
804
805 %

```

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

```

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

```

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

```

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

```

IV.10 Writing to the line-list file

We have now defined all the counters, lists, and commands involved in reading the line-list file at the start of a section. Now we will cover the commands that `reledmac` uses within the text of a section to write commands out to the line-list.

`\linenum@outR` The file for right texts will be opened on output stream `\linenum@outR`.

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

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

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

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

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

`\new@lineL` The `\new@lineL` macro sends the `\@nl` command to the left text line-list file, to mark the start of a new text line.

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



```
869 }%
870 %
```

\new@lineR The `\new@lineR` macro sends the `\@nl` command to the right text line-list file, to mark the start of a new text line.

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

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

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

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

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

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

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

`\skipnumbering`

V Marking text for notes

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

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

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

V.1 Specific hooks and commands for notes

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

```
\newseries@par77 \newcommand{\newseries@par}[1]{%  
878 %
```

V.1.1 Notes to be printed on one side only

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

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

V.1.2 Tools specific to familiar footnotes

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

Managing correct number One problem with using familiar footnotes in parallel typesetting is the fact that the order of reading notes is not the same as the order they are typeset, because \LaTeX reads first all the notes on one side, then all the notes on the other side. Then, however, \LaTeX alternates between typesetting left-side note and right-side notes. Consequently, if we do nothing special, the note numbers are sorted in the reading order, not in the typesetting order. So we could obtain something like 1,3,2,5,4.

To prevent this problem, we use a two new counters by series. Every note, in parallel typesetting, has three associated counters.

1. A \LaTeX counter `footnoteX`. This the only one manipulated by user, and the only one finally printed.
2. A \TeX counter `footnoteX@reading`. Its value is incremented when reading the `\footnoteX` command in left or right side environments. It is used to get the correct footnote number from the `.aux` file to be typeset in the main text. This counter is already defined in `reledmac`, as it is also used for hyperlink.
3. A \LaTeX counter `footnoteX@typeset`. Its value is increased when inserting footnotes. Its value is directly used in the footnote, and is stored in the aux files to be used on the next run for the main text.

So here, we only defined the new counter.

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

Familiar footnotes without marks The `\footnoteXnomk` commands are for notes which are printed on the left side, while they are called in the right side. Basically, they set first toggle `\nomark@` to true, then call the `\footnoteX`. and finally add the footnote counter in the footnote counter list.

First, check the `nofamiliar` option of `reledmac`.
So declare the list.

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

Then, declare the `\footnoteXnomk` command.

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

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

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

Second, and last, step: store the footnote counter in the footnote counters list. We use some `\let`, because `\xright@appenditem` is difficult to use with `\expandafter`.

```
900 \letcs{\@tmp}{footnote#1@mk}%
901 \numdef\@tmpa{\csuse{c@footnote#1}}%
902 \global\xright@appenditem{\@tmpa}\to\@tmp%
903 \global\cslet{footnote#1@mk}{\@tmp}%
904 \global\protected@csxdef{hr@thefnmark#1}{\noexpand\@tmpa @\the\
csname footnote#1@series\endcsname}%
905 }%
906 %
```

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

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

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

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

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

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

Define the footnote mark and print it

```

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

```

Restore previous footnote counter and finally add space.

```

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

```

End of tools specific to familiar notes.

```

923 \fi
924 %

```

End of `\newseries@par`.

```

925 }%
926 %

```

V.1.3 Get correct footnote number

`\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.

```

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

```

V.2 Create hooks

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

```

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

```

```
948
949
950 %
```

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

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

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

V.4 Tools specific to \LaTeX 's classical footnotes

As users can use classical footnotes of \LaTeX (`\footnote`) in parallel texts, we must integrate the same tools to get correct number as for `reledmac`' footnotes (V.1.2 p. 50).

```
\footnote@reading57 \newcount\footnote@reading%
\footnote@typeset58 \newcounter{footnote@typeset}%
959 %
```

VI Pstart numbers dumping and restoration

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

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

```
\list@pstartL@pc60 \list@create{\list@pstartL@pc}%
\list@pstartR@pc61 \list@create{\list@pstartR@pc}%
962 %
```

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@pc63 \def\dump@pstartL@pc{%
\dump@pstartR@pc64   \xright@appenditem{\the\c@pstartL}\to\list@pstartL@pc%
965   \global\cslet{numberpstart@L\the\l@dnumpstartsL}{\ifnumberpstart}%
966   \global\cslet{numberline@L\the\l@dnumpstartsL}{\ifnumberline}%
967   \global\cslet{linenumannotationotherside@L\the\l@dnumpstartsL}{\iflinenumannotationotherside}%
```

```

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

```

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

```

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

```

VII Parallel environments

The initial set up for parallel processing is deceptively simple.

`pairs` `pages`

`chapterinpages` The `pairs` environment is for parallel columns and the `pages` environment for parallel pages.

```

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

```

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

```
1006 \newcommand{\AtBeginPairs}[1]{\gdef\at@begin@pairs{#1}}%
1007 \def\at@begin@pairs{}%
1008
1009 %
```

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.

```
1010 \newenvironment{pages}{%
1011   \l@dpairingtrue
1012   \l@dpagingtrue
1013   \initnumbering@quote
1014   \save@familiarfootnote@number%
1015   \if@ledgroup%
1016     \get@familiarfootnote@number%
1017   \fi%
1018   \save@section@number%
1019   \Lcolwidth=\textwidth%
1020   \Rcolwidth=\textwidth%
1021 }{%
1022   \l@dpairingfalse
1023   \l@dpagingfalse
1024   \global\Lcolwidth@pages=\Lcolwidth%Does not use \setlength, in case of
calc package redefiniton (cf. https://github.com/latex3/latex2e/issues/58)
1025   \global\Rcolwidth@pages=\Rcolwidth%
1026 }
1027
1028 %
```

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.

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

```

1044 \led@error@missing@numbering{Leftside}%
1045 \fi%
1046 \Leftsidehookend}
1047 %

```

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

```

\Leftsidehookend
\Rightsidehook
\Rightsidehookend
1048 \newcommand*\Leftsidehook{}
1049 \newcommand*\Leftsidehookend{}
1050 \newcommand*\Rightsidehook{}
1051 \newcommand*\Rightsidehookend{}
1052 %
1053 %

```

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

```

\Leftpagehook
1054 \newcommand*\Rightpagehook{}%
1055 \newcommand*\Leftpagehook{}%
1056 %

```

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.

```

1057 \newenvironment{Rightside}{%
1058 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else%
1059 \led@err@Rightside@PreviousNotPrinted%
1060 \fi%
1061 \ledRcoltrue
1062 \let\beginnumbering\beginnumberingR
1063 \let\endnumbering\endnumberingR
1064 \let\pausenumbering\pausenumberingR
1065 \let\resumenumbering\resumenumberingR
1066 \let\memorydump\memorydumpR
1067 \let\thepstart\thepstartR
1068 \let\pstart\pstartR
1069 \let\pend\pendR
1070 \let\ledpb\ledpbR
1071 \let\lednopb\lednopbR
1072 \let\lineation\lineationR
1073 \Rightsidehook
1074 \let\old@startstanza\@startstanza%
1075 \def\@startstanza[##1][##2]{\global\instanzaRtrue\old@startstanza
[##1][##2]}%
1076 }{%
1077 \ledRcolfalse
1078 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
1079 \led@error@missing@numbering{Rightside}%
1080 \fi%
1081 \Rightsidehookend
1082 }
1083 %
1084 %

```


VIII Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than straight onto the vertical list, and when the paragraph ends we slice the paragraph into its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

VIII.1 Boxes, counters, `\pstart` and `\pend`

`\num@linesR` Here are numbers and flags that are used internally in the course of the paragraph decomposition.
`\one@lineR`
`\par@lineR`

When we first form the paragraph, it goes into a box register, `\l@dLcolrawbox` or `\l@dRcolrawbox` for right text, instead of onto the current vertical list. The `\ifnumberedpar@` flag will be true while a paragraph is being processed in that way. `\num@lines(R)` will store the number of lines in the paragraph when it is complete. When we chop it up into lines, each line in turn goes into the `\one@line` or `\one@lineR` register, and `\par@line(R)` will be the number of that line within the paragraph.

```
1085 \newcount\num@linesR
1086 \newbox\one@lineR
1087 \newcount\par@lineR
1088 %
```

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

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

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

```
1089
1090 \newcounter{pstartL}
1091 \renewcommand{\thepstartL}{\bfseries\@arabic\c@pstartL. }
1092 \newcounter{pstartR}
1093 \renewcommand{\thepstartR}{\bfseries\@arabic\c@pstartR. }
1094
1095 \newcommandx*{\pstartL}[2][1,2,usedefault]{%
1096   \if@nobreak%
1097     \let\@oldnobreak\@nobreaktrue%
1098   \else%
1099     \let\@oldnobreak\@nobreakfalse%
1100   \fi%
1101   \@nobreaktrue%
1102   \ifluatex%
1103     \xdef\l@luatextextdir@L{\the\textdir}%
1104     \xdef\l@luatexpardir@L{\the\pardir}%
1105     \xdef\l@luatexbodydir@L{\the\bodydir}%
1106   \fi%
1107   \ifnumbering \else%
```

```

1108 \led@err@PstartNotNumbered%
1109 \beginnumbering%
1110 \fi%
1111 \ifnumberedpar%
1112 \led@err@PstartInPstart%
1113 \pend%
1114 \fi%
1115 %

```

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

```

1116 \ifpst@rtedL\else%
1117 \list@clear{\inserts@list}%
1118 \global\let\next@insert=\empty%
1119 \global\pst@rtedLtrue%
1120 \fi%
1121 \begingroup\everypar{}%
1122 %

```

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.

```

1123 \global\advance\l@dnumpstartsL \@ne%
1124 \global\advance\pstarts@read@L \@ne%
1125 \ifnum\l@dnumpstartsL>\l@dc@maxchunks%
1126 \led@err@TooManyPstarts%
1127 \global\l@dnumpstartsL=\l@dc@maxchunks%
1128 \fi%
1129 \global\setnamebox{l@dLcolrawbox\the\l@dnumpstartsL}=\vbox\bgroup%
1130 %

```

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.

```

1131 \l@dzeropenalties%
1132 \ifboolexpr{%
1133   bool{autopar}%
1134   and bool{by@autopar}%
1135 }%
1136 {}%
1137 {%
1138   \ifnumberpstart%
1139     \ifsidepstartnum%
1140     \else%
1141       \thepstartL%
1142     \fi%
1143   \fi%
1144 }%m
1145 \@at@start@every@pstart%
1146 \hsize=\Lcolwidth%
1147 \numberedpar@true%
1148 \iflabelpstart%
1149   \protected@edef\@currentlabel{\p@pstartL\thepstartL}%
1150 \fi%
1151 %

```

Dump the optional arguments

```

1152 \providetoggle{before@pstartL@the\l@dnumpstartsL @par}%
1153 \ifboolexpr{%
1154   test {\ifstrempy{#1}}%
1155   and test {\ifstrempy{#2}}%
1156 }%
1157 {%
1158   \ifat@every@pstart@star@%
1159   \global\togglefalse{before@pstartL@the\l@dnumpstartsL @par}%
1160   \else%
1161   \global\toggletrue{before@pstartL@the\l@dnumpstartsL @par}%
1162   \fi%
1163   \csgdef{before@pstartL@the\l@dnumpstartsL}{\at@every@pstart}%
1164 }%
1165 {%
1166   \ifstrempy{#1}{}%
1167   \global\toggletrue{before@pstartL@the\l@dnumpstartsL @par}%
1168   \csgdef{before@pstartL@the\l@dnumpstartsL}{\noindent#1}%
1169   }%
1170   \ifstrempy{#2}{}%
1171   {%
1172     \csgdef{before@pstartL@the\l@dnumpstartsL}{#2}%
1173     \global\togglefalse{before@pstartL@the\l@dnumpstartsL @par}%
1174     }%
1175   }%
1176   \at@every@pstart@call%
1177 %

```

Let \by@autopar to FALSE.

```

1178 \global\by@autoparfalse%
1179 %

```

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

```

1180 \ignorespaces%
1181 }
1182 %

```

The same for right side.

```

1183 \newcommand*{\pstartR}[2][1,2,usedefault]{%
1184   \if@nobreak%
1185     \let\@oldnobreak\@nobreaktrue%
1186   \else%
1187     \let\@oldnobreak\@nobreakfalse%
1188   \fi%
1189   \@nobreaktrue%
1190   \ifluatex%
1191     \xdef\l@luatextextdir@R{\the\textdir}%
1192     \xdef\l@luatexpardir@R{\the\pardir}%
1193     \xdef\l@luatexbodydir@R{\the\bodydir}%
1194   \fi%
1195   \ifnumberingR \else%
1196     \led@err@PstartNotNumbered%
1197   \beginnumberingR%

```

```

1198 \fi%
1199 \ifnumberedpar%
1200   \led@err@PstartInPstart%
1201   \pendR%
1202 \fi%
1203 \ifpstart@rtedR\else%
1204   \list@clear{\inserts@listR}%
1205   \global\let\next@insertR=\empty%
1206   \global\pstart@rtedRtrue%
1207 \fi%
1208 \begingroup\everypar{}%
1209 \global\advance\l@dnumpstartsR \@ne%
1210 \ifnum\l@dnumpstartsR>\l@dc@maxchunks%
1211   \led@err@TooManyPstarts%
1212   \global\l@dnumpstartsR=\l@dc@maxchunks%
1213 \fi%
1214 \global\setnamebox{\l@dRcolrawbox\the\l@dnumpstartsR}=\vbox\bgroup%
1215   \l@dzeropenalties%
1216   \ifboolexpr{%
1217     bool{autopar}%
1218     and bool{by@autopar}%
1219   }%
1220   {%
1221     {%
1222       \ifnumberpstart%
1223       \ifsidepstartnum\else%
1224         \thepstartR%
1225       \fi%
1226     }%
1227   }%
1228 \@at@start@every@pstart%
1229 \hsize=\Rcolwidth%
1230 \numberedpar@true%
1231 \iflabelpstart%
1232   \protected@edef\@currentlabel{\p@pstartR\thepstartR}%
1233 \fi%
1234 \providetoggle{before@pstartR@the\l@dnumpstartsR @par}%
1235 \ifboolexpr{%
1236   test {\ifstrempy{#1}}%
1237   and test {\ifstrempy{#2}}%
1238 }%
1239 {%
1240   \ifat@every@pstart@star%
1241   \global\togglefalse{before@pstartR@the\l@dnumpstartsR @par}%
1242   \else%
1243     \global\toggletrue{before@pstartR@the\l@dnumpstartsR @par}%
1244   \fi%
1245   \csgdef{before@pstartR@the\l@dnumpstartsR}{\at@every@pstart}%
1246 }%
1247 {%
1248   \ifstrempy{#1}{%
1249     \global\toggletrue{before@pstartR@the\l@dnumpstartsR @par}%
1250     \csgdef{before@pstartR@the\l@dnumpstartsR}{\noindent#1}%
1251   }%

```

```

1252     \ifstrempy{#2}{}%
1253     {%
1254         \csgdef{before@pstartR@the\l@dnumpstartsR}{#2}%
1255         \global\togglefalse{before@pstartR@the\l@dnumpstartsR @par}%
1256     }%
1257 }%
1258 \at@every@pstart@call%
1259 \global\by@autoparf@false%
1260 \ignorespaces%
1261 }
1262 %

```

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

```

1263 \newcommandx*{\pendL}[2][1,2,usedefault]{%
1264     \ifnumbering \else%
1265         \led@err@PendNotNumbered%
1266     \fi%
1267     \ifnumberedpar@ \else%
1268         \led@err@PendNoPstart%
1269     \fi%
1270 %

```

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.

```

1271     \@at@end@every@pend%
1272     \endgraf\global\num@lines=\prevgraf\egroup%
1273     \global\par@line=0%
1274     \get@firsthalf@antilabe@wd%
1275 %

```

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

```

1276     \endgroup%
1277     \ignorespaces%
1278     \@oldnbreak%
1279     \dump@pstartL@pc%
1280     \ifnumberpstart%
1281         \addtocounter{pstartL}{1}%
1282         \ifcontinuousnumberingwithcolumns%
1283             \addtocounter{pstart}{1}%
1284         \fi%
1285     \fi
1286     \parledgroup@beforenotes@save{L}%
1287 %

```

Dump content of the optional argument.

```

1288     \providetoggle{after@pendL@the\l@dnumpstartsL @par}%
1289     \ifboolexpr{%
1290         test {\ifstrempy{#1}}%
1291         and test {\ifstrempy{#2}}%
1292     }%
1293     {%
1294         \ifat@every@pend@star%

```

```

1295 \global\togglefalse{after@pendL@the\l@dnumpstartsL @par}%
1296 \else%
1297 \global\toggletrue{after@pendL@the\l@dnumpstartsL @par}%
1298 \fi%
1299 \csgdef{after@pendL@the\l@dnumpstartsL}{\at@every@pend}%
1300 }%
1301 {%
1302 \ifstrempy{#1}{}%
1303 \global\toggletrue{after@pendL@the\l@dnumpstartsL @par}%
1304 \csgdef{after@pendL@the\l@dnumpstartsL}{\noindent#1}%
1305 }%
1306 \ifstrempy{#2}{}%
1307 {%
1308 \csgdef{after@pendL@the\l@dnumpstartsL}{#2}%
1309 \global\togglefalse{after@pendL@the\l@dnumpstartsL @par}%
1310 }%
1311 }%
1312 }%
1313 %

```

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

```

1314 \newcommandx*{\pendR}[2][1,2,usedefault]{%
1315 \ifnumberingR \else%
1316 \led@err@PendNotNumbered%
1317 \fi%
1318 \ifnumberedpar@ \else%
1319 \led@err@PendNoPstart%
1320 \fi%
1321 \@at@end@every@pend%
1322 \endgraf\global\num@linesR=\prevgraf\egroup%
1323 \global\par@lineR=0%
1324 \get@firshalf@antilabe@wd%
1325 \endgroup%
1326 \ignorespaces%
1327 \@oldnobreak%
1328 \dump@pstartR@pc%
1329 \ifnumberpstart%
1330 \addtocounter{pstartR}{1}%
1331 \fi%
1332 \parledgroup@beforenotes@save{R}%
1333 \providetoggle{after@pendR@the\l@dnumpstartsR @par}%
1334 \ifboolexpr{%
1335 test {\ifstrempy{#1}}%
1336 and test {\ifstrempy{#2}}%
1337 }%
1338 {%
1339 \ifat@every@pend@star@%
1340 \global\togglefalse{after@pendR@the\l@dnumpstartsR @par}%
1341 \else%
1342 \global\toggletrue{after@pendR@the\l@dnumpstartsR @par}%
1343 \fi%
1344 \csgdef{after@pendR@the\l@dnumpstartsR}{\at@every@pend}}%
1345 {%

```

```

1346 \ifstrempy{#1}{}{%
1347 \csgdef{after@pendR@the\l@dnumpstartsR}{\noindent#1}%
1348 \global\toggletrue{after@pendR@the\l@dnumpstartsR @par}%
1349 }%
1350 \ifstrempy{#2}{}{%
1351 \csgdef{after@pendR@the\l@dnumpstartsR}{#2}%
1352 \global\togglefalse{after@pendR@the\l@dnumpstartsR @par}%
1353 }%
1354 }%
1355 }
1356
1357 %

```

\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.

```

1358 \newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
1359 \gdef\at@every@pstart@call{}%
1360 %

```

\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

```

1361 \newif\ifprint@last@after@pendL%
1362 \newif\ifprint@last@after@pendR%
1363 %

```

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

```

1364 \newbox\l@dleftbox
1365 \newbox\l@drightbox
1366
1367 %

```

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

\countRline

```

1368 \newcount\countLline
1369 \countLline \z@
1370 \newcount\countRline
1371 \countRline \z@
1372
1373 %

```

\@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

```

1374 \newcount\@donereallinesL
1375 \newcount\@donetotallinesL
1376 \newcount\@donereallinesR
1377 \newcount\@donetotallinesR
1378
1379 %

```

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

```

1380 \newcommand*\do@lineL{%
1381   \letcs{ifnumberpstart}{numberpstart@L\the\l@dpscl}%
1382   \letcs{ifnumberline}{numberline@L\the\l@dpscl}%
1383   \letcs{iflinenumannotationotherside}{linenumannotationotherside@L\the\l@dpscl}%
1384   \advance\countLline \@ne%
1385   \ifvbox\namebox{1@dLcolrawbox\the\l@dpscl}%
1386     {\vbadness=1000%
1387      \splittopskip=\z@%
1388      \do@lineLhook%
1389      \l@emptyd@ta%
1390      \global\setbox\one@line=\vsplit\namebox{1@dLcolrawbox\the\l@dpscl}%
1391        to\baselineskip}%
1392   \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\parledgroup@notes@startL}{}%
1393   \unvbox\one@line \global\setbox\one@line=\lastbox%
1394   \@writepageofparL%
1395   \getline@numL%
1396   \ifnum\@lock>\@ne%
1397     \inserthangingsymboltrue%
1398   \else%
1399     \inserthangingsymbolfalse%
1400   \fi%
1401   \setbox\l@dleftbox%
1402   \hb@xt@ \Lcolwidth{%
1403     \ifl@dhidnumber%
1404       \global\l@dhidnumberfalse%
1405       \f@x@l@cks%
1406     \else%
1407       \affixline@num%
1408     \fi%
1409     \xifinlist{\the\pstarts@typeset@L}{\eled@sections@@}%
1410     {\unless\ifshiftedpstarts%
1411       \add@apparatusL%
1412       \fi%
1413     }%
1414     {\print@lineL}%
1415   }%
1416   \add@penaltiesL%
1417   \global\advance\@donereallinesL\@ne%
1418   \global\advance\@donetotallinesL\@ne%
1419 \else%
1420   \iflinenumLevenifblank
1421   \setbox\l@dleftbox \hb@xt@ \Lcolwidth{%

```



```

1422     \new@lineL%
1423     \l@emptyd@ta%
1424     \getline@numL%
1425     \affixline@num%
1426     \l@dld@ta%
1427     \hspace*{\Lcolwidth}%
1428     \ledrlfill\l@drd@ta%
1429   }%
1430 \else%
1431   \setbox\l@dleftbox \hb@xt@ \Lcolwidth{%
1432     \ifcontinuousnumberingwithcolumns%
1433       \new@lineL%
1434       \getline@numL%
1435       \fi%
1436       \hspace*{\Lcolwidth}%
1437     }%
1438   \fi%
1439   \global\advance\@donetotallinesL\@ne%
1440 \fi%
1441 }%
1442
1443
1444 %

```

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

```

1445 \def\print@lineL{%
1446   \affixpstart@numL%
1447   \l@dld@ta%
1448   \add@apparatusL%
1449   \l@dlsn@te%
1450   \do@insidelineLhook%
1451   \csuse{insidethis@the\absline@num @the\section@num}%
1452   \global\csundef{insidethis@the\absline@num @the\section@num}%
1453   \hb@xt@ \Lcolwidth{\ledllfill\hb@xt@ \wd\one@line{%
1454     \ifluatex%
1455       \textdir\l@luatextextdir@L%
1456     \fi%
1457     \new@lineL%
1458     \inserthangingsymbolL%
1459     \l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta%
1460   \l@drsn@te}}%
1461
1462 %

```

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

```

1463 \def\print@eledsectionL{%
1464   \disable@edindex%
1465   \addtocounter{pstartL}{-1}%
1466   \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{ }
1467   \ifdefstring{\@eledsectmark}{L}{ }\ledsectnomark}
1468   \numdef{\temp@}{\pstarts@typeset@L-1}%

```

```

1469 \xifinlist{\temp@}{\eled@sections@}{\@nobreaktrue}{\@nobreakfalse}%
1470 \@eled@sectioningtrue%
1471 \bgroup%
1472 \ifluatex%
1473 \textdir\l@luatextextdir@L%
1474 \pardir\l@luatexpardir@L%
1475 \bodydir\l@luatexbodydir@L%
1476 \ifdefstring{\l@luatextextdir@L}{TRT}{\@RTLtrue}{}%
1477 \fi%
1478 \csuse{eled@sectioning@the\pstarts@typeset@L}%
1479 \egroup%
1480 \@eled@sectioningfalse%
1481 \global\csundef{eled@sectioning@the\pstarts@typeset@L}%
1482 \ifRTL%
1483 \hspace{-3\paperwidth}%
1484 {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1485 \else%
1486 \hspace{3\paperwidth}%
1487 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1488 \fi%
1489 \vskip\eledsection@correcting@skip%
1490 \restore@edindex%
1491 }
1492
1493 %

```

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

```

1494 \newcommand{\add@apparatusL}{%
1495 \if@firstlineofpage%
1496 \set@Xtxtbeforenotes%
1497 \set@txtbeforenotesX%
1498 \global\@firstlineofpagefalse%
1499 \fi%
1500 \ifdefstring{\ms@data@position}{msdata-regular}{%
1501 \insert@msdata%
1502 \add@inserts%
1503 \add@Xgroupbyline%
1504 }{%
1505 \add@inserts%
1506 \add@Xgroupbyline%
1507 \insert@msdata%
1508 }%
1509 \affixside@note%
1510 }%
1511 %

```

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

`\dolineRhook`

`\doinsidelineLhook`

`\doinsidelineRhook`

```

1512 \newcommand*{\dolineLhook}[1]{\gdef\do@lineLhook{#1}}%
1513 \newcommand*{\dolineRhook}[1]{\gdef\do@lineRhook{#1}}%
1514 \newcommand*{\doinsidelineLhook}[1]{\gdef\do@insidelineLhook{#1}}%

```

```

1515 \newcommand*\doinsidelineRhook}[1]{\gdef\do@insidelineRhook{#1}}%
1516
1517 %

```

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

```

\do@lineRhook
\do@insidelineLhook 1518 \newcommand*\do@lineLhook{}
\do@insidelineRhook 1519 \newcommand*\do@lineRhook{}
1520 \newcommand*\do@insidelineLhook{}
1521 \newcommand*\do@insidelineRhook{}
1522
1523 %

```

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

```

1524 \newcommand*\do@lineR{%
1525   \let\linenumrepL\linenumrep%
1526   \let\sublinenumrepL\sublinenumrep%
1527   \let\linenumrep\linenumrepR%
1528   \let\sublinenumrep\sublinenumrepR%
1529   \letcs\ifnumberpstart\ifnumberpstart@R\the\l@dpscr}%
1530   \letcs\ifnumberline\ifnumberline@R\the\l@dpscr}%
1531   \letcs\iflinenumannotationotherside\iflinenumannotationotherside@R\the\
l@dpscr}%
1532   \ledRcol@true%
1533   \advance\countRline \@ne%
1534   \ifvbox\namebox{l@dRcolrawbox\the\l@dpscr}%
1535     {\vbadness=10000%
1536     \splittopskip=\z@%
1537     \do@lineRhook%
1538     \l@demtyd@ta%
1539     \global\setbox\one@lineR=\vsplit\namebox{l@dRcolrawbox\the\l@dpscr}%
1540     to\baselineskip}%
1541   \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startR}{}%
1542   \unvbox\one@lineR \global\setbox\one@lineR=\lastbox%
1543   \@writepageofparR%
1544   \getline@numR%
1545   \ifnum\@lockR>\@ne%
1546     \inserthangingsymbolRtrue%
1547   \else%
1548     \inserthangingsymbolRfalse%
1549   \fi%
1550   \setbox\l@drightbox%
1551   \hb@xt@ \Rcolwidth{%
1552     \ifl@dhidenumber%
1553       \global\l@dhidenumberfalse%
1554       \f@x@l@cksR%
1555     \else%
1556       \affixline@numR%
1557     \fi%
1558   \xifinlist{\the\l@dpscr}{\eled@sectionsR@}%
1559   {\unless\ifshiftedpstarts%

```

```

1560     \add@apparatusR%
1561     \fi%
1562   }%
1563   {\print@lineR}%
1564 }%
1565 \add@penaltiesR%
1566 \global\advance\@donereallinesR\@ne%
1567 \global\advance\@donetotallinesR\@ne%
1568 \else%
1569   \iflinenumberRevenifblank%
1570     \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1571       \new@lineR%
1572       \l@demptyd@ta%
1573       \getline@numR%
1574       \affixline@numR%
1575       \l@dld@ta%
1576       \hspace*\Rcolwidth}%
1577     \ledrlfill\l@drd@ta%
1578   }%
1579 \else%
1580   \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1581     \ifcontinuousnumberingwithcolumns%
1582       \new@lineR%
1583       \getline@numR%
1584     \fi%
1585     \hspace*\Rcolwidth}%
1586   }%
1587 \fi%
1588 \global\advance\@donetotallinesR\@ne%
1589 \fi%
1590 \ledRcol@false%
1591 \let\linenumrep\linenumrepL%
1592 \let\sublinenumrep\sublinenumrepL%
1593 }
1594
1595
1596 %

```

```

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

```

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

```

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

```

```

1608 \add@insertsR%
1609 \add@Xgroupbyline%
1610 \insert@msdata%
1611 }%
1612 \affixside@noteR%
1613 }%
1614 %

```

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.

```

1615 \newcommand*{\getline@numR}{%
1616 \global\advance\absline@numR \@ne
1617 \do@actionsR
1618 \do@ballastR
1619 \ifledgroupnotesR@else
1620 \ifnumberline
1621 \ifsublines@R
1622 \ifnum\sub@lockR<\tw@
1623 \global\advance\subline@numR \@ne
1624 \fi
1625 \else
1626 \ifnum\@lockR<\tw@
1627 \global\advance\line@numR \@ne
1628 \global\subline@numR \z@
1629 \fi
1630 \fi
1631 \fi
1632 \fi
1633 }
1634 \newcommand*{\getline@numL}{%
1635 \global\advance\absline@num \@ne
1636 \do@actions
1637 \do@ballast
1638 \ifledgroupnotesL@else
1639 \ifnumberline
1640 \ifsublines@
1641 \ifnum\sub@lock<\tw@
1642 \global\advance\subline@num \@ne
1643 \fi
1644 \else
1645 \ifnum\@lock<\tw@
1646 \global\advance\line@num \@ne
1647 \global\subline@num \z@
1648 \fi
1649 \fi
1650 \fi
1651 \fi
1652 }
1653
1654
1655 %

```

`\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.

```

1656 \newcommand*{\do@ballastR}{\global\ballast@count=\z@
1657   \begingroup
1658     \advance\absline@numR \@ne
1659     \ifnum\next@actionlineR=\absline@numR
1660       \ifnum\next@actionR>-1001
1661         \global\advance\ballast@count by -\c@ballast
1662       \fi
1663     \fi
1664   \endgroup}
1665 %

```

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

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

`\do@actions@nextR`

```

1666
1667 \newif\ifl@dskipversenumberR
1668 \newcommand*{\do@actions@fixedcodeR}{%
1669   \ifcase\@l@dtmpcnta%
1670     \or% % 1001 = starting sublineation
1671       \global\sublines@Rtrue
1672     \or% % 1002 = ending sublineation
1673       \global\sublines@Rfalse
1674     \or% % 1003 = starting locking number
1675       \ifnum\@lockR=\thr@@% If there was a \endlock (implicit or explicit)
just before
1676         \global\@lockR=\tw@%
1677       \else%
1678         \global\@lockR=\@ne%
1679       \fi%
1680     \or% % 1004 = ending locking number
1681       \ifnum\@lockR=\tw@
1682         \global\@lockR=\thr@@
1683       \else
1684         \global\@lockR=\z@
1685       \fi
1686     \or% % 1005 = starting locking subnumber
1687       \global\sub@lockR=\@ne
1688     \or% % 1006 = ending locking subnumber
1689       \ifnum\sub@lockR=\tw@
1690         \global\sub@lockR=\thr@@
1691       \else
1692         \global\sub@lockR=\z@
1693       \fi
1694     \or% % 1007 = skipping numbering
1695       \l@dskipnumbertrue
1696     \or% % 1008 = skipping numbering in stanza
1697       \l@dskipversenumberRtrue%
1698     \or% % 1009 = hiding number
1699       \l@dhidnumbertrue%

```

```

1700 \or%                                % 1010 = inserting msdata
1701   \add@msdata%
1702 \else%
1703   \led@warn@BadAction
1704 \fi%
1705 }
1706
1707
1708 \newcommand*{\do@actionsR}{%
1709   \global\let\do@actions@nextR=\relax
1710   \@l@dttempcntb=\absline@numR
1711   \ifnum\@l@dttempcntb<\next@actionlineR\else
1712     \ifnum\next@actionR>-1001\relax
1713       \ifboolexpr{%
1714         bool{resumenumberingR@start}%
1715         and test {\ifdimgreater{\pagedepth}{\z@}}%
1716       }%
1717       {}%
1718       {\@firstlineofpageRtrue}%
1719       \global\page@numR=\next@actionR
1720       \ifcsdef{reset@lineR\the\absline@numR @\the\section@numR}%
1721         {%
1722           \global\line@numR=\z@ \global\subline@numR=\z@%
1723         }%
1724         {}%
1725       \global\resumenumberingR@startfalse%
1726       \add@msdata@firstlineofpage%
1727     \else
1728       \ifnum\next@actionR<-4999\relax%
1729         \@l@dttempcnta=-\next@actionR
1730         \advance\@l@dttempcnta by -5001\relax
1731         \ifsublines@R
1732           \global\subline@numR=\@l@dttempcnta
1733         \else
1734           \global\line@numR=\@l@dttempcnta
1735         \fi
1736       \else
1737         \@l@dttempcnta=-\next@actionR
1738         \advance\@l@dttempcnta by -1000\relax
1739         \do@actions@fixedcodeR
1740       \fi
1741     \fi
1742     \ifx\actionlines@listR\empty
1743       \gdef\next@actionlineR{1000000}%
1744     \else
1745       \glp\actionlines@listR\to\next@actionlineR
1746       \glp\actions@listR\to\next@actionR
1747       \global\let\do@actions@nextR=\do@actionsR
1748     \fi
1749   \fi
1750   \do@actions@nextR}
1751
1752 %

```

VIII.4 Line number printing

```

\l@dcalcnm \affixline@numR is the right text version of the \affixline@num macro.
\ch@cksub@l@ckR
\ch@ck@l@ckR1753
\fx@l@cksR1754
\affixline@numR1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803

```

```

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

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

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

\newcommand*{\fx@l@cksR}{%
  \ifcase\@lockR
  \or
    \global\@lockR \tw@
  \or \or

```



```

1804 \global\@lockR \z@
1805 \fi
1806 \ifcase\sub@lockR
1807 \or
1808 \global\sub@lockR \tw@
1809 \or \or
1810 \global\sub@lockR \z@
1811 \fi}
1812
1813
1814 \let\linenumberlistR\empty%
1815 \newcommand*\affixline@numR}{%
1816 \ifledgroupnotesR@else\ifnumberline
1817 \ifl@dskipnumber
1818 \global\l@dskipnumberfalse
1819 \else
1820 \ifsublines@R
1821 \l@dttempcntb=\subline@numR
1822 \l@dcalcnm{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR
}%
1823 \ch@cksub@l@ckR
1824 \else
1825 \l@dttempcntb=\line@numR
1826 \ifx\linenumberlistR\empty%
1827 \l@dcalcnm{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%
1828 \else
1829 \l@dttempcnta=\line@numR
1830 \edef\rem@inderR{\linenumberlistR,\number\line@numR,}%
1831 \edef\sc@n@list{\def\noexpand\sc@n@list
1832 ###1,\number\l@dttempcnta,###2|\def\noexpand\rem@inderR{###2}}
%
1833 \sc@n@list\expandafter\sc@n@list\rem@inderR|%
1834 \ifx\rem@inderR\empty\advance\l@dttempcnta\@ne\fi%
1835 \fi
1836 \ch@ck@l@ckR
1837 \fi
1838 \ifnum\l@dttempcnta=\l@dttempcntb
1839 \ifl@dskipversenumberR\else
1840 \iftwocolumn
1841 \if@firstcolumn
1842 \gdef\l@dld@ta{\llap{\leftlinenumR}}%
1843 \iflinenumannotationotherside%
1844 \gdef\l@drd@ta{\rlap{\rightlinenumannotationR}}%
1845 \fi%
1846 \else
1847 \gdef\l@drd@ta{\rlap{\rightlinenumR}}%
1848 \iflinenumannotationotherside%
1849 \gdef\l@dld@ta{\llap{\leftlinenumannotationR}}%
1850 \fi%
1851 \fi
1852 \else
1853 \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columnsR}{\m@ne}}}%
1854 {\l@dttempcntb=\line@margin@columnsR}%

```

```

1855     {\@l@tempcntb=\line@marginR}%
1856     \ifnum\@l@tempcntb>\@ne
1857       \advance\@l@tempcntb by \numexpr\csuse{@page@forlineR@the\
section@numR @the\absline@numR}+0\relax%
1858     \fi
1859     \ifboolexpr{%
1860       bool {l@dprintingcolumns}%
1861       and (%
1862         (test {\ifdefstring{\linenum@OnlyPages@ForColumnsR}{left}}%
1863         and test {\ifnumodd{\csuse{@page@forlineR@the\
the\absline@numR}+0}}%
1864         )%
1865         or%
1866         (test {\ifdefstring{\linenum@OnlyPages@ForColumnsR}{right}}%
1867         and not test {\ifnumodd{\csuse{@page@forlineR@the\
section@numR @the\absline@numR}+0}}%
1868         )%
1869       )%
1870     }%
1871     {}%
1872     {%
1873       \ifodd\@l@tempcntb%
1874         \gdef\l@drd@ta{\rlap{\rightlinenumR}}%
1875         \iflinenumannotationotherside%
1876           \gdef\l@dld@ta{\llap{\leftlinenumannotationR}}%
1877         \fi%
1878       \else%
1879         \gdef\l@dld@ta{\llap{\leftlinenumR}}%
1880         \iflinenumannotationotherside%
1881           \gdef\l@drd@ta{\rlap{\rightlinenumannotationR}}%
1882         \fi%
1883       \fi%
1884     }%
1885     \fi
1886     \fi
1887     \fi
1888     \f@x@l@cksR
1889 \fi
1890 \fi
1891 \fi}
1892 %

```

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@numL93
\affixpstart@numR
\leftpstartnumR
\rightpstartnumR
\leftpstartnumL
\rightpstartnumL
\ifpstartnumR

```

```

1894 \newcommand*{\affixpstart@numL}{%
1895 \ifsidepstartnum
1896 \if@twocolumn
1897   \if@firstcolumn
1898     \gdef\l@dld@ta{\llap{\leftpstartnumL}}%
1899   \else
1900     \gdef\l@dld@ta{\rlap{\rightpstartnumL}}%
1901   \fi
1902 \else
1903   \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columns}{\m@ne}}}%
1904     {\@l@tempcntb=\line@margin@columns}%
1905     {\@l@tempcntb=\line@margin}%
1906   \ifnum\@l@tempcntb>\@ne
1907     \advance\@l@tempcntb \page@num
1908   \fi
1909   \ifodd\@l@tempcntb
1910     \gdef\l@dld@ta{\rlap{\rightpstartnumL}}%
1911   \else
1912     \gdef\l@dld@ta{\llap{\leftpstartnumL}}%
1913   \fi
1914 \fi
1915 \fi
1916 }
1917 \newcommand*{\affixpstart@numR}{%
1918 \ifsidepstartnum
1919 \if@twocolumn
1920   \if@firstcolumn
1921     \gdef\l@dld@ta{\llap{\leftpstartnumR}}%
1922   \else
1923     \gdef\l@dld@ta{\rlap{\rightpstartnumR}}%
1924   \fi
1925 \else
1926   \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columnsR}{\m@ne}}}%
1927     {\@l@tempcntb=\line@margin@columnsR}%
1928     {\@l@tempcntb=\line@marginR}%
1929   \ifnum\@l@tempcntb>\@ne
1930     \advance\@l@tempcntb \page@numR
1931   \fi
1932   \ifodd\@l@tempcntb
1933     \gdef\l@dld@ta{\rlap{\rightpstartnumR}}%
1934   \else
1935     \gdef\l@dld@ta{\llap{\leftpstartnumR}}%
1936   \fi
1937 \fi
1938 \fi
1939 }
1940
1941 \newcommand*{\leftpstartnumL}{
1942 \ifpstartnum
1943 \thepstartL
1944 \kern\linenumsep\global\pstartnumfalse\fi
1945 }

```

```

1946 \newcommand*{\rightpstartnumL}{
1947 \ifpstartnum\kern\linenumsep
1948 \thepstartL
1949 \global\pstartnumfalse\fi
1950 }
1951 \newif\ifpstartnumR
1952 \pstartnumRtrue
1953 \newcommand*{\leftpstartnumR}{
1954 \ifpstartnumR
1955 \thepstartR
1956 \kern\linenumsep\global\pstartnumRfalse\fi
1957 }
1958 \newcommand*{\rightpstartnumR}{
1959 \ifpstartnumR\kern\linenumsep
1960 \thepstartR
1961 \global\pstartnumRfalse\fi
1962 }
1963 %

```

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.

```

1964 \list@create{\inserts@listR}
1965 %

```

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

```

1966 \newcommand*{\add@insertsR}{%
1967 \global\let\add@inserts@nextR=\relax
1968 \ifx\inserts@listR\empty \else
1969 \ifx\next@insertR\empty
1970 \ifx\insertlines@listR\empty
1971 \global\noteschanged@true
1972 \gdef\next@insertR{100000}%
1973 \else
1974 \gl@p\insertlines@listR\to\next@insertR
1975 \fi
1976 \fi
1977 \ifnum\next@insertR=\absline@numR
1978 \gl@p\inserts@listR\to\@insertR
1979 \@insertR
1980 \global\let\@insertR=\undefined
1981 \global\let\next@insertR=\empty
1982 \global\let\add@inserts@nextR=\add@insertsR
1983 \fi
1984 \fi
1985 \add@inserts@nextR}
1986
1987 %

```

VIII.7 Penalties

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

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

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

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

```
1988 \newcommand*{\add@penaltiesL}{}
1989 \newcommand*{\add@penaltiesR}{}
1990
1991 %
```

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.

```

1992 \newcommand*{\flush@notesR}{%
1993   \iftoggle{notfirststrun@\jobname.\extensionchars\the\section@numR R}{%
1994     \@xloop%
1995     \ifx\inserts@listR\empty \else%
1996       \gl@p\inserts@listR\to\@insertR%
1997       \@insertR%
1998       \global\let\@insertR=\undefined%
1999     \repeat%
2000   }-{}%
2001 }%
2002
2003
2004 %

```

IX Footnotes

IX.1 Footnotes output specific to \Pages

```

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

```

The `\Xonlyside` and `\onlysideX` hooks for `\Pages` allow notes to be printed either in left or right pages only. The implementation of such features is delegated to `\print@Xnotes@forpages`, which replaces `\print@Xnotes` inside `\Pages`. Here is how we proceed⁶:

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

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

```
2005 \newcommand\print@Xnotes@forpages[1]{%
2006 %
```

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

```
2007   \ifcseempty{Xonlyside@#1}{%
2008     \csuse{#1footstart}{#1}%
2009     \csuse{#1footgroup}{#1}%
2010   }%
2011 %
```

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

```
2012   {%
2013     \ifboolexpr{%
2014       ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page
2015       }})%
2016       or%
2017       (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}))%
2018     }%
2019 %
```

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

```
2019   {%
2020     \csuse{#1footstart}{#1}%
2021     \csuse{#1footgroup}{#1}%
2022   }%
2023 %
```

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.

```
2023     \global\count\csuse{#1footins}=\z@%
2024     \global\skip\csuse{#1footins}=\z@%
2025     \global\dimen\csuse{#1footins}=\maxdimen%
2026   }%
2027 %
```

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

```
2028   {%
2029     \restore@Xnotes@settings{#1}%
2030   }%
2031 %
```

End of \print@Xnotes@forpages.

```
2032   }%
2033 }%
2034 %
```

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

```
2035 \newcommand{\restore@Xnotes@settings}[1]{%
2036   \unless\ifnocritical@
2037     \global\count\csuse{#1footins}=\csuse{default@#1footins}%
2038     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
2039     \bgroup%
```

```

2040 \csuse{Xnotefontsize@#1}%
2041 \global\dimen\csuse{#1footins}=\csuse{Xmaxhnotes@#1}%
2042 \egroup%
2043 \global\setnamebox{#1footins@kept}=\box\namebox{#1footins}%
2044 \fi
2045 }%
2046 %

```

And now, the same for familiar footnotes.

```

2047 \newcommand\print@notesX@forpages[1]{%
2048 \ifcsemtyp{onlysideX@#1}{%
2049 \csuse{footstart#1}{#1}%
2050 \csuse{footgroup#1}{#1}%
2051 }%
2052 {%
2053 \ifboolexpr{%
2054 ((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page
2055 }}))%
2056 or%
2057 (test {\ifcsstring{onlysideX@#1}{R}} and test{\ifnumodd{\c@page}}))%
2058 }%
2059 {%
2060 \csuse{footstart#1}{#1}%
2061 \csuse{footgroup#1}{#1}%
2062 \global\count\csuse{footins#1}=\z@%
2063 \global\skip\csuse{footins#1}=\z@%
2064 \global\dimen\csuse{footins#1}=\maxdimen%
2065 }%
2066 {%
2067 \restore@notesX@settings{#1}%
2068 }%
2069 }%
2070 \newcommand{\restore@notesX@settings}[1]{%
2071 \unless\ifnofamiliar@
2072 \global\count\csuse{footins#1}=\csuse{default@footins#1}%
2073 \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
2074 \bgroup%
2075 \csuse{Xnotefontsize@#1}%
2076 \global\dimen\csuse{footins#1}=\csuse{maxhnotesX@#1}%
2077 \egroup%
2078 \global\setnamebox{footins#1@kept}=\box\namebox{footins#1}%
2079 \fi
2080 }%
2081 %

```

`\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.

```

2082 \newcommand{\insert@notes@for@onlyside}{%
2083 \def\do##1{%
2084 \unless\ifnocritical@
2085 \ifvoid\csuse{##1footins@kept}\else%
2086 \expandafter\insert\csname ##1footins\endcsname%

```



```

2087     \bgroup%
2088     \unvnamebox{##1footins@kept}%
2089     \egroup%
2090     \fi%
2091     \restore@Xnotes@settings{##1}%
2092     \fi%
2093     \unless\ifnofamiliar@%
2094     \ifvoid\csuse{footins##1@kept}\else%
2095     \expandafter\insert\csname footins##1\endcsname%
2096     \bgroup%
2097     \unvnamebox{footins##1@kept}%
2098     \egroup%
2099     \fi%
2100     \restore@notesX@settings{##1}%
2101     \fi%
2102   }%
2103   \dolistloop{\@series}%
2104 }%
2105 %

```

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
2106 \newcount\edtext@now%
2107 \newcount\edtext@later%
2108 %

2109 \newcommand{\edtextlater}[2]{%#1 lemma, #2 critical notes
2110 %

```

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

```

2111 \global\advance\edtext@later by \@ne%
2112 %

2113 % 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}.
2114 \advance\@edtext@level by \@ne%
2115 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
2116     {\global\let\sw@inthisedtext\empty}%
2117     {\expandafter\glp\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
2118 %

```

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

```

2119 %
2120 \csxdef{edtext@later@\the\edtext@later}{%
2121 %

```

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

```

2122 \noexpand\edtext{%
2123 }%
2124 %

```

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

```

2125 {%
2126 \unexpanded{%
2127 \lemma{\no@expands #1}%
2128 }%
2129 \noexpand\xxref{start:edtext:later:\the\edtext@later}{end:edtext:
later:\the\edtext@later}%
2130 \noexpand\linenum{||||\edfont@info}%
2131 \unexpanded{\def\sw@inthisedtext}{\expandonce{\sw@inthisedtext}}%
2132 %

```

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

```

2133 \ifledRcol%
2134 \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@true}}%
2135 {%
2136 \noexpand\setcounter{pstartR}{\noexpand\xpstartref{start:edtext:
later:\the\edtext@later}}%
2137 }%
2138 \else%
2139 \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@false}}%
2140 \unexpanded{\appto\@beforeinsertofthisedtext}%
2141 {%
2142 \noexpand\setcounter{pstartL}{\noexpand\xpstartref{start:edtext:
later:\the\edtext@later}}%
2143 }%
2144 \fi%
2145 %

```

And the footnote command of this `\edtextlater`.

```

2146 \unexpanded{#2}%
2147 }%
2148 }%
2149 %

```

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`

```

2150 \edlabel{start:edtext:later:\the\edtext@later}%
2151 \flag@start@later%
2152 \bgroup%
2153 \def\lemma##1{%
2154 \ifledRcol%
2155 \write\linenum@outR{\string\@lemma}%
2156 \else%
2157 \write\linenum@out{\string\@lemma}%
2158 \fi%

```

```

2159 }%
2160 \renewcommand{\do}[1]{\expandafter\renewcommandx\csname ##1footnote\
endcsname[2][1,usedefault]{}\unskip}%\unskip because of a spurious space in
\newcommandx
2161 \dolistloop{\@series}%
2162 #2%
2163 \egroup%
2164 \showlemma{#1}%
2165 \edlabel{end:edtext:later:\the\edtext@later}%
2166 \flag@end@later%
2167 %

```

We decrease the counter increased at the beginning.

```

2168 \advance\@edtext@level by -\@ne%
2169 }%
2170 %

```

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

```

2171 \newcommand{\edtextnow}[0]{%
2172 \global\advance\edtext@now by \@ne
2173 \csuse{edtext@later@\the\edtext@now}%
2174 }%
2175 %

```

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.

```

2176 \list@create{\labelref@listR}
2177 %
2178 %

```

`\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`.

```

2179 \def\l@dmake@labelsR#1|#2|#3|#4|#5|#6|#7{%
2180 \expandafter\ifx\csname the@label\csuse{XR@prefix}#7\endcsname%
\relax%
2181 \else%
2182 \led@warn@DuplicateLabel{\csuse{XR@prefix}#7}%
2183 \fi%
2184 \csgdef{the@label\csuse{XR@prefix}#7}{#1|#2|#3|#4|#5|#6|\@Rlineflag}%
2185 \global\providetoggle{label@#7@ledRcol}%False is the default value of
2186 this toggle, which tells us whether a label is linked to the right or left
side
2187 \global\toggletrue{label@#7@ledRcol}%
2188 \ignorespaces}
2189 \AtBeginDocument{%
2190 \def\l@dmake@labelsR#1|#2|#3|#4|#5|#6|#7{%
2191 }

```

```
2192 %
2193 %
```

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

It is defined on `reledmac`.

XI Sidenotes

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

```
\sidenote@marginR Specifies which margin sidenotes can be in.
\sidenotemargin*
\sidenotemarginR
2194 \WithSuffix\newcommand\sidenotemargin*[1]{%
2195 \l@dgetsidenote@margin{#1}
2196 \global\sidenote@marginR=\@l@dttempcntb
2197 \global\sidenote@margin=\@l@dttempcntb
2198 }
2199 \newcommand{\sidenotemarginR}[1]{%
2200 \l@dgetsidenote@margin{#1}%
2201 \global\sidenote@marginR=\@l@dttempcntb%
2202 }%
2203 \newcount\sidenote@marginR
2204 \global\sidenote@marginR=\@ne%
2205
2206 \ifmovecolumnsonrightpage%
2207 \sidenotemargin{inner}%
2208 \sidenotemarginR{outer}%
2209 \fi%
2210
2211 %
```

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

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

```
2212 \newdimen\@morespace@leftnote@rightcolumn%
2213 \newdimen\@morespace@rightnote@leftcolumn%
2214
2215
2216
2217 \newcommand{\get@sidenote@morespace@columns}{%
2218 %
```

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

```

2219 \if@sidenotesmarginpage%
2220 %

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

2221 \global\@morespace@leftnote@rightcolumn=\dimexpr\intercolumns@width + \
Lcolwidth\relax%
2222 \global\@morespace@rightnote@leftcolumn=\dimexpr\intercolumns@width + \
Rcolwidth\relax%
2223 %

If \if@sidenotesmarginpage is False

2224 \else%
2225 \global\@morespace@leftnote@rightcolumn=\z@%
2226 \global\@morespace@rightnote@leftcolumn=\z@%
2227 %

End of \get@sidenote@morespace@columns.

2228 \fi%
2229 }%
2230 %

```

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

```

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

```

```

2259 \ifodd\@l@dttempcntb%
2260 \setl@drp@rbox{\sidenotecontent@}%
2261 \gdef\sidenotecontent@{}%
2262 \numdef{\itemcount@}{0}%
2263 \dolistloop{\l@dcsnotetext@l}%
2264 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
2265 \setl@dlp@rbox{\sidenotecontent@}%
2266 \else%
2267 \setl@dlp@rbox{\sidenotecontent@}%
2268 \gdef\sidenotecontent@{}%
2269 \numdef{\itemcount@}{0}%
2270 \dolistloop{\l@dcsnotetext@r}%
2271 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
2272 \setl@drp@rbox{\sidenotecontent@}%
2273 \fi%
2274 \fi%
2275 \fi%
2276 \advance\@edindex@fornote@\m@ne%
2277 }
2278
2279 %

```

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.

```

\inserthangingsymbolL80 \newif\ifinserthangingsymbolR
\inserthangingsymbolR81 \newcommand{\inserthangingsymbolL}{%
2282 \ifinserthangingsymbol%
2283 \ifinstanzaL%
2284 \@hangingsymbol%
2285 \fi%
2286 \ifinstanzaL%
2287 \@hangingsymbol%
2288 \fi%
2289 \fi%
2290 }%
2291 \newcommand{\inserthangingsymbolR}{%
2292 \ifinserthangingsymbolR%
2293 \ifinstanzaR%
2294 \@hangingsymbol%
2295 \fi%
2296 \ifinstanzaR%
2297 \@hangingsymbol%
2298 \fi%
2299 \fi%
2300 }%
2301 %

```

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.

```

2302 \chardef\next=\catcode`&
2303 \catcode`\&=\active
2304
2305 %

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

2306 \newenvironmentx{astanza}[2][1,2,usedefault]{%
2307 \ifledRcol%
2308 \global\inastanzaRtrue%
2309 \else%
2310 \global\inastanzaLtrue%
2311 \fi%
2312 \catcode`\&=\active
2313 \global\stanza@count\@ne\stanza@modulo\@ne
2314 \@advancestanzanumber%
2315 \newdimen\parindent@beforestanza%
2316 \parindent@beforestanza=\parindent%Keep in memory the standard \parindent
2317 \ifnum\usernamecount{sza@00}=\z@
2318 \let\stanza@hang\relax
2319 \let\endlock\relax
2320 \else
2321 \rightskip\z@ plus 1fil\relax
2322 \fi
2323 \ifnum\usernamecount{szp@00}=\z@
2324 \let\sza@penalty\relax
2325 \fi
2326 \def&{%
2327 \endlock\mbox{}%
2328 \sza@penalty
2329 \global\advance\stanza@count\@ne
2330 \@astanza@line}%
2331 \def\&{\@stopastanza}%
2332 \ifboolexpr{%
2333 not test{\ifdefvoid{\@at@every@stanza}}%
2334 and test{\ifstrempy{#1}}%
2335 and test{\ifstrempy{#2}}}%
2336 {\pstart [] [\@at@every@stanza] \@at@start@every@stanza}%
2337 {\pstart [#1] [#2] \@at@start@every@stanza}%
2338 \@astanza@line
2339 \@insertstanzanumber%
2340 \let\par\relax\ignorespaces%No paragraph in verses
2341 }{%
2342 \global\undef\parindent@beforestanza%
2343 }%
2344
2345 %

```

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

```

2346 \newcommandx{\@stopastanza}[2][1,2,usedefault]{%

```

```

2347 \endlock\mbox{}}%
2348 \ifboolexpr{%
2349   not test{\ifdefvoid{\at@every@stop@stanza}}}%
2350   and test{\ifstrempy{#1}}}%
2351   and test{\ifstrempy{#2}}}%
2352 }%
2353   {\before@every@stop@stanza\pend [] [\at@every@stop@stanza]}%
2354   {\before@every@stop@stanza\pend [#1] [#2]}%
2355 }%
2356 %

```

`\@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.

```

2357 \newcommand*{\@astanza@line}{%
2358   \stanza@indent%
2359   \endgraf
2360   \stanza@hang%
2361   \ignorespaces}
2362 %
2363 %

```

Lastly reset the modified category codes.

```

2364 \catcode`\&=\next
2365 %
2366 %

```

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

```

\thestanzaR
2367 \newcounter{stanzaL}
2368 \newcounter{stanzaR}
2369 \renewcommand{\thestanzaL}{%
2370   \textbf{\arabic{stanzaL}}%
2371 }
2372 \renewcommand{\thestanzaR}{%
2373   \textbf{\arabic{stanzaR}}%
2374 }
2375 %
2376 %

```

XIII Fixing babel and polyglossia

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

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

`\l@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 `\`.

```
2377 \patchcmd{\bbl@set@language}%
2378   {\select@language{\language}}%
2379   {\edef\language{#1}\select@language{\language}}%
2380   {}%
2381   {}%
2382
2383 %
```

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`

```
2384 \newcommand*{\l@duselanguage}[1]{%
2385   \gdef\theledlanguageL{}
2386   \gdef\theledlanguageR{}
2387
2388 %
```

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

```
2389 \AtBeginDocument{%
2390   \ifpackageloaded{babel}{%
2391     \ifundefined{bbl@main@language}{%
2392 %
```

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

```
2393   }{%
2394 %
```

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.

```
2395   \let\l@doldselectlanguage\selectlanguage
2396   \let\l@doldbbl@set@language\bbl@set@language
2397   \renewcommand{\selectlanguage}[1]{%
2398     \l@doldselectlanguage{#1}%
2399     \ifledRcol \gdef\theledlanguageR{#1}%
2400     \else      \gdef\theledlanguageL{#1}%
2401     \fi}
2402 %
```

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

```

2403 \renewcommand*\l@duselanguage}[1]{%
2404 \let\bbl@savelastskip\relax% Babel 3.61, which provokes spurious
vertical space
2405 \let\bbl@restorelastskip\relax%% Babel 3.61, which provokes spurious
vertical space
2406 \expandafter\l@doldselectlanguage\expandafter{#1}%
2407 }%
2408 %

```

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

```

2409 \gdef\theledlanguageL{\bbl@main@language}%
2410 \gdef\theledlanguageR{\bbl@main@language}%
2411 }%
2412 }{}%
2413 %

```

If use polyglossia

```

2414 \ifpackageloaded{polyglossia}{%
2415 \ifpackageversion{polyglossia}{2025/01/31}{\led@err@polyglossiaTooOld}
%
2416 \renewcommand*\l@duselanguage}[1]{%
2417 \csuse{no\language@name @numbers}%
2418 \selectlanguage{#1}%
2419 }%
2420 \AddToHook{env/Leftside/begin}{%
2421 \AddToHookNext{cmd/polyglossia@language@switched/after}{%
2422 \xdef\theledlanguageL{\language@name}%
2423 }%
2424 }%
2425 \AddToHook{env/Rightside/begin}{%
2426 \AddToHookNext{cmd/polyglossia@language@switched/after}{%
2427 \xdef\theledlanguageR{\language@name}%
2428 }%
2429 }%
2430 \xdef\theledlanguageL{\mainlanguage@name}%
2431 \xdef\theledlanguageR{\mainlanguage@name}%
2432 %

```

That is it.

```

2433 }{}%
2434 }%
2435 %

```

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`

```

2436 \newcount\l@dc@maxchunks
2437 \newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}
2438   \maxchunks{5120}
2439
2440 %

```

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

```

\l@dnumpstartsR
2441 \newcount\l@dnumpstartsR
2442
2443 %

```

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

```

\l@dpscr
2444 \newcount\l@dpscl
2445 \newcount\l@dpscr
2446
2447 %

```

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

```

2448 \newcommand*\l@dsetuprawboxes}{%
2449   \@l@tempcntb=\l@dc@maxchunks
2450   \loop\ifnum\@l@tempcntb>\z@
2451     \newnamebox{\l@dLcolrawbox\the\@l@tempcntb}
2452     \newnamebox{\l@dRcolrawbox\the\@l@tempcntb}
2453     \advance\@l@tempcntb \m@ne
2454     \repeat}
2455
2456 %

```

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

```

2457 \newcommand*\l@dsetupmaxlinecounts}{%
2458   \@l@tempcntb=\l@dc@maxchunks
2459   \loop\ifnum\@l@tempcntb>\z@
2460     \newnamecount{\l@dmaxlinesinpar\the\@l@tempcntb}
2461     \advance\@l@tempcntb \m@ne
2462     \repeat}
2463 \newcommand*\l@dzeromaxlinecounts}{%
2464   \begingroup
2465   \@l@tempcntb=\l@dc@maxchunks
2466   \loop\ifnum\@l@tempcntb>\z@
2467     \global\usenamecount{\l@dmaxlinesinpar\the\@l@tempcntb}=\z@
2468     \advance\@l@tempcntb \m@ne
2469     \repeat
2470   \endgroup}
2471
2472 %

```

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

```

2473 \AtBeginDocument{%
2474   \l@dsetuprawboxes
2475   \l@dsetupmaxlinecounts
2476   \l@dzeromaxlinecounts
2477   \l@dnumpstartsL=\z@
2478   \l@dnumpstartsR=\z@
2479   \l@dpscL=\z@
2480   \l@dpscR=\z@}
2481
2482 %

```

XV Checking text to be processed

```

\if@pstarts \check@pstarts returns \@pstartstrue if there are any unprocessed chunks.
\@pstartstrue
\@pstartstrue \newif\if@pstarts
\@pstartstrue \newcommand*\check@pstarts}{%
\check@pstarts 2483 \@pstartstrue
2484 \ifnum\l@dnumpstartsL>\l@dpscL
2485   \@pstartstrue
2486   \else
2487     \ifnum\l@dnumpstartsR>\l@dpscR
2488       \@pstartstrue
2489     \fi
2490   \fi
2491 }
2492
2493 %
2494
2495 %

```

```

\ifaraw@text \checkraw@text checks whether the current Left or Right box is void or not. If
\araw@texttrue one or other is not void it sets \araw@texttrue, otherwise both are void and it sets
\araw@textfalse \araw@textfalse.
\checkraw@text
2496 \newif\ifaraw@text
2497 \newcommand*\checkraw@text}{%
2498   \araw@textfalse
2499   \ifvbox\namebox{l@dLcolrawbox\the\l@dpscL}
2500     \araw@texttrue
2501   \else
2502     \ifvbox\namebox{l@dRcolrawbox\the\l@dpscR}
2503       \araw@texttrue
2504     \fi
2505   \fi
2506 }
2507
2508 %

```

```

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

```

```

2509 \newcommand*{\@writelinesinparL}{%
2510   \edef\next{%
2511     \write\linenum@out{\string\@pend[\the\@donereallinesL]}}%
2512   \next
2513   \global\@donereallinesL \z@}
2514 \newcommand*{\@writelinesinparR}{%
2515   \edef\next{%
2516     \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}%
2517   \next
2518   \global\@donereallinesR \z@}
2519
2520 %

```

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

\@writepageofparR

```

2521 \newcommand*{\@writepageofparL}[0]{%
2522   \ifnum\@donereallinesL=\z@%
2523     \edef\next{%
2524       \write\linenum@out{\string\@pstart{\the\l@dpscl}{\the\c@page}{\the\
numpagelinesL}}%
2525     }%
2526     \next%
2527     \fi%
2528   }%
2529 \newcommand*{\@writepageofparR}[0]{%
2530   \ifnum\@donereallinesR=\z@%
2531     \edef\next{%
2532       \write\linenum@outR{\string\@pstartR{\the\l@dpscr}{\the\c@page}{\the\
numpagelinesR}}%
2533     }%
2534     \next%
2535     \fi%
2536   }%
2537 %

```

XVI Parallel columns

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

\@eledsectionR

```

2538 \newsavebox{\@eledsectionL}%
2539 \newsavebox{\@eledsectionR}%
2540 %

```

\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.

```

2541 \newcommand*{\Columns}{%
2542   \ifl@dpairing%
2543     \led@err@Columns@InsideEnv%
2544     \fi%
2545     \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{
l@dLcolrawbox1}}}{%
2546       \ifcsboxvoid{l@dRcolrawbox1}{%

```

```

2547     \ifcsboxvoid{l@dLcolrawbox1}%
2548         {\led@err@Columns@WithoutEnv}%
2549         {\led@err@Columns@WithoutRightside}%
2550     }%
2551     {\led@err@Columns@WithoutLeftside}%
2552 }{%
2553 \global\l@dprintingcolumnstrue%
2554 \eledsection@correcting@skip=-\baselineskip% Correction for sections'
titles
2555 \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else
2556     \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
2557 \fi
2558 %

```

Start a group and zero counters, etc.

```

2559 \begingroup
2560     \l@dzeropenalties
2561     \endgraf\global\num@lines=\prevgraf
2562         \global\num@linesR=\prevgraf
2563     \global\par@line=\z@
2564     \global\par@lineR=\z@
2565     \global\l@dpscL=\z@
2566     \global\l@dpscR=\z@
2567     \get@familiarfootnote@number%
2568     \get@intercolumns@width%
2569     \get@sidenote@morespace@columns%
2570 %

```

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

```

2571     \check@pstarts
2572     \loop\if@pstarts
2573         \global\pstartnumtrue
2574         \global\pstartnumRtrue
2575 %

```

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

```

2576     \global\advance\l@dpscL \@ne
2577     \global\advance\l@dpscR \@ne
2578     \global\advance\pstarts@typeset@L\@ne%
2579     \restore@pstartL@pc%
2580     \restore@pstartR@pc%
2581 %

```

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

```

2582     \Columns@print@before@pstart%
2583 %

```

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

```

2584 \checkraw@text
2585 { \loop\ifaraw@text
2586 %

```

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

```

2587 \l@duselanguage{\theledlanguageL}%
2588 \do@lineL
2589 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@@}%
2590 {%
2591 \ifdefstring{\@eledsectmark}{L}%
2592 {\csuse{eled@sectmark@\the\pstarts@typeset@L}%
2593 }{}%
2594 \global\csundef{eled@sectmark@\the\pstarts@typeset@L}%
2595 \savebox{\@eledsectionL}{\parbox[t]{}[t]{\Lcolwidth}{\vbox
{}\print@eledsectionL}}%\vbox{-> prevent alignment troubles with RTL
language
2596 }%
2597 {}%
2598 \l@duselanguage{\theledlanguageR}%
2599 \do@lineR
2600 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}
2601 {%
2602 \ifdefstring{\@eledsectmark}{R}%
2603 {\csuse{eled@sectmark@\the\l@dpscR R}%
2604 }{}%
2605 \global\csundef{eled@sectmark@\the\l@dpscR R}%
2606 \savebox{\@eledsectionR}{\parbox[t]{}[t]{\Rcolwidth}{\vbox
{}\print@eledsectionR}}%\vbox{-> prevent alignment troubles with RTL
language
2607 }%
2608 \hb@xt@ \hsize{%
2609 \ifdefstring{\columns@position}{L}{-}{\hfill }%
2610 \print@leftcolumn%
2611 \print@columnseparator%
2612 \print@rightcolumn%
2613 \ifdefstring{\columns@position}{R}{-}{\hfill}%
2614 }%
2615 \checkraw@text
2616 \checkverseL
2617 \checkverseR
2618 \checkpb@columns
2619 \repeat}
2620 %

```

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.

```

2621 \@writelinesinparL
2622 \@writelinesinparR
2623 \check@pstarts
2624 \ifbypstart@%
2625 \unless@ifinstanza%
2626 \write\linenum@out{\string\@set[1]}%

```

```

2627     \resetprevline@%
2628     \fi%
2629     \fi
2630     \ifbypstart@R
2631         \unless\ifinstanza%
2632             \write\linenum@outR{\string\@set[1]}%
2633             \resetprevline@%
2634         \fi%
2635     \fi
2636     \Columns@print@after@pend%
2637 \repeat
2638 %

```

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.

```

2639     \flush@notes
2640     \flush@notesR
2641 \endgroup
2642 %
2643 \global\l@dpscL=\z@
2644 \global\l@dpscR=\z@
2645 \global\l@dnumstartsL=\z@
2646 \global\l@dnumstartsR=\z@
2647 \global\l@dprintingcolumnsfalse%
2648 \ignorespaces
2649 \global\instanzaLfalse%
2650 \global\instanzaRfalse%
2651 }%
2652 }%
2653 %
2654 %

```

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

```

2655 \def\print@columnseparator{%
2656     \ifdim\beforecolumnseparator<0pt%
2657         \hfill%
2658     \else%
2659         \hspace{\beforecolumnseparator}%
2660     \fi%
2661     \columnseparator%
2662     \ifdim\aftercolumnseparator<0pt%
2663         \hfill%
2664     \else%
2665         \hspace{\aftercolumnseparator}%
2666     \fi%
2667 }%
2668 %

```

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

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

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

```

2669 \newdimen\intercolumns@width%
2670 \newcommand{\get@intercolumns@width}{%
2671 %   \begin{macrocode}
2672   \global\intercolumns@width=\z@%
2673   \ifdefstring{\columns@position}{C}%
2674 %

```

First case, the columns are centered.

```

2675   {%
2676 %

```

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

```

2677   \ifboolexpr{%
2678     test {\ifdimless{\beforecolumnseparator}{\z@}}%
2679     and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2680   }%
2681   {%
2682     \global\intercolumns@width=\dimexpr%
2683       \columnrulewidth +%
2684       (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth)*1/2%
The total of the width before and after column separator
2685     \relax%
2686   }%
2687   }%
2688 %

```

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

```

2689   \ifboolexpr{%
2690     test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2691     and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2692   }%
2693   {%
2694     \global\intercolumns@width=\dimexpr%
2695     \columnrulewidth + \beforecolumnseparator + \aftercolumnseparator
2696 %
2697     \relax%
2698   }%
2699 %

```

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

```

2700 \ifboolexpr{%
2701   test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2702   and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2703 }%
2704 {%
2705   \global\intercolumns@width=\dimexpr%
2706     \columnrulewidth + \beforecolumnseparator +%
2707     (\textwidth - \lcolwidth - \Rcolwidth - \columnrulewidth - \
beforecolumnseparator) * 1/3%
2708   \relax%
2709 }%
2710 {}%
2711 %

```

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

```

2712 \ifboolexpr{%
2713   test {\ifdimless{\beforecolumnseparator}{\z@}}%
2714   and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2715 }%
2716 {%
2717   \global\intercolumns@width=\dimexpr%
2718     \columnrulewidth + \aftercolumnseparator +%
2719     (\textwidth - \lcolwidth - \Rcolwidth - \columnrulewidth - \
aftercolumnseparator) * 1/3%
2720   \relax%
2721 }%
2722 {}%
2723 %

```

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

```

2724 }%
2725 %

```

Other case, the columns are left or right aligned.

```

2726 {%
2727 %

```

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

```

2728 \ifboolexpr{%
2729   test {\ifdimless{\beforecolumnseparator}{\z@}}%
2730   and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2731 }%
2732 {%
2733   \global\intercolumns@width=\dimexpr%
2734     \columnrulewidth +%
2735     (\textwidth - \lcolwidth - \Rcolwidth - \columnrulewidth)*2/3%
The total of the width before and after column separator
2736   \relax%
2737 }%
2738 {}%
2739 %

```

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

```

2740 \ifboolexpr{%
2741   test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2742   and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2743 }%
2744 {%
2745   \global\intercolumns@width=\dimexpr%
2746   \columnrulewidth + \beforecolumnseparator + \aftercolumnseparator
%
2747   \relax%
2748 }%
2749 {}%
2750 %

```

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

```

2751 \ifboolexpr{%
2752   test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2753   and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2754 }%
2755 {%
2756   \global\intercolumns@width=\dimexpr%
2757   \columnrulewidth + \beforecolumnseparator +%
2758   (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
beforecolumnseparator) * 1/2%
2759   \relax%
2760 }%
2761 {}%
2762 %

```

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

```

2763 \ifboolexpr{%
2764   test {\ifdimless{\beforecolumnseparator}{\z@}}%
2765   and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2766 }%
2767 {%
2768   \global\intercolumns@width=\dimexpr%
2769   \columnrulewidth + \aftercolumnseparator +%
2770   (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
aftercolumnseparator) * 1/2%
2771   \relax%
2772 }%
2773 {}%
2774 %

```

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

```

2775 }%
2776 %

```

End of `\get@intercolumns@width`.

```

2777 }%
2778 %

```

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

```

2779 \newcommand{\print@leftcolumn}{%
2780   \if\page@num>\page@numR%
2781     \@l@tempcntb=\page@num%
2782   \else%
2783     \@l@tempcntb=\page@numR%
2784   \fi%
2785   \ifboolexpr{%
2786     not bool {movecolumnpositiononrightpage}%
2787     or test {\ifnumodd{\@l@tempcntb}}%
2788   }%
2789   {%
2790     \unhbox\l@leftbox%
2791     \ifhbox\@eledsectionL%
2792       \usebox{\@eledsectionL}%
2793     \fi%
2794   }%
2795   {%
2796     \unhbox\l@rightbox%
2797     \ifhbox\@eledsectionR%
2798       \usebox{\@eledsectionR}%
2799     \fi%
2800   }%
2801 }%
2802
2803
2804 \newcommand{\print@rightcolumn}{%
2805   \if\page@num>\page@numR%
2806     \@l@tempcntb=\page@num%
2807   \else%
2808     \@l@tempcntb=\page@numR%
2809   \fi%
2810   \ifboolexpr{%
2811     not bool {movecolumnpositiononrightpage}%
2812     or test {\ifnumodd{\@l@tempcntb}}%
2813   }%
2814   {%
2815     \unhbox\l@rightbox%
2816     \ifhbox\@eledsectionR%
2817       \usebox{\@eledsectionR}%
2818     \fi%
2819   }%
2820   {%
2821     \unhbox\l@leftbox%
2822     \ifhbox\@eledsectionL%
2823       \usebox{\@eledsectionL}%
2824     \fi%
2825   }%
2826 }%
2827 %

```

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

```

2828
2829 \newcommand{\checkpb@columns}{%
2830   \newif\if@pb
2831   \newif\if@nopb
2832   \IfStrEq{\led@pb@setting}{before}{
2833     \numdef{next@absline}{\the\absline@num+1}%
2834     \numdef{next@abslineR}{\the\absline@numR+1}%
2835     \xifinlist{next@absline}{\l@prev@pb}{\@pbtrue}{}%
2836     \xifinlist{next@abslineR}{\l@prev@pbR}{\@pbtrue}{}
2837     \xifinlist{next@absline}{\l@prev@nopb}{\@nopbtrue}{}%
2838     \xifinlist{next@abslineR}{\l@prev@nopbR}{\@nopbtrue}{}
2839   }{}
2840   \IfStrEq{\led@pb@setting}{after}{
2841     \xifinlist{the\absline@num}{\l@prev@pb}{\@pbtrue}{}%
2842     \xifinlist{the\absline@numR}{\l@prev@pbR}{\@pbtrue}{}
2843     \xifinlist{the\absline@num}{\l@prev@nopb}{\@nopbtrue}{}%
2844     \xifinlist{the\absline@numR}{\l@prev@nopbR}{\@nopbtrue}{}
2845   }{}
2846   \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
2847   \if@pb\pagebreak[4]\fi
2848 }
2849 %

```

`\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).

```

2850 \newcommand*{\columnseparator}{%
2851   \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}
2852   \newdimen\columnrulewidth
2853   \columnrulewidth=\z@
2854 }
2855 %

```

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

```

\columns@position
2856 \newcommand*{\columnspan}[1]{%
2857   \xdef\columns@position{#1}%
2858 }%
2859 \xdef\columns@position{R}%
2860 %

```

`\beforecolumnseparator` `\beforecolumnseparator` and `\aftercolumnseparator` lengths are defined to -1pt.

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

```

2861 \newlength{\beforecolumnseparator}%
2862 \setlength{\beforecolumnseparator}{-2pt}%
2863
2864 \newlength{\aftercolumnseparator}%

```

```

2865 \setlength{\aftercolumnseparator}{-2pt}%
2866
2867 %

```

`setwidthliketwocolumns` The `\setwidthliketwocolumns` macro is called in `\beginnumbering` in a **non-parallel** typesetting context, to fix the width of the lines to be vertically aligned with parallel columns. It is also called at the beginning of a note's group, if some options are enabled. The `\setposition...` macros are called in `\beginnumbering` in a **non-parallel** typesetting context to fix the position of the lines. The `\setnoteposition...` macros are called in `\xxxfootstart` in a **non-parallel** typesetting context to fix the position of the notes block.

```

2868 \newcommand{\setwidthliketwocolumns}{%
2869   \get@intercolumns@width%As \columns is not necessary called before \
setwidthliketwocolumns
2870   \hsize=\dimexpr \Lcolwidth + \intercolumns@width + \Rcolwidth \relax%
2871 }%
2872
2873 \newcommand{\setpositionliketwocolumns@L}{%
2874   \renewcommand{\ledrfill}{\hfill}%
2875 }%
2876
2877 \newcommand{\setnotespositionliketwocolumns@L}{%
2878 }%
2879
2880 \newcommand{\setpositionliketwocolumns@C}{%
2881   \doinsidelinehook{\hfill}%
2882   \renewcommand{\ledrfill}{\hfill}%
2883 }%
2884
2885 \newcommand{\setnotespositionliketwocolumns@C}{%
2886   \newdimen\temp%
2887   \newdimen\tempa%
2888   \temp=\hsize%
2889   \tempa=\Lcolwidth%
2890   \advance\tempa\Rcolwidth%
2891   \advance\temp-\tempa%
2892   \divide\temp by 2%
2893   \leftskip=\temp%
2894   \rightskip=-\temp%
2895 }%
2896
2897 \newcommand{\setpositionliketwocolumns@R}{%
2898   \doinsidelinehook{\hfill}%
2899 }%
2900 %

```

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

```

2901 \newcommand{\Columns@print@before@pstart}{%
2902   \ifboolexpr{%
2903     test{\ifcsstring{before@pstartL@the\l@dpscl}{\at@every@pstart}}%

```

```

2904     and test {\ifcsstring{before@pstartR@the\l@dpscR}{\at@every@pstart}}%
2905     and test {\ifdefempty{\at@every@pstart}}%
2906     {}%
2907     {%
2908     \ifboolexpr{not togl{before@pstartR@the\l@dpscR @par} and not togl{
before@pstartL@the\l@dpscL @par}}{%
2909         \csuse{before@pstartL@the\l@dpscL}%
2910         \csuse{before@pstartR@the\l@dpscR}%
2911     }{%
2912         \hb@xt@ \hsize{%
2913             \ifdefstring{\columns@position}{L}{\hfill }%
2914             \par\parbox[t] [] [t]{\Lcolwidth}{%
2915                 \csuse{before@pstartL@the\l@dpscL}%
2916             }%
2917             \print@columnseparator%
2918             \parbox[t] [] [t]{\Rcolwidth}{%
2919                 \set@sectcountR%
2920                 \csuse{before@pstartR@the\l@dpscR}%
2921             }%
2922             \ifdefstring{\columns@position}{R}{\hfill}%
2923         }%
2924     }%
2925     }%
2926     \global\csundef{before@pstartL@the\l@dpscL}%
2927     \global\csundef{before@pstartR@the\l@dpscR}%
2928 }%
2929 \newcommand{\Columns@print@after@pend}{%
2930     \ifboolexpr{%
2931         test{\ifcsstring{after@pendL@the\l@dpscL}{\at@every@pend}}%
2932         and test {\ifcsstring{after@pendR@the\l@dpscR}{\at@every@pend}}%
2933         and test {\ifdefempty{\at@every@pend}}%
2934         {}%
2935         {%
2936         \ifboolexpr{not togl{after@pendR@the\l@dpscR @par} and not togl{
after@pendL@the\l@dpscL @par}}{%
2937             \csuse{after@pendL@the\l@dpscL}%
2938             \csuse{after@pendR@the\l@dpscR}%
2939         }{%
2940             \hb@xt@ \hsize{%
2941                 \ifdefstring{\columns@position}{L}{\hfill }%
2942                 \parbox[t] [] [t]{\Lcolwidth}{%
2943                     \csuse{after@pendL@the\l@dpscL}%
2944                 }%
2945                 \print@columnseparator%
2946                 \parbox[t] [] [t]{\Rcolwidth}{%
2947                     \set@sectcountR%
2948                     \csuse{after@pendR@the\l@dpscR}%
2949                 }%
2950                 \ifdefstring{\columns@position}{R}{\hfill}%
2951             }%
2952         }%
2953     }%
2954     \global\csundef{after@pendL@the\l@dpscL}%
2955     \global\csundef{after@pendR@the\l@dpscR}%

```

```
2956 }%
2957 %
```

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`

```
2958 \newcount\numpagelinesL
2959 \newcount\numpagelinesR
2960 \newcount\l@dminpagelines
2961
2962 %
```

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.

```
2963 \newcommandx*{\Pages}[1][1,usedefault]{%
2964   \ifl@pairing%
2965     \led@err@Pages@InsideEnv%
2966   \fi%
2967   \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{l@dLcolrawbox1}}}{%
2968     \ifcsboxvoid{l@dRcolrawbox1}{%
2969       \ifcsboxvoid{l@dLcolrawbox1}{%
2970         {\led@err@Pages@WithoutEnv}%
2971         {\led@err@Pages@WithoutRightside}%
2972       }%
2973       {\led@err@Pages@WithoutLeftside}%
2974     }{%
2975     \ifstrequal{#1}{mainmatter}{\Pages@mainmattertrue}{\Pages@mainmatterfalse}
2976   }%
2977   \eledsection@correcting@skip=-2\baselineskip% line correcting for section
2978   titles.
2979   \parledgroup@notespacing@set@correction%
2980   \typeout{}%
2981   \typeout{***** PAGES *****}%
2982   \ifnum\l@dnumstartsL=\l@dnumstartsR\else%
2983     \led@err@BadLeftRightPstarts{\the\l@dnumstartsL}{\the\l@dnumstartsR}%
2984   \fi%
2985   %
```

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

```
2984 \cleartol@devenpage%
2985 \global\l@dprintingpagestrue%
```



```
2986 \begingroup%
2987 %
```

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

```
2988 \Lcolwidth=\Lcolwidth@pages%
2989 \Rcolwidth=\Rcolwidth@pages%
2990 %
```

```
2991 \l@dzeroopenalties%
2992 \endgraf\global\num@lines=\prevgraf%
2993 \global\num@linesR=\prevgraf%
2994 \global\par@line=\z@%
2995 \global\par@linenR=\z@%
2996 \global\l@dpscL=\z@%
2997 \global\l@dpscR=\z@%
2998 \writtenlinesLfalse%
2999 \writtenlinesRfalse%
3000 \get@familiarfootnote@number%
3001 %
```

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

```
3002 \let\print@Xnotes\print@Xnotes@forpages%
3003 \let\print@notesX\print@notesX@forpages%
3004 %
```

Check if there are chunks to be processed.

```
3005 \check@pstarts%
3006 \loop\if@pstarts%
3007 %
```

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

```
3008 \global\advance\l@dpscL \@ne%
3009 \global\advance\l@dpscR \@ne%
3010 %
```

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

```
3011 \getlinesfromparlistL%
3012 \getlinesfromparlistR%
3013 \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}%
3014 {\usenamecount{l@dmaxlinesinpar\the\l@dpscL}}%
3015 \check@pstarts%
3016 \repeat%
3017 %
```

Zero the counts again, ready for the next bit.

```
3018 \global\l@dpscL=\z@%
3019 \global\l@dpscR=\z@%
3020 %
```

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

```

3021 \getlinesfrompagelistL%
3022 \getlinesfrompagelistR%
3023 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
3024 {\l@dminpagelines}%
3025 %

```

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.

```

3026 \check@pstarts%
3027 \if@pstarts%
3028 %

```

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

```

3029 \global\advance\l@dpscL \@ne%
3030 \global\advance\l@dpscR \@ne%
3031 \global\advance\pstarts@typeset@L\@ne%
3032 \restore@pstartL@pc%
3033 \restore@pstartR@pc%
3034 %

```

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

```

3035 \global\@donereallinesL=\z@%
3036 \global\@donetotallinesL=\z@%
3037 \global\@donereallinesR=\z@%
3038 \global\@donetotallinesR=\z@%
3039 %

```

Start a loop over the boxes (chunks).

```

3040 \checkraw@text%
3041 %
%
3042 \begingroup
3043 { \loop\ifaraw@text%
3044 %

```

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

```

3045 \checkpageL%
3046 \Leftpagehook%
3047 \l@duselanguage{\theledlanguageL}%
3048 { \loop\ifl@dsamepage%
3049 %

```

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

```

3050 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}%
3051 \csuse{before@pstartL@the\l@dpscL}%
3052 \global\csundef{before@pstartL@the\l@dpscL}%
3053 \do@lineL%
3054 \xifinlist{\the\pstarts@typeset@L}{\eled@sections@@}%
3055 {\print@eledsectionL}%
3056 {}%

```

```

3057         \advance\numpagelinesL \@ne%
3058 %

```

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`

```

3059         \ifshiftedpstarts%
3060             \ifdim\ht\l@dleftbox>0pt%
3061                 \parledgroup@correction@notespacing{L}%
3062                 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
3063             \else%
3064                 \xifinlist{the\pstarts@typeset@L}{\eled@sections@
}%
3065                 {\add@apparatusL}%
3066                 {}%
3067                 \unless\ifadvancedshiftedpstarts%
3068                     \dimen0=\pagetotal%
3069                     \advance\dimen0 by \baselineskip%
3070                     \global\pagetotal=\dimen0%
3071                 \else%
3072                     \ifnomaxlines%
3073                         \numdef{\@tmp}{\the\l@dpscL+1}%
3074                         \ifcsdef{minpage@pstart@\@tmp}{%
3075 minpage@pstart@\@tmp}}%
3076                         {\dimen0=\pagetotal%
3077                         \advance\dimen0 by \baselineskip%
3078                         \global\pagetotal=\dimen0%
3079                         }%
3080                         {}%
3081                     }{}%
3082                 \fi%
3083                 \fi%
3084                 \fi%
3085             \else%
3086                 \parledgroup@correction@notespacing{L}%
3087                 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
3088             \fi%
3089 %

```

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

```

3090         \get@nextboxL%
3091         \global\l@dskipversenumberfalse%
3092         \ifprint@last@after@pendL%
3093             \csuse{after@pendL@\the\l@dpscL}%
3094             \global\csundef{after@pendL@\the\l@dpscL}%

```

```

3095         \fi%
3096         \checkpageL%
3097         \checkverseL%
3098         \checkpbL%
3099         \repeat%
3100 %

```

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.

```

3101         \ifl@dpagfull%
3102         \@writelinesonpageL{\the\numpagelinesL}%
3103         \else%
3104         \@writelinesonpageL{1000}%
3105         \fi%
3106 %

```

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.

```

3107         \numpagelinesL \z@%
3108         \parledgroup@correction@notespacing@init%
3109         \clearl@dleftpage }%
3110 %

```

Now do the same for the right text.

```

3111         \checkpageR%
3112         \Rightpagehook%
3113         \l@duselanguage{\theledlanguageR}%
3114 {
3115         \loop\ifl@dsamepage%
3116         \set@sectcountR%
3117         \ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{}%
3118         \csuse{before@pstartR@the\l@dpscR}%
3119         \global\csundef{before@pstartR@the\l@dpscR}%
3120         \do@liner%
3121         \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
3122         {\print@eledsectionR}%
3123         {}%
3124         \advance\numpagelinesR \@one%
3125         \ifshiftedpstarts%
3126         \ifdim\ht\l@drightbox>0pt%
3127         \parledgroup@correction@notespacing{R}%
3128         \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
3129         \else%
3130         \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
3131         {\add@apparatusR}%
3132         {}%
3133         \unless\ifadvancedshiftedpstarts%
3134         \dimen0=\pagetotal%
3135         \advance\dimen0 by \baselineskip%
3136         \global\pagetotal=\dimen0%
3137         \else%

```

```

3137         \ifnomaxlines%
3138         \numdef{\@tmp}{\the\l@dpscR+1}%
3139         \ifcsdef{minpage@pstart@\@tmp}{%
3140         \ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}%
3141         {\dimen0=\pagetotal%
3142         \advance\dimen0 by \baselineskip%
3143         \global\pagetotal=\dimen0%
3144         }%
3145         }%
3146         }%
3147         \fi%
3148         \fi%
3149         \fi%
3150     \else%
3151         \parledgroup@correction@notespacing{R}%
3152         \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
3153         \fi%
3154         \get@nextboxR%
3155         \global\l@dskipversenumberRfalse%
3156         \ifprint@last@after@pendR%
3157         \csuse{after@pendR@\the\l@dpscR}%
3158         \global\csundef{after@pendR@\the\l@dpscR}%
3159         \fi%
3160         \checkpageR%
3161         \checkverseR%
3162         \checkpbR%
3163         \repeat%
3164         \ifl@dpagefull%
3165         \@writelinesonpageR{\the\numpagelinesR}%
3166         \else%
3167         \@writelinesonpageR{1000}%
3168         \fi%
3169         \numpagelinesR=\z@%
3170         \parledgroup@correction@notespacing@init%
3171     %

```

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

```

3172         \clearl@drightpage}%
3173     %

```

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.

```

3174         \checkraw@text%
3175         \ifaraw@text%
3176         \getlinesfrompagelistL%
3177         \getlinesfrompagelistR%
3178         \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
3179         {\l@dminpagelines}%
3180         \fi%
3181         \repeat}%
3182     %

```

We have now output the text from all the chunks.

```
3183 \fi%
3184 %
```

Make sure that there are no inserts hanging around.

```
3185 \flush@notes%
3186 \flush@notesR%
3187 \endgroup%
3188 %
```

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

```
3189 \global\l@dpscL=\z0%
3190 \global\l@dpscR=\z0%
3191 \global\l@dnumstartsL=\z0%
3192 \global\l@dnumstartsR=\z0%
3193 \global\instanzaLfalse%
3194 \global\instanzaRfalse%
3195 \global\inastanzaLfalse%
3196 \global\inastanzaRfalse%
3197 \global\l@dprintingpagesfalse%
3198 %
```

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

```
3199 \ifnum\edtext@later=\edtext@now%
3200 \else%
3201 \led@error@edtext@later@now%
3202 \fi%
3203 \global\edtext@later=\z0%
3204 \global\edtext@now=\z0%
3205 %
```

Prevent final notes from overlapping the line number

```
3206 \finish@Pages@notes%
3207 \ignorespaces}}%
3208
3209
3210 %
```

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.

```
3211 \newcommand{\finish@Pages@notes}{%
3212 \def\do##1{%
3213 %
```

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

```
3214 \ifnocritical@%
3215 \global\newnamebox{##1footins}%
3216 \fi
3217 \ifnofamiliar@%
```

```

3218     \global\newnamebox{footins##1}%
3219     \fi
3220 %

```

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

```

3221     \restore@Xnotes@settings{##1}%
3222     \restore@notesX@settings{##1}%
3223 %

```

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

```

3224     \ifvoid\cuse{##1footins}%
3225     \ifvoid\cuse{footins##1}\else%
3226         \newpage\null%
3227         \listbreak%
3228     \fi%
3229 \else%
3230     \newpage\null%
3231     \listbreak%
3232 \fi%
3233 }%
3234 \dolistloop{\@series}%
3235 }%
3236 %

```

XVII.4 Struts

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

```

\ledstrutR
3237 \newcommand*\ledstrutL{}
3238 \newcommand*\ledstrutR{}
3239
3240 %

```

XVII.5 Page clearing

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

```

3241 \providecommand{\cleartoevenpage}[1][\@empty]{%
3242     \clearpage
3243     \ifodd\c@page\hbox{##1}\clearpage\fi}
3244
3245 \newcommand*\cleartol@devenpage{%
3246     \ifdim\pagetotal<\topskip% on an empty page
3247     \else
3248         \clearpage
3249         \Pages@mainmatter%
3250     \fi
3251     \ifodd\c@page%
3252         \ifprevpnotnumbered%
3253             \addtocounter{par@page}{-1}%
3254     \fi%

```

```

3255 \ifdef{\prevpgstyle}{\thispagestyle{\prevpgstyle}}{}%
3256 \hbox{}\clearpage%
3257 \fi%
3258 }%
3259 %

```

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

```

3260 \newcommand*\clear@dleftpage{%
3261 \ifdim\pagetotal=Opt\hbox{}\fi%
3262 \newpage%
3263 \insert@notes@for@onlyside%
3264 \ifodd\c@page\else
3265 \led@err@LeftOnRightPage
3266 \hbox{}%
3267 \cleardoublepage
3268 \fi}
3269
3270 \newcommand*\clear@drightpage{%
3271 \ifdim\pagetotal=Opt\hbox{}\fi%
3272 \newpage%
3273 \insert@notes@for@onlyside%
3274 \ifodd\c@page
3275 \led@err@RightOnLeftPage
3276 \hbox{}%
3277 \cleartoevenpage
3278 \fi}
3279
3280 %

```

XVII.6 Lines managing

`\getlinesfromparlistL` and `\getlinesfromparlistR` get 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@linesinparL
3281 \newcommand*\getlinesfromparlistL{%
3282 \ifx\linesinpar@listL\empty
3283 \gdef\@cs@linesinparL{0}%
3284 \else
3285 \gl@p\linesinpar@listL\to\@cs@linesinparL
3286 \fi}
3287 \newcommand*\getlinesfromparlistR{%
3288 \ifx\linesinpar@listR\empty
3289 \gdef\@cs@linesinparR{0}%
3290 \else
3291 \gl@p\linesinpar@listR\to\@cs@linesinparR
3292 \fi}

```



```

3293
3294 %
\getlinesfrompagelistL \getlinesfrompagelistL gets the next entry from the \linesonpage@listL and
\@cs@linesonpageL puts it into \@cs@linesonpageL; if the list is empty, it sets \@cs@linesonpageL to
\getlinesfrompagelistR 1000. Similarly for \getlinesfrompagelistR.
\@cs@linesonpageR
3295 \newcommand*\getlinesfrompagelistL{%
3296   \ifx\linesonpage@listL\empty
3297     \gdef\@cs@linesonpageL{1000}%
3298   \else
3299     \gl@p\linesonpage@listL\to\@cs@linesonpageL
3300   \fi}
3301 \newcommand*\getlinesfrompagelistR{%
3302   \ifx\linesonpage@listR\empty
3303     \gdef\@cs@linesonpageR{1000}%
3304   \else
3305     \gl@p\linesonpage@listR\to\@cs@linesonpageR
3306   \fi}
3307
3308 %

```

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

```

3309 \newcommand*\@writelinesonpageL}[1]{%
3310   \edef\next{\write\linenum@out{\string\@lopL{#1}}}%
3311   \next}
3312 \newcommand*\@writelinesonpageR}[1]{%
3313   \edef\next{\write\linenum@outR{\string\@lopR{#1}}}%
3314   \next}
3315
3316 %

```

`\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>`.

```

3317 \newcommand*\l@dcalc@maxoftwo}[3]{%
3318   \ifnum #2>#1\relax
3319     #3=#2\relax
3320   \else
3321     #3=#1\relax
3322   \fi}
3323 \newcommand*\l@dcalc@minoftwo}[3]{%
3324   \ifnum #2<#1\relax
3325     #3=#2\relax
3326   \else
3327     #3=#1\relax
3328   \fi}
3329
3330 %

```

XVII.7 Page break managing

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

```

3331 \checkpageL
3332 \checkpageR
3333 \newif\ifl@dsamepage
3334 \l@dsamepagetrue
3335 \newif\ifl@dpagfull
3336 \newcommand*{\checkpageL}{%
3337   \l@dpagfulltrue
3338   \l@dsamepagetrue
3339   \check@goal
3340   \ifdim\pagetotal<\ledthegoal
3341     \ifnum\numpagelinesL<\l@dsminpagelines
3342       \else
3343         \ifnomaxlines%
3344           \else%
3345             \l@dsamepagefalse%
3346             \l@dpagfullfalse%
3347           \fi%
3348         \fi
3349       \else
3350         \l@dsamepagefalse
3351         \l@dpagfulltrue
3352       \fi%
3353     \ifprint@last@after@pendL%
3354       \l@dpagfullfalse%
3355       \l@dsamepagefalse%
3356       \print@last@after@pendLfalse%
3357     \fi%
3358   }%
3359 \newcommand*{\checkpageR}{%
3360   \l@dpagfulltrue
3361   \l@dsamepagetrue
3362   \check@goal
3363   \ifdim\pagetotal<\ledthegoal
3364     \ifnum\numpagelinesR<\l@dsminpagelines
3365       \else
3366         \ifnomaxlines%
3367           \else%
3368             \l@dsamepagefalse%
3369             \l@dpagfullfalse%
3370           \fi%
3371         \fi
3372       \else
3373         \l@dsamepagefalse
3374         \l@dpagfulltrue
3375       \fi%

```

```

3376 \ifprint@last@after@pendR%
3377 \l@dpagfullfalse%
3378 \l@dsamepagefalse%
3379 \print@last@after@pendRfalse%
3380 \fi%
3381 }%
3382
3383 %

```

\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.

```

3384 \newcommand{\checkpbL}{
3385   \IfStrEq{\led@pb@setting}{after}{
3386     \xifinlist{the\absline@num}{\l@prev@pb}{\l@dpagfulltrue\
l@dsamepagefalse}{
3387     \xifinlist{the\absline@num}{\l@prev@nopb}{\l@dpagfullfalse\
l@dsamepagetrue}{
3388     }{
3389     \IfStrEq{\led@pb@setting}{before}{
3390       \numdef{next@absline}{the\absline@num+1}
3391       \xifinlist{next@absline}{\l@prev@pb}{\l@dpagfulltrue\
l@dsamepagefalse}{
3392       \xifinlist{next@absline}{\l@prev@nopb}{\l@dpagfullfalse\
l@dsamepagetrue}{
3393       }{
3394     }
3395
3396 \newcommand{\checkpbR}{
3397   \IfStrEq{\led@pb@setting}{after}{
3398     \xifinlist{the\absline@numR}{\l@prev@pbR}{\l@dpagfulltrue\
l@dsamepagefalse}{
3399     \xifinlist{the\absline@numR}{\l@prev@nopbR}{\l@dpagfullfalse\
l@dsamepagetrue}{
3400     }{
3401     \IfStrEq{\led@pb@setting}{before}{
3402       \numdef{next@abslineR}{the\absline@numR+1}
3403       \xifinlist{next@abslineR}{\l@prev@pbR}{\l@dpagfulltrue\
l@dsamepagefalse}{
3404       \xifinlist{next@abslineR}{\l@prev@nopbR}{\l@dpagfullfalse\
l@dsamepagetrue}{
3405       }{
3406     }
3407   %

```

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

```

3408 \newcommand{\checkverseL}{%
3409 \ifinstanzL%
3410   \iflednopbinverse%
3411   \ifinserthangingsymbol%
3412     \numdef{\prev@abslineverse}{the\absline@num-1}%
3413     \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{

```

```

3414 \IfStrEq{\led@pb@setting}{before}{%
3415 \ifnum\numpagelinesL<3%
3416 \ledpbnum{\prev@abslineverse}%
3417 \fi%
3418 }{}%
3419 \fi%
3420 \fi%
3421 \fi%
3422 }
3423 \newcommand{\checkverseR}{%
3424 \ifinstanzaR%
3425 \iflednopbinverse%
3426 \ifinserthangingsymbolR%
3427 \numgdef{\prev@abslineverse}{\the\absline@numR-1}%
3428 \IfStrEq{\led@pb@setting}{after}{\lednopbnumR{\prev@abslineverse}}{}%
3429 \IfStrEq{\led@pb@setting}{before}{%
3430 \ifnum\numpagelinesR<3%
3431 \ledpbnumR{\prev@abslineverse}%
3432 \fi%
3433 }{}%
3434 \fi%
3435 \fi%
3436 \fi%
3437 }
3438 %

```

`\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`.

```

\goalfraction
\check@goal
3439 \newdimen\ledthegoal
3440 \ifshiftedpstarts
3441 \newcommand*{\@goalfraction}{0.95}
3442 \else
3443 \newcommand*{\@goalfraction}{0.9}
3444 \fi
3445
3446 \newcommand*{\check@goal}{%
3447 \ledthegoal=\@goalfraction\pagegoal}
3448 \newcommand{\setgoalfraction}[1]{%
3449 \xdef\@goalfraction{#1}%
3450 }
3451 %

```

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

```

\ifwrittenlinesL
3452 \newif\ifwrittenlinesL
3453 \newif\ifwrittenlinesR
3454
3455 %

```

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.

```
3456 \newif\if@getnextbox%
3457 %
```

`\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.

```
3458 \newcommand*{\get@nextboxL}{%
3459   \ifvbox\namebox{1@dLcolrawbox\the\1@dpscL}% box is not empty
3460 %
```

The current box is not empty; do nothing.

```
3461   \else%                               box is empty
3462 %
```

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

```
3463   \@getnextboxtrue%Should be local, but be cautious
3464 %
```

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

```
3465   \ifnum\usernamecount{1@dmaxlinesinpar\the\1@dpscL}>\@donetotallinesL
3466     \parledgroup@notes@endL%
3467     \unless\ifnosyncpstarts%
3468       \@getnextboxfalse%
3469 %
```

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.

```
3470   \ifnomaxlines%
3471     \ifdim\pagetotal<\ledthegoal%
3472       \numdef{\@tmp}{\1@dpscL+1}%
3473       \ifcsdef{afterlines@pstart@\@tmp R}{%
3474         \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp R}}
3475           {}%
3476         {\ifcsdef{minpage@pstart@\@tmp}%
3477           {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3478             {\ifnum\numpagelinesL=\1@dminpagelines%
3479               \@getnextboxtrue%
3480             \fi%
3481           }%
3482         {\@getnextboxtrue}}%

```

```

3483         {\@getnextboxtrue}%
3484     }%
3485 }%
3486 {}%
3487 \fi%
3488 \fi%
3489 \fi%
3490 \else%
3491 \ifnomaxlines%
3492 \numdef{\@tmp}{\the\l@dpscl+1}%
3493 \ifcsdef{minpage@pstart@\@tmp}{%
3494 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3495 {\ifdingreater{\pagetotal}{\ledthegoal}%
3496 {\@getnextboxtrue}%
3497 {\@getnextboxfalse}%
3498 }%
3499 {\@getnextboxtrue}%
3500 }{}%
3501 \fi%
3502 \fi%
3503 %

```

Sufficient lines have been output.

```

3504 \if@getnextbox%
3505 \ifnum\usenamecount{l@dmxlinesinpar\the\l@dpscl}=\@donetotallinesL
3506 \parledgroup@notes@endL
3507 \fi
3508 \ifwrittenlinesL\else
3509 %

```

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

```

3510 \@writelinesinparL
3511 \writtenlinesLtrue
3512 \fi
3513 \ifnum\l@dnumpstartsL>\l@dpscl
3514 %

```

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.

```

3515 \writtenlinesLfalse
3516 \ifbypstart@
3517 \global\line@num=0%
3518 \resetprevline@%
3519 \fi
3520 % Add the content of the optional argument of the previous \protect\cs{pend
3521 }.
3521 % \begin{macrocode}
3522 \csuse{after@pendL@\the\l@dpscl}%
3523 \global\csundef{after@pendL@\the\l@dpscl}%
3524 %

```

Check the number of lines

```

3525 \l@dcalc@maxoftwo{\the\usernamecount{1@dmaxlinesinpar\the\l@dpscL}}%
3526         {\the\@donetotallinesL}%
3527         {\usernamecount{1@dmaxlinesinpar\the\l@dpscL}}%
3528 \global\@donetotallinesL \z@
3529 %

```

Go to the next pstart

```

3530 \global\advance\l@dpscL \@ne
3531 \global\advance\pstarts@typeset@L\@ne%
3532 \global\pstartnumtrue%
3533 \restore@pstartL@pc%
3534 %

```

Add notes of parallel ledgroup.

```

3535 \parledgroup@notes@endL
3536 \parledgroup@correction@notespacing@final{L}
3537 \else
3538 %

```

```

3539 \print@last@after@pendLtrue%
3540 \fi
3541 \fi
3542 \fi}
3543 %

```

```

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

```

```

3572 \else%
3573 \ifnomaxlines%
3574 \numdef{\@tmp}{\the\l@dpscR+1}%
3575 \ifcsdef{minpage@pstart@\@tmp}{%
3576 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3577 {\ifdimgreater{\pagetotal}{\ledthegoal}%
3578 {\@getnextboxtrue}%
3579 {\@getnextboxfalse}%
3580 }%
3581 {\@getnextboxtrue}%
3582 }{}
3583 \fi%
3584 \fi%
3585 \if@getnextbox%
3586 \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscR}=\@donetotallinesR
3587 \parledgroup@notes@endR
3588 \fi
3589 \ifwrittenlinesR\else
3590 \@writelinesinparR
3591 \writtenlinesRtrue
3592 \fi
3593 \ifnum\l@dnumpstartsR>\l@dpscR
3594 \writtenlinesRfalse
3595 \ifbypstart@R
3596 \unless\ifinstanzaR%
3597 \global\line@numR=0%
3598 \resetprevline@%
3599 \fi%
3600 \fi
3601 \csuse{after@pendR@\the\l@dpscR}%
3602 \global\csundef{after@pendR@\the\l@dpscR}%
3603 \l@dcalc@maxoftwo{\the\usernamecount{1@dmaxlinesinpar\the\l@dpscR}}%
3604 {\the\@donetotallinesR}%
3605 {\usernamecount{1@dmaxlinesinpar\the\l@dpscR}}%
3606 \global\@donetotallinesR \z@
3607 \global\advance\l@dpscR \@ne
3608 \global\pstartnumRtrue%
3609 \restore@pstartR@pc%
3610 \parledgroup@notes@endR
3611 \parledgroup@correction@notespadding@final{R}
3612 \else
3613 \print@last@after@pendRtrue%
3614 \fi
3615 \fi
3616 \fi}
3617
3618 %

```

XVIII The mainmatter option of \Pages

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

`\Pages@mainmatter`, called by `\cleartol@devenpage`.

```

\ifPages@mainmatter 3621 \newif\ifPages@mainmatter
\Pages@mainmatter 3622 \newcommand{\Pages@mainmatter}{%
3623   \ifPages@mainmatter%
3624   \pagenumbering{arabic}%
3625   \addtocounter{page}{1}%
3626   \addtocounter{par@page}{-1}%
3627   \patchcmd{\thepage}{page}{par@page}{-}{-}%
3628   \fi%
}
```

XIX Sections' titles' commands

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

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

```

\chapter 3630 \pretocmd{\chapter}{%
3631   \ifl@dprintingpages%
3632   \vbox{}%
3633   \fi%
3634   }%
3635   {}%
3636   {}%
```

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

```

3637 \newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}}
3638 \eledsectnotoc{R}
3639 %
```

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

```

3640 \newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}}
3641 \eledsectmark{L}
3642 %
```

`\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`.

```

3643 \newskip\eledsection@correcting@skip
3644 %
```

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

```

3645 \newwrite\eled@sectioningR@out
3646 %
```

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

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

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

```
3647 \def\l@prev@pbR{}
3648 \def\l@prev@nopbR{}
3649 %
```

`\ledpbR` The `\ledpbR` macro writes the call to `\led@pbR` in line-list file. The `\ledpbnmR` macro writes the call to `\led@pbnmR` 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.

```
3650 \newcommand{\ledpbR}{\write\linenum@outR{\string\led@pbR}}
3651 \newcommand{\ledpbnmR}[1]{\write\linenum@outR{\string\led@pbnmR{#1}}}
3652 \newcommand{\lednopbR}{\write\linenum@outR{\string\led@nopbR}}
3653 \newcommand{\lednopbnumR}[1]{\write\linenum@outR{\string\led@nopbnumR{#1}}}
3654 %
```

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

```
3655 \newcommand{\led@pbR}{\listxadd{\l@prev@pbR}{\the\absline@numR}}
3656 \newcommand{\led@pbnmR}[1]{\listxadd{\l@prev@pbR}{#1}}
3657 \newcommand{\led@nopbR}{\listxadd{\l@prev@nopbR}{\the\absline@numR}}
3658 \newcommand{\led@nopbnumR}[1]{\listxadd{\l@prev@nopbR}{#1}}
3659 %
```

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).

```
3660 \newmarks\parledgroup@
3661 \newmarks\parledgroup@series
3662 \newmarks\parledgroup@type
3663 %
```

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

```

3664 \newcommand{\parledgroup@notes@startL}{%
3665   \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscl}>0%
3666     \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
bhooknoteX@\splitfirstmarks\parledgroup@series}}{}}%
3667     \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
bhookXnote@\splitfirstmarks\parledgroup@series}}{}}%
3668     \fi%
3669     \global\ledgroupnotesL@true%
3670     \insert@noterule@ledgroup{L}%
3671 }
3672 \newcommand{\parledgroup@notes@startR}{%
3673   \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscR}>0%
3674     \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
bhooknoteX@\splitfirstmarks\parledgroup@series}}{}}%
3675     \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
bhookXnote@\splitfirstmarks\parledgroup@series}}{}}%
3676     \fi%
3677     \global\ledgroupnotesR@true%
3678     \insert@noterule@ledgroup{R}%
3679 }
3680 %

```

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

```

3681 \newcommand{\parledgroup@notes@endL}{%
3682   \global\ledgroupnotesL@false%
3683 }
3684 \newcommand{\parledgroup@notes@endR}{%
3685   \global\ledgroupnotesR@false%
3686 }
3687 %

```

`\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`

```

3688 \newcommand{\insert@noterule@ledgroup}[1]{
3689   \IfStrEq{\splitbotmarks\parledgroup@}{begin}{%
3690     \IfStrEq{\splitbotmarks\parledgroup@type}{Xfootnote}{
3691       \csuse{ifledgroupnotes#1@}
3692       \vskip\skip\csuse{mp\splitbotmarks\parledgroup@series footins}
3693       \csuse{\splitbotmarks\parledgroup@series footnoterule}
3694       \fi
3695     }
3696   {}
3697   \IfStrEq{\splitbotmarks\parledgroup@type}{footnoteX}{
3698     \csuse{ifledgroupnotes#1@}
3699     \vskip\skip\csuse{mpfootins\splitbotmarks\parledgroup@series}
3700     \csuse{footnoterule\splitbotmarks\parledgroup@series}
3701     \fi
3702   }{}}
3703 }
3704 {}
3705 }
3706 %

```

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

```
3707 \newcommand{\setparledgroupnotespacing}[1]{\gdef\@parledgroupnotespacing
3708 {#1}}
3709 \newcommand{\@parledgroupnotespacing}{}
3709 %
```

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

```
3710 \dimdef{\parledgroup@notespacing@correction}{0pt}
3711 \newcommand{\parledgroup@notespacing@set@correction}{%
3712   {\@getfirstseries\csuse{Xnotefontsize@\@firstseries}}%We suppose all the
3713   series has the same footnote size setup
3714   \@parledgroupnotespacing\dimdef{\temp@spacing}{\baselineskip}}%
3715   \dimdef{\parledgroup@notespacing@correction}{\baselineskip-\temp@spacing
3716   }%
3715 }
3716 %
```

`\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.

```
3717 \newcommand{\parledgroup@correction@notespacing@init}{
3718   \dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}
3719   \dimdef{\parledgroup@notespacing@correction@modulo}{0pt}
3720 }
3721 \parledgroup@correction@notespacing@init
3722 %
```

`\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`.

```
3723 \newcommand{\parledgroup@correction@notespacing@final}[1]{
3724   \ifparledgroup
3725   \vspace{\parledgroup@notespacing@correction@accumulated}
3726   \parledgroup@correction@notespacing@init%
3727   \ifstrequal{#1}{L}{
3728     \numdef{\@checking}{\the\l@dpscL-1}
3729   }{
3730     \numdef{\@checking}{\the\l@dpscR-1}
3731   }
3732   \dimdef{\@beforenotes@current@diff}{\csuse{\parledgroup@beforenotes@
3733   \@checking L}-\csuse{\parledgroup@beforenotes@\@checking R}}%
3734   \ifstrequal{#1}{L}{
3735     {% Left
3736     \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{-\
3736   \@beforenotes@current@diff}}%
3736   }%
3736 }
```

```

3737     {% Right
3738     \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{\
@beforenotes@current@diff}}{}}
3739     }%
3740     \fi
3741   }
3742   %

```

`\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.

```

3743 \newcommand{\parledgroup@correction@notespacing}[1]{%
3744   \csuse{ifledgroupnotes#1@}%
3745   \vspace{-\parledgroup@notespacing@correction}%
3746   \dimdef{\parledgroup@notespacing@correction@accumulated}{\
parledgroup@notespacing@correction@accumulated+\
parledgroup@notespacing@correction}%
3747   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo+\
parledgroup@notespacing@correction}%
3748   \ifdimless{\parledgroup@notespacing@correction@modulo}{\baselineskip
}]{\advance\numpagelinesL -\@ne%
3749   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo-\baselineskip}%
3750   }% mean greater than equal
3751   \fi%
3752 }
3753 %

```

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

```

3754 \dimdef\parledgroup@beforenotesL{0pt}
3755 \dimdef\parledgroup@beforenotesR{0pt}
3756 %

```

The macro `\parledgroup@beforenotes@save` dumps the space before notes of the current parallel ledgroup in a macro named with the current `pstart` number.

```

3757 \newcommand{\parledgroup@beforenotes@save}[1]{
3758   \ifparledgroup
3759     \csdimgdef{@parledgroup@beforenotes@\the\csuse{1@dnumpstarts#1}#1}{\
csuse{\parledgroup@beforenotes#1}}

```

```
3760 \csdingdef{parledgroup@beforenotes#1}{0pt}  
3761 \fi  
3762 }  
3763 %
```

XXII Compatibility with eledmac

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

```
3764 \ifeledmaccompat@%  
3765  
3766  
3767 \unless\ifnocritical@  
3768 \let\onlyXside\Xonlyside  
3769 \fi  
3770 \fi  
3771 %
```

XXIII The End

</code>

Appendix A Some things to do when changing version

A.1 Migration to eledpar 1.4.3

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

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

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

See the following two examples:

With standard `\hangingsymbol`:

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

With the modification of the `hangingsymbol`:

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

A.2 Migration from eledpar to reledpar

As for migration from eledmac to reledmac:

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

A.2.1 Deprecated options

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

A.2.2 `\renewcommand` replaced with command

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

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

A.2.3 Commands the names of which have changed

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

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

A.3 Migration to `reledpar 2.2.0`

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

A.4 Migration to `reledpar 2.3.0`

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

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

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

A.5 Migration to `reledpar 2.4.0`

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

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

A.6 Migration to `reledpar 2.5.0`

If you use either `\stanza` or `astanza` environment, please read [A.12 p. 388](#).

A.7 Migration to `reledpar 2.6.0`

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

A.8 Migration to `reledpar 2.6.1`

If you use `perpage` package to control footnote numbering, please read the handbook on [6.3.3 p. 14](#).

A.9 Migration to `reledpar 2.14.0`

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

A.10 Migration to reledpar 2.14.1

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

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

Please contact us if that causes problems.

A.11 Migration to reledpar v. 2.17.2

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

A.12 Migration to reledpar v. 2.24.3

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

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

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

References

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

Index

Symbols

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

<code>\@donereallinesR</code>	1
<code>\@donetotallinesL</code>	1
<code>\@donetotallinesR</code>	1
<code>\@eledsectionL</code>	1
<code>\@eledsectionR</code>	1
<code>\@lab</code>	1
<code>\@lopL</code>	1
<code>\@lopR</code>	1
<code>\@morespace@leftnote@rightcolumn</code>	1
<code>\@morespace@rightnote@leftcolumn</code>	1
<code>\@nl</code>	1
<code>\@nl@regR</code>	1
<code>\@par@sync@option</code>	1
<code>\@par@this@sync@option</code>	1
<code>\@parledgroupnotespacing</code>	1
<code>\@pend</code>	1
<code>\@pendR</code>	1
<code>\@pstart</code>	1
<code>\@pstartR</code>	1
<code>\@pstartsfalse</code>	1
<code>\@pstartstrue</code>	1
<code>\@ref</code>	1
<code>\@ref@reg@parseR</code>	1
<code>\@ref@regR</code>	1
<code>\@set</code>	1
<code>\@stopastanza</code>	1
<code>\@writelinesinparL</code>	1
<code>\@writelinesinparR</code>	1
<code>\@writelinesonpageL</code>	1
<code>\@writelinesonpageR</code>	1
<code>\@writepageofparL</code>	1
<code>\@writepageofparR</code>	1
COMMAND\+	52
COMMAND\@Rlineflag	83, 153
COMMAND\@adv	43, 151
COMMAND\@cs@linesinparL	112
COMMAND\@cs@linesonpageL	113
COMMAND\@eledsectionL	93
COMMAND\@eledsectionR	93
COMMAND\@eledsectmark	121
COMMAND\@eledsectnotoc	121
COMMAND\@goalfraction	13, 116
COMMAND\@l@dtempcnta	77
COMMAND\@lab	41, 84, 151
COMMAND\@lopL	47, 107, 113
COMMAND\@lopR	47, 113
COMMAND\@morespace@leftnote@rightcolumn	84, 85
COMMAND\@morespace@rightnote@leftcolumn	84, 85
COMMAND\@namedef	29
COMMAND\@namuse	29
COMMAND\@nl	42, 48, 49, 84, 151
COMMAND\@nl@regR	42
COMMAND\@page	84
COMMAND\@par@sync@option	41

COMMAND\@parledgroupnotespacing	124
COMMAND\@pend	46
COMMAND\@pendR	46
COMMAND\@pstart	47
COMMAND\@pstartstrue	92
COMMAND\@ref	44–46, 49, 152
COMMAND\@ref@later	46
COMMAND\@ref@reg@parseargR	46
COMMAND\@ref@regR	45
COMMAND\@set	43, 152
COMMAND\@sw	45
COMMAND\AtBeginPairs	8, 55, 150, 154
COMMAND\AtEveryPend	10, 150, 151
COMMAND\AtEveryPend*	10
COMMAND\AtEveryPstart	2, 10, 19, 63, 94, 150, 151
COMMAND\AtEveryPstart*	10
COMMAND\AtEveryPstartCall	2, 19, 63, 151
COMMAND\AtEveryStanza	153
COMMAND\AtEveryStopStanza	153
COMMAND\Clear the right lines for \read@linelist	41
COMMAND\Columns	7, 8, 19, 53, 93, 101, 147–149, 151–153
COMMAND\Columns@print@after@pend	102
COMMAND\Columns@print@before@pstart	102
COMMAND\Lcolwidth	8, 11, 105, 129
COMMAND\Lcolwidth	154
COMMAND\Leftpagehook	20, 156
COMMAND\Leftsidehook	147
COMMAND\Leftsidehookend	147
COMMAND\MakePerPage	14
COMMAND\Pages	4, 7, 11, 13, 19, 26, 52, 53, 74, 78, 80, 104, 105, 110, 120, 124, 147, 149–153, 155
COMMAND\Pages@mainmatter	121
COMMAND\Pairs	52
COMMAND\Rcolwidth	8, 11, 105, 129
COMMAND\Rcolwidth	154
COMMAND\Rightpagehook	20, 156
COMMAND\Rightsidehook	147
COMMAND\Rightsidehookend	147
COMMAND\Rlineflag	127
COMMAND\Xendlineflag	16, 153
COMMAND\Xlineflag	15, 128, 153
COMMAND\Xmaxhnotes	14
COMMAND\Xnoteswidthliketwocolumns	9, 150
COMMAND\Xonlyside	14, 78, 128, 154, 156
COMMAND\Xparindent	156
COMMAND\Xpstart	155
COMMAND\Xtxtbeforenotes	153
COMMAND\&	20
COMMAND\absline@numR	40
COMMAND\add@apparatusL	66
COMMAND\add@apparatusR	68
COMMAND\add@penalties	77
COMMAND\add@penaltiesL	77
COMMAND\advanceline	43, 49, 152
COMMAND\affixline@num	72

COMMAND\affixline@numR	72, 147, 148
COMMAND\affixpstart@num	74
COMMAND\affixpstart@numR	74
COMMAND\affixside@note	85
COMMAND\aftercolumnseparator	9, 101, 149, 154, 155
COMMAND\araw@textfalse	92
COMMAND\araw@texttrue	92
COMMAND\at@begin@pairs	55
COMMAND\autopar	19, 155
COMMAND\ballast@count	77
COMMAND\baselineskip	101, 125
COMMAND\bbl@set@language	89, 152
COMMAND\beforecolumnseparator	9, 101, 149, 155
COMMAND\begin	20
COMMAND\beginnumbering	9, 10, 16, 17, 19, 30, 41, 102, 148, 149, 151, 154, 155
COMMAND\beginnumberingR	48
COMMAND\bf	148
COMMAND\bfseries	148
COMMAND\brokenpenalty	77
COMMAND\by@autopar	59
COMMAND\chapter	121, 147
COMMAND\check@goal	116
COMMAND\check@pstarts	92
COMMAND\checkpageL	107, 114
COMMAND\checkpb@columns	101
COMMAND\checkpbL	115
COMMAND\checkpbR	115
COMMAND\checkdraw@text	92
COMMAND\checkverseL	115
COMMAND\checkverseR	115
COMMAND\clear(double)page	111
COMMAND\clearl@dleftpage	112
COMMAND\clearl@drightpage	112
COMMAND\clearpage	112, 150
COMMAND\cleartoevenpage	111
COMMAND\cleartol@devenpage	111, 121
COMMAND\columnrulewidth	8, 101
COMMAND\columns	154
COMMAND\columns@position	101
COMMAND\columnseparator	8
COMMAND\columnspan	9, 149
COMMAND\continuousnumberingwithcolumns	155
COMMAND\correct@Xfootins@box	151
COMMAND\correct@footinsX@box	151
COMMAND\critext	150
COMMAND\csname	53
COMMAND\displaywidowpenalty	77
COMMAND\do@actions	70
COMMAND\do@actions@fixedcode	147
COMMAND\do@actions@nextR	70
COMMAND\do@actionsR	70, 147
COMMAND\do@ballast	77
COMMAND\do@ballastR	70
COMMAND\do@insidelineLhook	148

COMMAND\do@insidelineRhook	148
COMMAND\do@line	63
COMMAND\do@line(L/R)	67
COMMAND\do@lineL	64, 77, 147, 148
COMMAND\do@lineLhook	147
COMMAND\do@lineR	67, 147–149
COMMAND\do@lineRhook	147
COMMAND\do@lockoff	152
COMMAND\do@lockoffR	44
COMMAND\do@lockon	152
COMMAND\do@lockonR	43
COMMAND\doinsidelineLhook	20, 129, 150, 154
COMMAND\doinsidelineRhook	20, 129, 150, 154
COMMAND\doinsidelinehook	20
COMMAND\doinsidethislinehook	156, 157
COMMAND\dolineLhook	20, 150
COMMAND\dolineRhook	20, 150
COMMAND\dolinehook	20
COMMAND\edindex	150
COMMAND\edlabel	148, 150
COMMAND\edlineref	154
COMMAND\edtext	15, 44, 49, 81, 150, 154, 155
COMMAND\edtext@later	81, 83, 110
COMMAND\edtext@now	110
COMMAND\edtextlater	15, 81, 82, 154, 155
COMMAND\edtextnow	15, 81–83, 154
COMMAND\eled@sectioningR@out	121
COMMAND\eledchapter	151
COMMAND\eledsection	150–152, 156
COMMAND\eledsection@correcting@skip	121
COMMAND\eledsectmark	22, 121
COMMAND\eledsectnotoc	22, 121
COMMAND\eledxxx	149
COMMAND\end	20
COMMAND\endgraf	61
COMMAND\endlock	49, 152
COMMAND\endnumbering	9, 10, 16, 19, 32, 151, 154, 155
COMMAND\endnumberingR	48
COMMAND\endsub	49, 152
COMMAND\endumbering	16
COMMAND\expandafter	51
COMMAND\extensionchars	30
COMMAND\firstlinenum	17, 149, 152, 153
COMMAND\firstsublinenum	149, 152, 153
COMMAND\fix@page	43, 152
COMMAND\flag@end	49, 147, 149
COMMAND\flag@start	49, 149
COMMAND\flush@notesR	77
COMMAND\footnote	53
COMMAND\footnoteX	50, 51
COMMAND\footnoteX@reading	153
COMMAND\footnoteXmk	15
COMMAND\footnoteXnomk	15, 51, 157
COMMAND\frontmatter	13, 22

COMMAND\get@intercolumns@width	96, 99
COMMAND\get@nextboxL	148
COMMAND\get@nextboxR	148
COMMAND\get@sidenote@morespace@columns	84, 85
COMMAND\getline@numL	69
COMMAND\getline@numR	69
COMMAND\getlinesfrompagelistL	113
COMMAND\getlinesfrompagelistR	113
COMMAND\getlinesfromparlistL	112
COMMAND\getlinesfromparlistR	112
COMMAND\gl@p	51
COMMAND\goalfraction	127
COMMAND\hangingsymbol	127, 147
COMMAND\hfill	96, 101
COMMAND\hidenumbering	17, 151
COMMAND\if@getnextbox	117
COMMAND\if@sidenotesmarginpage	84, 85
COMMAND\ifPages@mainmatter	120
COMMAND\ifbypage@	151
COMMAND\ifbypstart@R	151
COMMAND\ifcsboxvoid	30
COMMAND\ifdim	96
COMMAND\ifinserthangingsymbol	86
COMMAND\ifinserthangingsymbolR	86
COMMAND\ifl@dpagefull	114
COMMAND\ifl@dpageing	26, 150
COMMAND\ifl@dpairing	26, 147
COMMAND\ifl@dsamelang	149
COMMAND\ifl@dsamepage	114
COMMAND\ifl@pagefull	107
COMMAND\ifledRcol	26
COMMAND\iflinenumberLevenifblank	39
COMMAND\iflinenumberRevenifblank	39
COMMAND\iflledRcol	147
COMMAND\ifmovecolumnsonpositiononrightpage	25
COMMAND\ifnumberedpar@	57
COMMAND\ifnumberingR	147
COMMAND\ifnumberline	53
COMMAND\ifnumberpstart	53
COMMAND\ifpst@rtedL	30, 32, 58, 147
COMMAND\ifpst@rtedR	30
COMMAND\ifsublines@	43
COMMAND\ifsublines@R	156
COMMAND\insert@countR	44
COMMAND\insert@noterule@ledgroup	123
COMMAND\insert@notes@for@onlyside	80
COMMAND\insertlines@list	44
COMMAND\insertlines@listR	45
COMMAND\inserts@list	57
COMMAND\inserts@listR	76
COMMAND\intercolumns@width	96
COMMAND\l@d@nums	49
COMMAND\l@d@set	43, 49, 152
COMMAND\l@dLcolrawbox	57

COMMAND\l@dLcolrawbox1	91
COMMAND\l@dLcolrawbox2	91
COMMAND\l@dRcolrawbox	57
COMMAND\l@dcalc@maxoftwo	113
COMMAND\l@dcalc@minoftwo	113
COMMAND\l@dchecklang	147, 149
COMMAND\l@dcsnote	149
COMMAND\l@dleftbox	63, 107, 150
COMMAND\l@dlinenumR	39, 147
COMMAND\l@dlsnote	149
COMMAND\l@dmake@labels	83
COMMAND\l@dmaxlinesinpar	105
COMMAND\l@dmaxlinesinpar1	91
COMMAND\l@dminpagelines	105, 147
COMMAND\l@dnumstartsL	91, 147
COMMAND\l@dprintingcolumnstrue	150
COMMAND\l@dprintingpagestrue	150
COMMAND\l@dpscL	94, 105, 106, 118
COMMAND\l@dpscR	94, 105, 106
COMMAND\l@drsnote	149
COMMAND\l@dsetupmaxlinecounts	91
COMMAND\l@duselanguage	89, 147
COMMAND\l@dzeromaxlinecounts	91
COMMAND\l@prev@nopbR	122
COMMAND\l@prev@pbR	122
COMMAND\labelpstarttrue	148
COMMAND\labelref@list	84
COMMAND\labelref@listR	83
COMMAND\lang	89
COMMAND\last@page@numR	43
COMMAND\led	148
COMMAND\led@nopbR	122
COMMAND\led@nopbnumR	122
COMMAND\led@pbR	122
COMMAND\led@pbnumR	122
COMMAND\ledinnernote	21
COMMAND\ledleftnote	21
COMMAND\lednopb	21, 101, 115
COMMAND\lednopbR	122
COMMAND\lednopbnumR	122
COMMAND\ledouternote	21
COMMAND\ledpb	101, 115
COMMAND\ledpbR	122
COMMAND\ledpbnumR	122
COMMAND\ledrightnote	21
COMMAND\ledsidenote	21
COMMAND\ledstrutL	147
COMMAND\ledstrutR	147, 152
COMMAND\ledthegoal	116
COMMAND\ledtrutL	147, 152
COMMAND\leftlinenumR	39, 147
COMMAND\leftlinenumannotationR	40
COMMAND\lemma	82
COMMAND\let	51

COMMAND\line@list@R	45
COMMAND\line@list@stuff	41, 48
COMMAND\line@margin	35
COMMAND\line@marginR	35, 147
COMMAND\line@numR	40
COMMAND\lineation	17, 151, 155
COMMAND\lineation*	17, 35, 149
COMMAND\lineationR	17, 34, 151, 154
COMMAND\linenum	82
COMMAND\linenum@out	84
COMMAND\linenum@outR	47
COMMAND\linenumOnlyPagesForColumns	18, 36, 155
COMMAND\linenumOnlyPagesForColumnsR	18, 36, 155, 157
COMMAND\linenumannotationothersidetrue	156
COMMAND\linenumLevenifblanktrue	18, 153
COMMAND\linenumRevenifblank	153
COMMAND\linenumRevenifblanktrue	18
COMMAND\linenumannotation	155
COMMAND\linenumlist	17, 128, 154
COMMAND\linenumlistR	17, 128, 154
COMMAND\linenumstyle	17, 128, 154
COMMAND\linenumstyle*	128
COMMAND\linenumstyleR	17, 128
COMMAND\linenumincrement	17, 149, 152, 153
COMMAND\linenummargin	18, 35, 147, 152, 153
COMMAND\linenummargin*	18, 35, 152
COMMAND\linenummarginColumns	18, 36
COMMAND\linenummarginColumns*	18
COMMAND\linenummarginColumnsR	18
COMMAND\linenummarginR	18, 35, 152
COMMAND\linenumrepR	38, 147
COMMAND\linesinpar@listL	46, 112
COMMAND\linesonpage@listL	47, 113
COMMAND\lock@off	44
COMMAND\lock@on	43
COMMAND\lockoff	157
COMMAND\lockon	157
COMMAND\mainmatter	2, 13, 22, 152
COMMAND\makeatletter	66
COMMAND\markboth	129, 154
COMMAND\maxchunks	7, 20, 91, 92
COMMAND\maxdimen	78, 79
COMMAND\maxhnotesX	14
COMMAND\memorydump	16, 34
COMMAND\movecolumnsonrightpage	100
COMMAND\msdata	154
COMMAND\n@num	151
COMMAND\new@lineL	48
COMMAND\new@lineR	49
COMMAND\newhookcommand@series	52
COMMAND\newif	151
COMMAND\newpage	111, 112, 150
COMMAND\newseries	53
COMMAND\newseries@	50

COMMAND\newseries@par	50, 52, 53
COMMAND\next@line@list@stuffR	48, 155
COMMAND\noeledxxx	150
COMMAND\nomark@	51
COMMAND\nomaxlines	47
COMMAND\notesXwidthliketwocolumns	9, 150
COMMAND\num@lines	77
COMMAND\num@lines(R)	57
COMMAND\numberingR	32
COMMAND\numberlinefalse	7, 156
COMMAND\numberlinetrue	156
COMMAND\numberonlyfirstinline	148
COMMAND\numberpstartfalse	17, 155
COMMAND\numberpstarttrue	17, 147, 152, 155
COMMAND\numberstanztrue	156
COMMAND\one@line	57, 86
COMMAND\one@lineR	57
COMMAND\onlyXside	128
COMMAND\onlysideX	14, 78, 80, 151, 154, 156
COMMAND\page@action	43, 152
COMMAND\pagenumbering	152
COMMAND\pages	13
COMMAND\pagetotal	107, 150
COMMAND\par@line	77
COMMAND\par@line(R)	57
COMMAND\par@sync@option	24
COMMAND\parindentX	156
COMMAND\parledgroup@	122
COMMAND\parledgroup@beforenotes@save	125
COMMAND\parledgroup@beforenotesL	125
COMMAND\parledgroup@beforenotesR	125
COMMAND\parledgroup@correction@notespacing	125
COMMAND\parledgroup@correction@notespacing@final	124
COMMAND\parledgroup@correction@notespacing@init	124
COMMAND\parledgroup@notes@endL	123
COMMAND\parledgroup@notes@endR	123
COMMAND\parledgroup@notes@startL	122
COMMAND\parledgroup@notes@startR	122
COMMAND\parledgroup@notespacing@correction	124, 125
COMMAND\parledgroup@notespacing@correction@accumulated	125
COMMAND\parledgroup@notespacing@correction@modulo	125
COMMAND\parledgroup@notespacing@set@correction	124
COMMAND\parledgroup@series	122
COMMAND\parledgroup@type	122
COMMAND\parledgroupnotespacing	127
COMMAND\parledgrouptrue	21
COMMAND\patchcmd	152
COMMAND\pausenumbering	9, 10, 33, 34, 153
COMMAND\pend	3, 7, 10, 11, 19–21, 49, 53, 57, 61–63, 90, 102, 149–152
COMMAND\pendL	149
COMMAND\pendR	149
COMMAND\pends	19
COMMAND\perpage	14
COMMAND\prev@nopbR	122

COMMAND\prev@pbR	122
COMMAND\prevpgstyle	25
COMMAND\print@Xnotes	78
COMMAND\print@Xnotes@forpages	78, 79, 151
COMMAND\print@columnseparator	96, 149
COMMAND\print@eledsectionL	65
COMMAND\print@leftcolumn	100
COMMAND\print@line	65
COMMAND\print@lineL	65
COMMAND\print@notesX@forpages	151
COMMAND\print@rightcolumn	100
COMMAND\printlinenumR	128
COMMAND\printlinesR	153
COMMAND\pstart	3, 7, 10, 11, 17, 19–21, 34, 43, 49, 53, 57, 58, 61, 63, 90, 91, 94, 102, 147, 149, 151, 152, 154–156
COMMAND\pstartL	63, 149
COMMAND\pstartR	63, 148, 149
COMMAND\pstartinfootnote	151
COMMAND\pstarts@typeset@L	94
COMMAND\raw@text	90
COMMAND\read@linelist	41, 131, 152
COMMAND\ref@reg	45
COMMAND\ref@regR	45, 152
COMMAND\relax	128
COMMAND\reledmac	152
COMMAND\renewcommand	20, 127
COMMAND\resumenumbering	9, 10, 33, 34, 149, 153
COMMAND\resumenumberingR	150
COMMAND\rightlinenumR	39, 147
COMMAND\rightlinenumannotationR	40
COMMAND\sameword	82
COMMAND\section	147
COMMAND\section@num	30
COMMAND\selectlanguage	19, 89
COMMAND\set@line	49, 152
COMMAND\set@line@action	43, 152
COMMAND\setRlineflag	18, 127
COMMAND\setgoalfraction	13, 127
COMMAND\sethangingsymbol	20
COMMAND\setline	43, 49, 152
COMMAND\setlinenum	43, 49, 152
COMMAND\setmsdataposition	154
COMMAND\setnoteposition...	102
COMMAND\setparledgroupnotespacing	127, 152
COMMAND\setposition...	102
COMMAND\setstanzaindents	9, 20, 127, 129
COMMAND\setwidthliketwocolumns	102
COMMAND\sidenotemargin	21, 149
COMMAND\sidenotemargin*	21, 149
COMMAND\sidenotemarginR	21, 154
COMMAND\skipnumbering	17, 151
COMMAND\sloppy	8
COMMAND\stanza	7, 9, 17, 20, 87, 128, 129, 148
COMMAND\stanzanumtrue	21

COMMAND\startlock	49, 152
COMMAND\startsub	49, 152
COMMAND\sub@action	43, 152
COMMAND\sub@off	84
COMMAND\sub@on	84
COMMAND\subline@numR	40
COMMAND\sublinenumberstyle	17, 128
COMMAND\sublinenumberstyle*	128
COMMAND\sublinenumberstyleR	17, 128
COMMAND\sublinenumincrement	149, 152, 153
COMMAND\sublinenumrepR	38, 147
COMMAND\syntaxonly	153
COMMAND\sza@0@	20
COMMAND\textheight	14
COMMAND\textwidth	55
COMMAND\the@labelX	153
COMMAND\theledlanguageL	89
COMMAND\theledlanguageR	89
COMMAND\thepage	22
COMMAND\thepstartL	17, 148
COMMAND\thepstartR	17, 148
COMMAND\thestanzaL	21
COMMAND\thestanzaR	21
COMMAND\this@section@last@page@numR	43
COMMAND\vbox	58
COMMAND\vskip	123
COMMAND\vsplit	77
COMMAND\widthliketwocolumns	9
COMMAND\widthliketwocolumnsfalse	9
COMMAND\widthliketwocolumnstrue	9
COMMAND\xflagref	153
COMMAND\xright@appenditem	51
COMMAND\xspace	23
COMMAND\xxxfootstart	102
ENVIRONMENTLeftside	10, 55, 100
ENVIRONMENTRightside	10, 56, 100
ENVIRONMENTastanza	20, 87, 128, 129, 152, 153, 155, 156
ENVIRONMENTcolumns	32, 153
ENVIRONMENTledgroup	6, 153
ENVIRONMENTleft	17
ENVIRONMENTpages	26, 32, 54, 129, 151, 153, 154
ENVIRONMENTpairs	54, 151, 153
PACKAGEEDMAC	129
PACKAGEEDSTANZA	129
PACKAGEEledmac	151
PACKAGEEledpar	151
PACKAGETABMAC	129
PACKAGEbabel	19, 88–90, 153, 156
PACKAGEcalc	156
PACKAGEeledmac	4, 91, 126, 127, 129, 149, 151
PACKAGEeledpar	4, 5, 13, 37, 127, 128, 149, 150
PACKAGEetoolbox	30, 96, 122
PACKAGEledmac	5
PACKAGEledpar	1, 5

PACKAGEMemoir	129
PACKAGEMusixtex	149
PACKAGEperpage	2, 14, 128, 153
PACKAGEpolyglossia	19, 88–90, 153, 156, 157
PACKAGEreledmac	1, 3, 5–7, 9, 10, 13, 14, 17–23, 25, 26, 30, 32, 35, 37, 40, 41, 43–53, 65, 74, 83, 84, 86, 105, 110, 127, 128, 151–156
PACKAGEreledpar	1, 3–7, 9–13, 20–25, 34, 40, 41, 46–48, 50, 52, 53, 83, 97, 127–129, 151
PACKAGEsetspace	2, 22
PACKAGESyntonly	153
PACKAGEXkeyval	23
PACKAGEX	153

A

<code>\absline@numR</code>	<u>1</u>
<code>\actionlines@listR</code>	<u>1</u>
<code>\actions@listR</code>	<u>1</u>
<code>\add@apparatusL</code>	<u>1</u>
<code>\add@apparatusR</code>	<u>1</u>
<code>\add@inserts@nextR</code>	<u>1</u>
<code>\add@insertsR</code>	<u>1</u>
<code>\add@penaltiesL</code>	<u>1</u>
<code>\add@penaltiesR</code>	<u>1</u>
<code>\advanceline</code>	<u>1</u>
<code>\affixline@numR</code>	<u>1</u>
<code>\affixpstart@numL</code>	<u>1</u>
<code>\affixpstart@numR</code>	<u>1</u>
<code>\affixside@noteR</code>	<u>1</u>
<code>\aftercolumnseparator</code>	<u>1, 9</u>
<code>\annot@listR</code>	<u>1</u>
<code>\araw@textfalse</code>	<u>1</u>
<code>\araw@texttrue</code>	<u>1</u>
<code>astanza (env.)</code>	<u>20</u>
<code>\AtBeginPairs</code>	<u>1, 8</u>
<code>\AtEveryPstartCall</code>	<u>1</u>
<code>\autopar</code>	<u>19</u>

B

<code>\bbl@set@language</code>	<u>1</u>
<code>\beforecolumnseparator</code>	<u>1, 9</u>
<code>\beginnumbering</code>	<u>16</u>
<code>\beginnumberingR</code>	<u>1</u>

C

<code>\c@firstlinenumR</code>	<u>1</u>
<code>\c@firstsublinenumR</code>	<u>1</u>
<code>\c@linenumincrementR</code>	<u>1</u>
<code>\c@sublinenumincrementR</code>	<u>1</u>
<code>\ch@ck@l@ckR</code>	<u>1</u>
<code>\ch@cksub@l@ckR</code>	<u>1</u>
<code>\chapter</code>	<u>1</u>
<code>\chapterinpages</code>	<u>1</u>
<code>\check@goal</code>	<u>1</u>
<code>\check@pstarts</code>	<u>1</u>
<code>\checkpageL</code>	<u>1</u>

<code>\checkpageR</code>	<u>1</u>
<code>\checkpb@columns</code>	<u>1</u>
<code>\checkpbL</code>	<u>1</u>
<code>\checkpbR</code>	<u>1</u>
<code>\checkraw@text</code>	<u>1</u>
<code>\checkverseL</code>	<u>1</u>
<code>\checkverseR</code>	<u>1</u>
<code>\clearl@dleftpage</code>	<u>1</u>
<code>\clearl@drightpage</code>	<u>1</u>
<code>\cleartoevenpage</code>	<u>1</u>
<code>\cleartol@devenpage</code>	<u>1</u>
<code>\columnrulewidth</code>	<u>1, 8</u>
<code>\Columns</code>	<u>1, 8</u>
<code>\columns@position</code>	<u>1</u>
<code>\Columns@print@after@pend</code>	<u>1</u>
<code>\Columns@print@before@pstart</code>	<u>1</u>
<code>\columnseparator</code>	<u>1, 8</u>
<code>\columnsposition</code>	<u>1, 9</u>
<code>\correct@footinsX@box</code>	<u>1</u>
<code>\correct@Xfootins@box</code>	<u>1</u>
<code>\countLline</code>	<u>1</u>
<code>\countRline</code>	<u>1</u>
<code>\critext</code>	<u>1</u>

D

<code>\do@actions@fixedcodeR</code>	<u>1</u>
<code>\do@actions@nextR</code>	<u>1</u>
<code>\do@actionsR</code>	<u>1</u>
<code>\do@ballastR</code>	<u>1</u>
<code>\do@insidelineLhook</code>	<u>1</u>
<code>\do@insidelineRhook</code>	<u>1</u>
<code>\do@lineL</code>	<u>1</u>
<code>\do@lineLhook</code>	<u>1</u>
<code>\do@lineR</code>	<u>1</u>
<code>\do@lineRhook</code>	<u>1</u>
<code>\do@lockoff</code>	<u>1</u>
<code>\do@lockoffR</code>	<u>1</u>
<code>\do@lockon</code>	<u>1</u>
<code>\do@lockonR</code>	<u>1</u>
<code>\doinsidelineLhook</code>	<u>1, 20</u>
<code>\doinsidelineRhook</code>	<u>1, 20</u>
<code>\dolineLhook</code>	<u>1, 20</u>
<code>\dolineRhook</code>	<u>1, 20</u>
<code>\dump@pstartL@pc</code>	<u>1</u>
<code>\dump@pstartR@pc</code>	<u>1</u>

E

<code>\edlabel</code>	<u>1</u>
<code>\edtext</code>	<u>1</u>
<code>\edtext@later</code>	<u>1</u>
<code>\edtext@now</code>	<u>1</u>
<code>\edtextlater</code>	<u>1, 15</u>
<code>\edtextnow</code>	<u>1, 15</u>
<code>\eled@sectioningR@out</code>	<u>1</u>

<code>\eledsection@correcting@skip</code>	<u>1</u>
<code>\eledsectmark</code>	<u>1</u> , 22
<code>\eledsectnotoc</code>	<u>1</u> , 22
<code>\endlock</code>	<u>1</u>
<code>\endnumbering</code>	<u>1</u> , 16
<code>\endnumberingR</code>	<u>1</u>
<code>\endsub</code>	<u>1</u>
environments:	
<code>astanza</code>	20
<code>Leftside</code>	16
<code>pages</code>	11
<code>pairs</code>	8
<code>Rightside</code>	16

F

<code>\f@x@l@cksR</code>	<u>1</u>
<code>\finish@Pages@notes</code>	<u>1</u>
<code>\first@linenum@out@Rfalse</code>	<u>1</u>
<code>\first@linenum@out@Rtrue</code>	<u>1</u>
<code>\firstlinenum</code>	<u>1</u> , 17
<code>\firstlinenum*</code>	<u>1</u> , 17
<code>\firstlinenumR</code>	<u>1</u> , 17
<code>\firstsublinenum</code>	<u>1</u> , 17
<code>\firstsublinenum*</code>	<u>1</u> , 17
<code>\firstsublinenumR</code>	<u>1</u> , 17
<code>\fix@page</code>	<u>1</u>
<code>\flag@end</code>	<u>1</u>
<code>\flag@start</code>	<u>1</u>
<code>\flush@notesR</code>	<u>1</u>
<code>\footnote@reading</code>	<u>1</u>
<code>\footnote@typeset</code>	<u>1</u>
<code>\footnoteXmk</code>	15
<code>\footnoteXnomk</code>	15

G

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

H

<code>\hidenumbering</code>	17
-----------------------------------	----

I

<code>\if@getnextbox</code>	<u>1</u>
<code>\if@pstarts</code>	<u>1</u>
<code>\if@sidenotesmarginpage</code>	<u>1</u>
<code>\ifaraw@text</code>	<u>1</u>

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

L

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

<code>\l@dsetuprawboxes</code>	1
<code>\l@dskipversenumberR</code>	1
<code>\l@duselanguage</code>	1
<code>\l@dzeromaxlinecounts</code>	1
<code>\labelref@listR</code>	1
<code>\last@page@enumR</code>	1
<code>\Lcolwidth</code>	1, 8, 11
<code>\led@err@BadLeftRightPstarts</code>	1
<code>\led@err@Columns@InsideEnv</code>	1
<code>\led@err@Columns@WithoutEnv</code>	1
<code>\led@err@Columns@WithoutLeftside</code>	1
<code>\led@err@Columns@WithoutRightside</code>	1
<code>\led@err@LeftOnRightPage</code>	1
<code>\led@err@Leftside@PreviousNotPrinted</code>	1
<code>\led@err@Pages@InsideEnv</code>	1
<code>\led@err@Pages@WithoutEnv</code>	1
<code>\led@err@Pages@WithoutLeftside</code>	1
<code>\led@err@Pages@WithoutRightside</code>	1
<code>\led@err@polyglossiaTooOld</code>	1
<code>\led@err@RightOnLeftPage</code>	1
<code>\led@err@Rightside@PreviousNotPrinted</code>	1
<code>\led@err@TooManyPstarts</code>	1
<code>\led@error@edtext@later@now</code>	1
<code>\led@error@fail@patch@mempnum</code>	1
<code>\led@error@fail@patch@outputpage</code>	1
<code>\led@error@fail@patch@pagenumbering</code>	1
<code>\led@error@fail@patch@thepage</code>	1
<code>\led@error@missing@numbering</code>	1
<code>\led@error@note@called@onleftside</code>	1
<code>\led@error@note@called@onrightside</code>	1
<code>\led@nopbnumR</code>	1
<code>\led@nopbR</code>	1
<code>\led@pbnumR</code>	1
<code>\led@pbR</code>	1
<code>\led@warn@ChangeSyncOption</code>	1
<code>\led@warn@setting@in@rightside</code>	1
<code>\lednopbnum</code>	1
<code>\lednopbnumR</code>	1
<code>\ledpbnumR</code>	1
<code>\ledpbR</code>	1
<code>\ledstrutL</code>	1
<code>\ledstrutR</code>	1
<code>\ledthegoal</code>	1
<code>\leftlinenumannotationR</code>	1
<code>\leftlinenumR</code>	1
<code>\Leftpagehook</code>	1, 20
<code>\leftpstartnumL</code>	1
<code>\leftpstartnumR</code>	1
<code>Leftside (env.)</code>	16
<code>\Leftsidehook</code>	1
<code>\Leftsidehookend</code>	1
<code>\line@list@stuffR</code>	1
<code>\line@listR</code>	1
<code>\line@marginR</code>	1

<code>\line@numR</code>	1
<code>\lineation*</code>	1 , 17
<code>\lineationR</code>	1 , 17
<code>\linenum@outR</code>	1
<code>\linenumberLevenifblanktrue</code>	18
<code>\linenumberRevenifblanktrue</code>	18
<code>\linenumberstyle*</code>	1 , 17
<code>\linenumberstyleR</code>	1 , 17
<code>\linenumincrement</code>	1 , 17
<code>\linenumincrement*</code>	1 , 17
<code>\linenumincrementR</code>	1 , 17
<code>\linenummargin</code>	1
<code>\linenummargin*</code>	1 , 18
<code>\linenummarginColumns</code>	1 , 18
<code>\linenummarginColumns*</code>	1 , 18
<code>\linenummarginColumnsR</code>	1 , 18
<code>\linenummarginR</code>	1 , 18
<code>\linenumOnlyPagesForColumns</code>	1 , 18
<code>\linenumOnlyPagesForColumnsR</code>	1 , 18
<code>\linenumrepR</code>	1
<code>\linesinpar@listL</code>	1
<code>\linesinpar@listR</code>	1
<code>\list@clearing@regR</code>	1
<code>\list@pstartL@pc</code>	1
<code>\list@pstartR@pc</code>	1
<code>\lock@off</code>	1

M

<code>\maxchunks</code>	1 , 7
<code>\maxlinesinpar@list</code>	1
<code>\memorydump</code>	16
<code>\memorydumpL</code>	1
<code>\memorydumpR</code>	1

N

<code>\n@num</code>	1
<code>\namebox</code>	1
<code>\new@lineL</code>	1
<code>\new@lineR</code>	1
<code>\newnamebox</code>	1
<code>\newnamecount</code>	1
<code>\newseries@par</code>	1
<code>\next@line@list@stuffR</code>	1
<code>\notesXwidthliketwocolumns</code>	9
<code>\num@linesR</code>	1
<code>\numberpstartfalse</code>	17
<code>\numberpstarttrue</code>	17
<code>\numpagelinesL</code>	1
<code>\numpagelinesR</code>	1

O

<code>\one@lineR</code>	1
<code>\onlysideX</code>	14
<code>optionadvancedshiftedpstarts</code>	11 , 12 , 24

optionauxdir	153
optioncontinuousnumberingwithcolumns	9, 153, 155
optionmainmatter	120
optionmovecolumnsonrightpage	10, 154
optionnomaxlines	11, 12, 24
optionnopbinverse	155
optionnosyncpstarts	12, 24, 117
optionprevpgnotnumbered	154
optionprevpgstyle	154
optionsameparallelpagelnumber	151, 156
optionshiftedpstarts	6, 12, 24, 148
optionshiftedverses	147, 148
optionsidenotesmarginpage	21, 25, 155
optionwidthliketwocolumns	9, 155

P

<code>\page@action</code>	1
<code>\page@numR</code>	1
<code>\Pages</code>	1, 11
<code>pages (env.)</code>	11
<code>\Pages@mainmatter</code>	1
<code>pairs (env.)</code>	8
<code>\par@lineR</code>	1
<code>\parledgroup@</code>	1
<code>\parledgroup@beforenotes@save</code>	1
<code>\parledgroup@beforenotesL</code>	1
<code>\parledgroup@beforenotesR</code>	1
<code>\parledgroup@correction@notespacing</code>	1
<code>\parledgroup@correction@notespacing@final</code>	1
<code>\parledgroup@correction@notespacing@init</code>	1
<code>\parledgroup@notes@startL</code>	1, 1
<code>\parledgroup@notes@startR</code>	1, 1
<code>\parledgroup@notespacing@correction</code>	1
<code>\parledgroup@notespacing@set@correction</code>	1
<code>\parledgroupseries@</code>	1
<code>\parledgroupstype@</code>	1
<code>\pausenumbering@page@numR</code>	1
<code>\pausenumberingR</code>	1
<code>\pend</code>	19
<code>\pendL</code>	1
<code>\pendR</code>	1
<code>\prev@nopbR</code>	1
<code>\prev@pbR</code>	1
<code>\prevpgstyle</code>	1
<code>\print@columnseparator</code>	1
<code>\print@eledsectionL</code>	1
<code>\print@eledsectionR</code>	1
<code>\print@leftcolumn</code>	1
<code>\print@lineL</code>	1
<code>\print@lineR</code>	1
<code>\print@notesX@forpages</code>	1
<code>\print@rightcolumn</code>	1
<code>\print@Xnotes@forpages</code>	1
<code>\pstart</code>	19

<code>\pstartL</code>	<u>1</u>
<code>\pstartR</code>	<u>1</u>

R

<code>\Rcolwidth</code>	<u>1</u> , <u>8</u> , <u>11</u>
<code>\read@linelist</code>	<u>1</u>
<code>\reledpar@error</code>	<u>1</u>
<code>\reledpar@warning</code>	<u>1</u>
<code>\restore@notesX@settings</code>	<u>1</u>
<code>\restore@pstartL@pc</code>	<u>1</u>
<code>\restore@pstartR@pc</code>	<u>1</u>
<code>\restore@Xnotes@settings</code>	<u>1</u>
<code>\resumenumberingR</code>	<u>1</u>
<code>\rightlinenumannotationR</code>	<u>1</u>
<code>\rightlinenumR</code>	<u>1</u>
<code>\Rightpagehook</code>	<u>1</u> , <u>20</u>
<code>\rightpstartnumL</code>	<u>1</u>
<code>\rightpstartnumR</code>	<u>1</u>
<code>Rightside (env.)</code>	<u>16</u>
<code>\Rightsidehook</code>	<u>1</u>
<code>\Rightsidehookend</code>	<u>1</u>
<code>\Rlineflag</code>	<u>1</u>

S

<code>\save@familiarfootnote@number</code>	<u>1</u>
<code>\save@section@number</code>	<u>1</u>
<code>\section@numR</code>	<u>1</u>
<code>\selectlanguage</code>	<u>1</u>
<code>\set@line</code>	<u>1</u>
<code>\set@line@action</code>	<u>1</u>
<code>\set@sectcountR</code>	<u>1</u>
<code>\setgoalfraction</code>	<u>1</u> , <u>13</u>
<code>\sethangingsymbol</code>	<u>20</u>
<code>\setline</code>	<u>1</u>
<code>\setlinenum</code>	<u>1</u>
<code>\setnamebox</code>	<u>1</u>
<code>\setnotepositionliketwocolumns@C</code>	<u>1</u>
<code>\setnotepositionliketwocolumns@L</code>	<u>1</u>
<code>\setnotepositionliketwocolumns@R</code>	<u>1</u>
<code>\setpositionliketwocolumns@C</code>	<u>1</u>
<code>\setpositionliketwocolumns@L</code>	<u>1</u>
<code>\setpositionliketwocolumns@R</code>	<u>1</u>
<code>\setRlineflag</code>	<u>18</u>
<code>\setwidthliketwocolumns</code>	<u>1</u>
<code>\sidenote@marginR</code>	<u>1</u>
<code>\sidenotemargin*</code>	<u>1</u>
<code>\sidenotemarginR</code>	<u>1</u>
<code>\skip@lockoff</code>	<u>1</u>
<code>\skipnumbering</code>	<u>1</u> , <u>17</u>
<code>\startlock</code>	<u>1</u>
<code>\startsub</code>	<u>1</u>
<code>\sub@action</code>	<u>1</u>
<code>\subline@numR</code>	<u>1</u>
<code>\sublinenumberstyle*</code>	<u>1</u> , <u>17</u>

<code>\sublinenumberstyleR</code>	1 , 17
<code>\sublinenumincrement</code>	1 , 17
<code>\sublinenumincrement*</code>	1 , 17
<code>\sublinenumincrementR</code>	1 , 17
<code>\sublinenumrepR</code>	1
T	
<code>\theledlanguageL</code>	1
<code>\theledlanguageR</code>	1
<code>\thepstartL</code>	17
<code>\thepstartR</code>	17
<code>\thestanzaL</code>	1 , 21
<code>\thestanzaR</code>	1 , 21
<code>\this@section@last@page@numR</code>	1
<code>\this@section@page@numR</code>	1
U	
<code>\unhnamebox</code>	1
<code>\unvnamebox</code>	1
<code>\usernamecount</code>	1
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.	
General: Added section of babel related code	88
Fix babel problems	1
\Columns: Added \l@dchecklang and \l@duselanguage to \Columns	94
\Pages: Added \l@duselanguage to \Pages	106
v0.3.0.	
General: Added \do@lineLhook and \do@lineRhook	67
Added hooks into Leftside environment	55
Reorganize for ledarab	1
\affixline@numR: Changed \affixline@numR to match new eledmac	72
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR	70
\do@lineL: Added \do@lineLhook to \do@lineL	64
Simplified \do@lineL by using macros for some common code	64
\do@lineR: Changed \do@lineR similarly to \do@lineL	67
\flag@end: Removed extraneous spaces from \flag@end	49
\ifledRcol: Moved \ifl@pairing to eledmac	26
\ifpst@rtedR: Moved \ifpst@rtedL to eledmac	30
\l@dlinenumR: Simplified \leftlinenumR and \rightlinenumR by introducing \l@dlinenumR	39
\l@dnumpstartsR: Moved \l@dnumpstartsL to eledmac	91
\ledstrutR: Added \ledstrutL and \ledstrutR	111
\Pages: Added \ledstrutL to \Pages	106
Added \ledstrutR to \Pages	108
\Rightsidehookend: Added \Leftsidehook, \Leftsidehookend, \Rightsidehook and \Rightsidehookend	56
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR	38
v0.3.a.	
General: Minor \linenummargin fix	1
\line@marginR: Do not just set \line@marginR in \linenummargin	35
v0.3.b.	
General: Improved parallel page balancing	1
\Pages: Added \l@dminpagelines calculation for succeeding page pairs	109
v0.3.c.	
General: Compatilby with Polyglossia	1
v0.4.0.	
General: No more ledparpatch. All patches are now in the main file.	1
v0.5.0.	
General: Corrections about \section and other titles in numbered sections	1
v0.6.0.	
General: Be able to us \chapter in parallel pages.	1
v0.7.0.	
General: Add shiftedverses option which makes there is no blank between two parallel verses with inequal length.	1
v0.8.0.	
General: Possibility to have a symbol on each hanging of verses, like in the french typography. Redefine the commande \hangingsymbol to define the character.	1
v0.9.0.	
General: Possibility to number \pstart.	17
Possilby to number the pstart with the commands \numberpstarttrue.	1
\ifledRcol: Moved \iflledRcol and \ifnumberingR to eledmac	26

v0.9.1.	General: The numbering of the pstarts restarts on each <code>\beginnumbering</code>	1
v0.9.2.	General: Debug: with <code>\Columns</code> , the hanging indentation now runs on the left columns and the hanging symbol is shown only when <code>\stanza</code> is used.	1
v0.9.3.	General: <code>\thepstartL</code> and <code>\thepstartR</code> use now <code>\bfseries</code> and not <code>\bf</code> , which is deprecated and makes conflicts with memoir class.	1
v0.10.0.	General: <code>\edlabel</code> commands on the right side are now correctly indicated.	1
	<code>\edlabel</code> commands which start a paragraph are now put in the right place.	1
v0.11.0.	General: Change <code>\do@lineL</code> and <code>\do@lineR</code> to allow line numbering by pstart (like in eledmac 0.15).	64
	Lineation can be by pstart (like in eledmac 0.15).	34
	New management of hangingsymbol insertion, preventing undesirable insertions.	86
	<code>\affixline@numR</code> : Changed <code>\affixline@numR</code> to allow to disable line numbering (like in eledmac 0.15).	72
	<code>\Columns</code> : Line numbering by pstart.	95
	<code>\get@nextboxR</code> : Change <code>\get@nextboxL</code> and <code>\get@nextboxR</code> to allow to disable line numbering (like in eledmac 0.15).	117
	Pstart number can be printed in side	118
	<code>\inserthangingsymbolR</code> : Prevent the column separator for hanging verse from shifting	86
v0.12.0.	General: New management of hangingsymbol insertion, preventing undesirable insertions.	86
v1.0.0.	General: Compatibility with eledmac. Change name to eledpar.	1
	Debug in lineation by pstart	34
v1.0.1.	General: Correction on <code>\numberonlyfirstinline</code> with lineation by pstart or by page.	1
v1.1.0.	General: The <code>shiftedverses</code> options becomes <code>shiftedpstarts</code>	1
	<code>\pstartR</code> : Add <code>\labelpstarttrue</code> (from eledmac).	57
v1.1.1.	<code>\pstartR</code> : Correct <code>\pstartR</code> bug introduced by 1.1.	57
v1.1.2.	<code>\affixside@noteR</code> : Remove spurious space between line number and line content	85
v1.2.0.	General: Support for <code>\led<section></code> commands in parallel texts.	1
v1.2.1.	<code>\set@sectcountR</code> : For the right section, the counter is defined only once.	32
v1.3.0.	<code>\edtext</code> : Manage RTL language.	49
v1.3.2.	General: Debug with some classes.	1
v1.3.3.	General: Debugging the left notes of the right column.	85
v1.3.4.	General: Allow use of commands in sidenotes, as introduced by eledmac 1.0.	85
v1.4.0.	General: Added <code>\do@insidelineLhook</code> and <code>\do@insidelineRhook</code>	67
v1.4.1.	General: Enable the use of <code>stanzaindentsrepetition</code> within <code>astanza</code> environment.	87

v1.4.3.	
General: Corrects a false hanging verse when a verse is exactly the length of a line.	1
<code>\inserthangingsymbolR</code> : Hanging verse is no longer automatically flush right.	86
<code>\pendL</code> : Spurious spaces in <code>\pendL</code>	61
<code>\pendR</code> : Spurious spaces in <code>\pstartR</code>	62
<code>\pstartR</code> : Spurious spaces in <code>\pstartL</code> and <code>\pstartR</code>	57
v1.5.0.	
General: Add, as in <code>eledmac</code> , features to manage page breaks.	1
<code>\sublinenumincrement*</code> : Add starred version of <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> to change both Left and Rightside.	37
v1.6.0.	
General: Add tool and documentation for parallel ledgroups	21
v1.7.0.	
General: Add, as in <code>eledmac</code> , features to make crossrefs with <code>pstart</code> numbers.	1
v1.8.0.	
General: <code>\beginnumbering</code> is defined only on <code>eledmac</code> , not on <code>eledpar</code>	30
<code>\l@dlsnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in <code>eledmac</code>	85
Add <code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code>	9
Add <code>\columnspan</code>	9
Add, as in <code>eledmac</code> , new system of sectioning commands.	1
Add, as in <code>eledmac</code> , option to insert something after <code>\pends</code> / verses.	1
Add, as in <code>eledmac</code> , option to insert something between <code>\pstarts</code> / verse.	1
Change <code>\do@lineR</code> and <code>\do@lineL</code> to allow new sectioning commands.	64
Compatibility with <code>musixtex</code>	1
Debug <code>eledmac</code> sectioning command after using <code>\resumenumbering</code>	1
New sectioning commands, as in <code>eledmac</code>	22
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.	88
<code>\Columns</code> : Modify <code>\Columns</code> to enable to add section's title.	93
Suppress <code>\l@dchecklang</code> from <code>\Columns</code>	94
<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.	89
<code>\Pages</code> : Modify <code>\Pages</code> to enable to add section's title.	104
<code>\pendL</code> : As in <code>eledmac</code> , <code>\pendL</code> can have an optional argument.	61
<code>\pendR</code> : As in <code>eledmac</code> , <code>\pendR</code> can have an optional argument.	62
<code>\print@columnseparator</code> : Move some code of <code>\Columns</code> to <code>\print@columnseparator</code>	96
<code>\pstartR</code> : As in <code>eledmac</code> , <code>\pendL</code> and <code>\pendR</code> can have an optional argument.	57
<code>\sidenotemarginR</code> : <code>\sidenotemargin</code> is now directly defined in <code>eledmac</code> to be able to manage <code>eledpar</code>	84
Add <code>\sidenotemargin*</code>	84
<code>\theledlanguageR</code> : Correct left/right language setting with <code>polyglossia</code>	90
v1.8.1.	
<code>\do@lineL</code> : Fix a bug with critical notes a the beginning of a page, (maybe added by v1.8.0) (?).	64
<code>\do@lineR</code> : Fix a bug with critical notes a the beginning of a page, added by v1.8.0 (?).	67
v1.8.2.	
General: Debug <code>\eledxxx</code> with some paper sizes	1
Debug left and side note (bugs added by 1.8.0)	1
<code>\flag@end</code> : <code>\flag@start</code> and <code>\flag@end</code> are now defined only one time for <code>eledmac</code> and <code>eledpar</code>	49
<code>\lineation*</code> : Add <code>\lineation*</code>	35

<code>\eledpar@error</code> : Errors specific to eledpar send to eledpar handbook	26
v1.8.3.	
General: Add <code>\noeledxxx</code> , as in eledmac	1
<code>\doinsidelineRhook</code> : Added <code>\dolineLhook</code> , <code>\dolineRhook</code> , <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code>	66
<code>\Pages</code> : Debug blank pages when using optional argument in the last <code>\pend</code> .	104
<code>\pausenumbering@page@numR</code> : Debug <code>\resumenumberingR</code>	33
v1.9.0.	
General: Add <code>\AtBeginPairs</code> macro.	8
Compatibility with <code>\Xnoteswidthliketwocolumns</code> and <code>\notesXwidthliketwocolumns</code>	1
<code>\ifwidthliketwocolumns</code> : Added <code>widthliketwocolumns</code> option	25
<code>\theledlanguageR</code> : Debug left/right language switching with polyglossia. Do not write in <code>.aux</code> file when setting left/right lines.	90
v1.9.1.	
<code>\ifledRcol</code> : Moved <code>\ifl@dpadding</code> to eledmac	26
v1.10.0.	
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
<code>\Pages</code> : Debug wrong pages splitting when no optional argument is used in last <code>\pend</code> (bug was added in v.1.8.3).	104
Debug wrong parallel pages synchronization when an <code>\edtext</code> falls across two pages.	104
v1.10.1.	
<code>\next@line@list@stuffR</code> : Revert modification of 1.4.2, which makes bugs with numbering. Leave vertical mode to solve spurious space before <code>minipage</code> .	48
v1.11.0.	
General: Compatibility of standard footnotes with some biblatex styles.	1
<code>\edtext</code> : <code>\critext</code> and <code>\edtext</code> are now defined only in eledmac.	49
v1.12.0.	
General: Compatibility with Lua \TeX RTL languages.	1
<code>\Columns</code> : Add <code>\l@dprintingcolumnstrue</code>	93
<code>\edlabel</code> : <code>\edlabel</code> and <code>\edindex</code> works now with hyperref when using <code>eledpar</code> . <code>\edlabel</code> is now defined only one time for both eledmac and eledpar	83
<code>\Pages</code> : Add <code>\l@dprintingpagestrue</code>	104
<code>\print@eledsectionL</code> : Compatibility with Lua \TeX RTL languages.	65
<code>\print@eledsectionR</code> : Compatibility with Lua \TeX RTL languages.	68
<code>\print@lineL</code> : Compatibility with Lua \TeX RTL languages.	65
v1.12.1.	
<code>\print@eledsectionL</code> : Fix bug with Lua \TeX RTL <code>\eledsection</code> .	65
v1.13.0.	
General: Enable the use of optional argument of <code>&</code> in <code>astanza</code> environment.	87
Fix a bug in <code>shiftedpstarts</code> when size difference between <code>pstarts</code> 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
<code>\clearl@drightpage</code> : Use <code>\newpage</code> instead of <code>\clearpage</code> .	112
<code>\ifledRcol</code> : Remove false boolean settings which are not needed.	26
<code>\Pages</code> : Prevent false overfull hboxes when using <code>\Pages</code> outside of pages environment.	105
When using <code>shiftedpstarts</code> option, a <code>\l@dleftbox</code> with a null height will advance the <code>\pagetotal</code> in any case.	104

v1.13.1.	
\correct@footinsX@box: Call \correct@footinsX@box and	
\correct@Xfootins@box directly in \print@notesX@forpages and	
\print@Xnotes@forpages.	78
Correct \correct@footinsX@box and \correct@Xfootins@box	78
\Pages: Prevent false empty page after \Pages (bug added in 1.13.0)	104
v1.14.0.	
General: Fix a bug with line number position when using \eledsection and similar	
commands for RTL texts with Lua \TeX	1
The \newifs are not followed by boolean values set to false, because it is the \TeX	
default setting.	1
v1.15.0.	
General: Add \AtEveryPstartCall.	1
Add sameparallelpagenumber option.	13
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
\do@actions@nextR: Add actions 1008 and 1009	70
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting	
when a verse is hanging	86
\lineationR: As \lineation, \lineationR automatically set the	
\pstartinfootnote.	34
\n@num: \n@num defined only one time for both Eledmac and Eledpar.	44
\skipnumbering: \skipnumbering defined only one time for both Eledmac and	
Eledpar	49
v1.16.0.	
General: Error message when calling \Pages inside pages environment and \Columns	
inside pairs environment.	1
Error message when starting a Leftside/a Rightside while the previous one has not	
been yet typeset.	1
Error message when using \beginnumbering...\endnumbering without \pstart.	1
Fix a bug with nofamiliar / nocritical option of eledmac.	1
New package option sameparallelpagenumber to have the same page number for both	
left and right side.	1
\newseries@par: Fix a bug with \onlysideX.	50
v1.16.1.	
General: Write information about line-list file version in the correct file.	1
v1.16.2.	
General: Fix a bug when adding empty lines before a \pend in combination with some	
specific penalties setting.	1
v1.17.0.	
General: Add compatibility of optional argument of \pstart/\pend and	
\AtEveryPstart/\AtEveryPend with two columns mode.	1
v1.21.0.	
General: Add \hidenumbering	17
v2.0.0.	
General: \@nl is now defined only in reledmac.	42
\ifbypage@ and \ifbypstart@R defined in eledmac.	34
Fix some bugs with sameparallelpagenumber option.	1
Many code refactored and moved to reledmac.	1
Package's name becomes reledpar.	1
Totally new implementation of sameparallelpagenumber option.	1
\@adv: \@adv defined only in reledmac.	43
\@lab: \@lab defined only in eledmac.	84

<code>\@ref@regR</code> : <code>\@ref</code> defined only in <code>reledmac</code> , code specific to right side moved in <code>\ref@regR</code> .	44
<code>\@set</code> : <code>\@set</code> defined only in <code>reledmac</code> .	43
<code>\advanceLine</code> : <code>\advanceLine</code> defined only in <code>reledmac</code> .	49
<code>\bbl@set@language</code> : Patch <code>\bbl@set@language</code> instead of redefining it	89
<code>\do@lockonR</code> : <code>\do@lockon</code> defined only in <code>reledmac</code> .	43
<code>\endlock</code> : <code>\startlock</code> and <code>\endlock</code> defined only in <code>reledmac</code> .	49
<code>\endsub</code> : <code>\startsub</code> and <code>\endsub</code> defined only in <code>reledmac</code> .	49
<code>\fix@page</code> : <code>\fix@page</code> is defined only once in <code>reledmac</code>	43
<code>chapterinpages</code> : Deleting the old system of managing parallel chapter, keep only the new one with <code>\patchcmd</code> .	55
<code>\l@d@set</code> : <code>\l@d@set</code> defined only in <code>reledmac</code> .	43
<code>\line@marginR</code> : <code>\linenummargin</code> now defined only once time in <code>reledmac</code> .	35
<code>\page@action</code> : <code>\page@action</code> defined only in <code>reledmac</code> .	43
<code>\read@linelist</code> : <code>\read@linelist</code> is defined only once time in <code>\reledmac</code> .	41
<code>\set@line</code> : <code>\set@line</code> defined only in <code>reledmac</code> .	49
<code>\set@line@action</code> : <code>\set@line@action</code> defined only in <code>reledmac</code> .	43
<code>\setline</code> : <code>\setline</code> defined only in <code>reledmac</code> .	49
<code>\setlinenum</code> : <code>\setlinenum</code> defined only in <code>reledmac</code> .	49
<code>\skip@lockoff</code> : <code>\do@lockoff</code> defined only in <code>reledmac</code> .	44
<code>\sub@action</code> : <code>\sub@action</code> defined only in <code>reledmac</code> .	43
<code>\sublinenumincrement*</code> : <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> are now defined only in <code>reledmac</code> .	37
v2.1.0.	
General: Fix a bug when using <code>\eledsection</code> and related on right pages when page width is short.	1
Fix a bug when using <code>\pagenumbering</code> with <code>memoir</code> (bug added in v2.0.0).	1
Fix a bug with <code>\setparLedgroupnotespacing</code> with the <code>shiftedpstarts</code> option.	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code>	1
Options to custom empty right page before <code>\Pages</code> .	1
v2.2.0.	
General: <code>astanza</code> environment can take an optional argument, which will be the optional argument of <code>\pstart</code> started by this environment.	1
New tools to number stanza	1
v2.2.1.	
General: Fix a bug with optional argument of last left <code>\pend</code>	1
v2.3.0.	
General: Change some internal codes in order to provide compatibility with <code>LaTeX</code> release of october 2015	1
Fix a bug with title number in parallel columns	1
New line setting command suffixed by <code>R</code> to set only the right side.	1
<code>\Pages</code> : Fix a bug when calling <code>\Columns</code> after a <code>\Pages</code> (bug added in v1.13.0).	105
v2.4.0.	
General: New way of (not) synchronizing the parallel pages.	1
Option to switch to <code>\mainmatter</code> when calling <code>\Pages</code>	1
<code>\ledstrutR</code> : Deleted <code>\ledstrutL</code> and <code>\ledstrutR</code>	111
Fix a bug with dotted letter	111
v2.5.0.	
General: Disable empty lines as paragraph in <code>astanza</code> .	1
Fix a bug introduced in v1.15.0 which made hanging indentation in verse not work anymore.	1
New commands <code>\linenummarginR</code> and <code>\linenummargin*</code>	1

v2.5.1.	General: Fix spurious space when using optional argument of <code>astanza</code> environment (introduced in v2.5.0).	1
v2.5.2.	General: Fix a bug introduced in v2.5.0 with <code>\linenummargin</code> , <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code>	1
v2.6.0.	General: <code>\Xlineflag</code> and <code>\Xendlineflag</code> added	1
	<code>\printlinesR</code> deleted	1
	Error message when calling <code>\Pages</code> or <code>\Columns</code> without previous <code>pages</code> or <code>pairs</code> environment.	1
	Fix a bug with footnote numbering when using the same series of familiar footnotes on both sides.	1
	Fix a bug with right side title number when using title commands before <code>pages</code> or <code>columns</code> environments.	1
	Fix compatibility with <code>babel</code> (broken in v2.0.0).	1
	No error messages about ends of left / right page when using the <code>\syntonly</code> command of the <code>syntonly</code> package.	1
	<code>\l@dmake@labelsR: \@Rlineflag</code> is not stored directly after the line number, but as a fifth argument of <code>\the@labelX</code> . Can be retrieved by <code>\xflagref</code>	83
v2.6.1.	General: Fix a bug, introduced in v2.6.0, with footnote numbering when using <code>perpage</code> package.	1
v2.6.2.	General: Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.6.0).	1
	Fix a bug (added in v2.6.0) with footnote numbering in parallel typesetting while using <code>polyglossia</code> with specific numbering systems (like Greek).	1
	<code>\newseries@par</code> : The TeX counter <code>\footnoteX@reading</code> is defined in <code>reledmac</code>	50
v2.6.3.	General: Fix spurious dot when using <code>\linenummargin</code> on right side (introduced in v2.5.0).	1
v2.7.0.	General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package.	1
v2.7.1.	General: Fix a bug added in <code>reledmac</code> 2.8.2, when typesetting parallel text just after a sectioning command	1
v2.8.0.	General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> options.	1
	Add <code>\linenumberLevenifblanktrue</code> and <code>\linenumberRevenifblank</code> commands	1
	Fix a bug when the right line number style is not the same to the left line number style	1
v2.9.0.	General: Add <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code>	1
	More specific error messages.	1
v2.9.1.	General: Prevent <code>\Xtxtbeforenotes</code> hook from causing notes to go beyond the bottom margin	1
v2.10.0.	General: Add new tools to make apparatuses of manuscripts	1
	<code>\do@actions@nextR</code> : Add action 1010	70
v2.11.0.	General: Compatibility with <code>reledmac</code> 's <code>auxdir</code> option.	1

\correct@footinsX@box: Clarification in the handbook about the use of \Xonlyside and onlysideX and error message if mesuse.	78
New implementation of \Xonlyside and \onlysideX hooks, prevent trouble with vertical spacing.	78
v2.12.0.	
General: Add \edtextlater and \edtextnow.	1, 81
Fix a bug with \AtBeginPairs.	1
Fix a bug with \lineationR.	1
v2.13.0.	
General: Compatibility with the new features of reledmac 2.15.0	1
v2.13.1.	
General: Simplification of the code concerning the sectioning command	1
v2.14.0.	
General: Add \linenumberlistR, equivalent to \linenumberlist for the right side.	1
Error message when left / right side are not defined.	1
More explicit error message when the stanza indentation is not defined.	1
New commands to have specific line margins for text running in parallel columns	1
When indexing texts in sidenotes with \edtext, referring to the line number where the sidenote is called.	1
v2.14.1.	
General: Fix a bug when using \markboth on \doinsidelineLhook and \doinsidelineRhook.	1
Fix a bug with vertical space before sectioning command in optional argument of \pstart (bug added in v2.13.1).	1
v2.14.2.	
General: Fix spurious space with \edtextlater.	1
v2.14.3.	
General: Fix a bug in \edtextnow.	1
v2.14.4.	
General: Take into account \linenumberstyle when using \edlineref.	1
v2.15.0.	
General: Add movecolumnsonrightpage option.	1
Fix a bug about space before sectioning commands after a pairs environment (bug added in v. 2.14.4).	1
Fix a bug with \msdata when using multiple \beginnumbering... \endnumbering	1
Fix a bug with page break when using \columns (bug added in v. 2.14.1).	1
New macro \sidenotemarginR, fix default value to right.	1
v2.16.0.	
General: Compatibility with \setmsdataposition	1
v2.16.1.	
General: Restore sidenotes on left side, deleted by mistake in v2.16.0	1
v2.16.2.	
General: Change log message when numbered files still don't exist, in order to improve compatibility with <i>latexmk</i>	1
v2.16.3.	
General: Fix a bug when setting \aftercolumnseparator.	1
v2.17.0.	
General: Change code for compatibility with reledmac 2.18.0	1
v2.17.1.	
General: Fix a bug with prevpgstyle option when not using prevpgnotnumbered.	1
v2.17.2.	
\Pages: Fix a bug when changing \Lcolwidth and \Rcolwidth in pages environment.	105

v2.17.3.	
General: Fix a bug when using multiple sidenotes on the same line, on right side.	1
v2.17.4.	
General: Fix bugs with sublines in parallel typesetting.	1
v2.18.0.	
General: Compatibility with v2.20.0 of <code>reledmac</code>	1
v2.19.0.	
General: Compatibility with v2.22.0 of <code>reledmac</code>	1
v2.19.1.	
General: Fix a bug with <code>\edtext</code> in sectioning command with <code>shiftedpstarts</code> mode.	1
v2.19.2.	
General: <code>continuousnumberingwithcolumns</code> also works for the <code>pstart</code> number.	1
v2.19.3.	
General: Do not print footnotes at the first run.	1
Fix a bug with <code>nopbinverse</code> when using consecutive <code>\Pages</code>	1
v2.19.4.	
General: Optimisation of the code in order to not print footnotes at the first run.	1
v2.19.5.	
General: Fix a bug added in <code>astanza</code> in v. 2.18.0	1
v2.20.0.	
General: Add <code>\linenumOnlyPagesForColumns</code> and <code>\linenumOnlyPagesForColumnsR</code> to tell to print the line numbers associated with a column only on some specific pages (for example, only on right pages).	1
Add <code>sidenotesmarginpage</code> option to make the sidenotes refer to page margins and not column margins.	1
Fix a bug with <code>\continuousnumberingwithcolumns</code> when using multiple series of continued numbering text. Now, the line number are correctly reset at each <code>\beginnumbering</code>	1
Fix a bug with <code>continuousnumberingwithcolumns</code> and inner / outer notes and line numbering.	1
Fix a bug with <code>continuousnumberingwithcolumns</code> when a <code>\beginnumbering</code> is called after another <code>\beginnumbering... \endnumbering</code>	1
Fix a bug with <code>widthliketwocolumns</code> when manually setting <code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code>	1
Fix compatibility between <code>continuousnumberingwithcolumns</code> and <code>\numberpstartfalse</code>	1
Fix incompatibility between <code>\lineation{page}</code> and <code>continuousnumberingwithcolumns</code> option	1
<code>\next@line@list@stuffR</code> : Add <code>\next@line@list@stuffR</code> internal hook.	48
v2.20.1.	
General: Fix bug with <code>continuousnumberingwithcolumns</code> when left column and right column have not the same size.	1
v2.20.2.	
General: Fix bugs with <code>continuousnumberingwithcolumns</code> when going to a new page	1
v2.21.0.	
General: Add compatibility with new features of <code>reledmac</code> 2.26.0	1
v2.21.1.	
General: Fix bug with <code>\linenumberannotation</code>	1
v2.21.2.	
General: Fix bug with <code>\Xpstart</code> and <code>\edtextlater</code>	1
v2.21.3.	
General: Fix bug when using <code>\numberpstarttrue</code> and alternating between manual <code>\pstart</code> and automatic <code>\pstart</code> produced by <code>\autopar</code>	1

v2.21.4.	General: Fix bug with using <code>\eledsection</code> and related when alternate normal typesetting and parallel typesetting	1
v2.21.5.	General: Compatibility with <code>reledmac</code> v. 2.26.6	1
v2.21.6.	General: Fix page number in footnote and endnotes when using <code>sameparallelpagenumber</code> option	1
v2.21.7.	General: Compatibility with <code>reledmac</code> 2.26.7	1
v2.21.8.	General: Fix bug with critical footnotes line number when alternating between <code>\numberlinefalse</code> and <code>\numberlinetrue</code>	1
	Fix bug with marginal line number in parallel typesetting when alternating between <code>\numberlinefalse</code> and <code>\numberlinetrue</code>	1
	Workaround with a bug of <code>TeX2018/04/01</code> patch level 3 to generate the <code>.sty</code> file from a <code>.ins</code> file	1
v2.21.9.	General: Fix (little) bug when using <code>calc</code> package.	1
v2.22.0.	General: Compatibility with <code>\doinsidethislinehook</code> macro.	1
v2.22.1.	General: Really reset line number annotation at each line	1
v2.22.2.	General: Fix bug with <code>\Xonlyside</code> and <code>\onlysideX</code>	1
v2.22.3.	General: When using <code>\parindentX</code> or <code>\Xparindent</code> and <code>stanza</code> , use the normal indentation in the footnote, not the verse indentation.	1
v2.22.4.	<code>\theledlanguageR</code> : Provide compatibility with the latest version of <code>polyglossia</code> (v1.45).	90
v2.23.0.	General: Add <code>\Leftpagehook</code> and <code>\Rightpagehook</code>	1
	Add hanging symbol feature to <code>astanza</code>	1
	<code>\theledlanguageR</code> : Restore compatibility with versions of <code>polyglossia</code> prior to v1.45.	90
v2.23.1.	General: Fix bug when using <code>polyglossia</code> v1.45 or later.	1
v2.24.0.	General: Add <code>antilabe</code> feature.	1
	Add <code>startstanzaindentsrepetition</code> feature.	1
v2.24.1.	General: Fix incompatibility between lineation by <code>\pstart</code> and <code>(a)stanza</code>	1
	Makes <code>\numberstantzatrue</code> working with <code>astanza</code>	1
v2.24.2.	General: Fix some bugs with lineation by page, when a numbered section starts at the very beginning of page.	1
v2.24.3.	General: Fix bug with hanging verse in parallel typesetting.	1
v2.25.0.	General: Add compatibility with <code>\linenumannotationothersidettrue</code> of <code>reledmac</code>	1
v2.25.1.	General: <code>\ifsublines@R</code> migrated to <code>reledmac</code>	1
v2.25.2.	General: Fix incompatibility with <code>babel</code> v3.61	1

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