

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	6
2 Options	6
2.1 Synchronization’s options	6
2.2 Other options	6
3 General	7

*This file (`reledpar.dtx`) has version number v2.16.3, last revised 2017/02/10.

[†]maieul at maieul dot net

[‡]herries dot press at earthlink dot net

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

<i>Contents</i>	3
12 Notes about page number	21
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 option	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	39
IV.7.1 Correspond to those in <code>reledmac</code> for regular or left text	39
IV.7.2 Specific to <code>reledpar</code>	40
IV.8 Reading the line-list file	40
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	49
V.1.1 Notes to be printed on one side only	49
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)	52
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, \pstart and \pend	57
VIII.2 Processing one line	62
VIII.3 Line and page number computation	67
VIII.4 Line number printing	70
VIII.5 Pstart number printing in side	72
VIII.6 Add insertions to the vertical list	74
VIII.7 Penalties	75
VIII.8 Printing leftover notes	76
IX Footnotes	76
IX.1 Footnotes output specific to \Pages	76
IX.2 Critical footnote printed in right side	79
X Cross referencing	81
XI Side notes	82
XII Verse	84
XIII Fixing babel and polyglossia	86
XIV Counts and boxes for parallel texts	88
XV Checking text to be processed	90
XVI Parallel columns	91
XVII Parallel pages	100
XVII.1 Specific counters	101
XVII.2 Main macro	101
XVII.3 Ensure all notes are printed at the end of parallel pages	107
XVII.4 Struts	108
XVII.5 Page clearing	108
XVII.6 Lines managing	109
XVII.7 Page break managing	111
XVII.8 Getting boxes content	114
XVIII Page numbering	118
XVIII.1 Global options	118
XVIII.2 mainmatter option of \Pages	120
XIX Sections' titles' commands	120
XX Page break/no page break, depending on the specific line	121
XXI Parallel ledgroup	122

XXII Compatibility with eledmac	126
XXIII The End	126
Appendix A Some things to do when changing version	127
Appendix A.1 Migration to eledpar 1.4.3	127
Appendix A.2 Migration from eledpar to reledpar	127
Appendix A.2.1 Deprecated options	127
Appendix A.2.2 \renewcommandreplaced with command	127
Appendix A.2.3 Commands the names of which have changed	128
Appendix A.3 Migration to reledpar 2.2.0	128
Appendix A.4 Migration to reledpar 2.3.0	128
Appendix A.5 Migration to reledpar 2.4.0	128
Appendix A.6 Migration to reledpar 2.5.0	128
Appendix A.7 Migration to reledpar 2.6.0	128
Appendix A.8 Migration to reledpar 2.6.1	128
Appendix A.9 Migration to reledpar 2.14.0	129
Appendix A.10 Migration to reledpar 2.14.1	129
References	129
Index	129
Change History	149

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. 10 to understand better those options.

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

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

nomaxlines allows facing pages to have different numbers of lines.

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

2.2 Other options

parledgroup allows the use of `ledgroup` environment with `reledpar`.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

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

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

²This option can either be used on `reledmac` or `reledpar`.

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

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

3 General

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

The `reledpar` package lets you typeset two *numbered* texts in 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 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.

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

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

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

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

4 Parallel columns

4.1 Basic use

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

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

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

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

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

There is no required pagebreak before or after the columns.

4.2 Setting

4.2.1 Column’s width

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

`\Rcolwidth`


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

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

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

4.2.2 Column’s separator

`\columnrulewidth`
`\columnseparator`

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

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

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

4.2.3 Column’s positions

`\columnspanposition`

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

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

`\beforecolumnseparator`
`\aftercolumnseparator`

By default, the spaces around column separator are the same as the space:

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

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

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

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

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

4.2.4 Mixing two columns and one column texts

`\widthliketwocolumns`

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

`\Xnoteswidthliketwocolumns`
`\notesXwidthliketwocolumns`

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

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

5 Keeping translation in the outside/inside column

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

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

6 Facing pages

6.1 Basic usage

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

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

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

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

6.2 Setting

6.2.1 Text width

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

`\Rcolwidth`

6.2.2 Way of 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.

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

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:

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

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

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

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

1. Regarding the number of lines a pair of chunks may have:

- (a) ‘`shiftedpstarts`’ setting merely moves any added blank lines from the bottom of the chunks to the bottom of the page. It does not allow to have more lines on a given page as it just removes the blank lines between the chunks and does nothing more. To understand better how this work, you may compare the total amounts of lines of text on a given page whether you have activated this setting or not: you will see that both amounts are the same.
- (b) ‘`advancedshiftedpstarts`’ prevents any blank lines from being inserted at the bottom of the chunks, also taking them away from the total amount of lines the page may have. This allows to get more lines on the pages. However, please note that:
 - Blank lines are taken into account as `reledpar` moves from one to the following chunk of text, so that every pair of chunks may always start on the same facing pages.

- Consequently, blank lines continue to be taken into account in the calculation of the amount of lines a given pair of pages may have. This is why when a longer chunk runs from one page to another the shorter corresponding one also runs across pages, even if this may result in some blank vertical space being left on the first page.
2. As regards the number of lines per page, including blank ones, the `nomaxlines` setting disregards the rule that forces two facing pages to have the same numbers of lines. So it allows to have more text on the pages. Then, by a complex mechanism it is ensured that two corresponding chunks may always start on the same facing pages, provided that `shiftedpstarts` or `advancedshiftedpstarts` settings shall not be activated.

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

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

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. 21.

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. 21.

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 L^AT_EX's feature.

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

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

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

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

numbering at the first run: 1 (left page), 3 (right page), 2 (left page), 4 (right page). But at the next run, you will obtain: 1 (left page), 2 (right page), 3 (left page), 4 (right page).

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

6.3.3 Using perpage package

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

So do not set:

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

But set:

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

6.3.4 Notes for one side only

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

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

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

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

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 (7.3 p. 17) to right critical footnotes and
`\Xendlineflag` `\Xendlineflag[⟨s⟩]` to add it to right critical endnotes.

7 Left and right texts

7.1 Environments

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

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

7.2 Numbering text lines and paragraphs

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

`\endnumbering`, like:

```
\beginnumbering
<text>
\endnumbering
```

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

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

`\memorydump`

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

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

`\numberstarttrue`
`\numberstartfalse`
`\thepstartL`
`\thepstartR`
`\skipnumbering`

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

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

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

7.3 Lineation system

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

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

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

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

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

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

`\firstsublinenumR` reledmac allows you to define a `\linenumberlist` to explicitly define (5.3.2 p. 19) in which line the line number will be printed. In parallel typesetting, this command affects only left side. For right side, you have to define `\linenumberlistR`.

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

`\sublinenumberstyle*` `\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.

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

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.

`\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.

By default, when a blank line is printed on one side, in order to synchronize with the other side, no line number is printed. However, you can decide to print them for

`\linenumberLevenifblanktrue`
`\linenumberRevenifblanktrue`

blank lines, also. Use `\linenumberLevenifblanktrue` to enable it on the left side, and `\linenumberRevenifblanktrue` to enable it on right side.

7.4 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. 17).

7.5 `\AtEveryPstart` and `\AtEveryPstartCall`

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

Consequently:

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

7.6 Language setting

If you are using the `babel` package or the `polyglossia` package, with different languages (via, say, `\selectlanguage`) for the left and right texts it is particularly important to select the appropriate language within the `Leftside` and `Rightside` environments. The initial language selected for the right text is the `babel` package's default.

Also, it is the *last* language setting in a side that controls the language used in any notes for that side when they get printed. If you are using multilingual notes then it is probably safest to explicitly specify the language(s) for each note rather than relying on the language selection for the side. The right side language is also applied to the right side line numbers.

7.7 Executing code at each line

`\dolineLhook` `\dolineRhook` `\doinsidelineLhook` `\doinsidelineRhook`

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

`\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 (5.5 p. 22).

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 \szi@00' it is because you have forgotten to use `\setstanzaindent` to set the stanza indents.

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

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

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

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

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

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

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

`\thestanzaL` When using `\stanzanumtrue` (9.9 p. 50) in parallel typesetting, stanza counter is
`\thestanzaR` 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`, `\ledouterote`, `\ledinnerrote`.

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.

10 Parallel ledgroups

10.1 General

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

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

See the following example:

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

```

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

```

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. 12 or `prevpgnotnumbered` option (6.2.5 p. 12), please read the following paragraph if you want to manipulate page numbers manually.

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

1. If you modify `\thepage` command, use the value of `par@page` counter inside and not the value of `page` counter.

2. If you want to modify a page number, modify the value of page counter AND the value `par@page counter`.

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

I Implementation overview

\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}[2017/02/10 v2.16.3 reledmac extension for
  parallel texts]%
4
5 %
```

II.2 Package’s requirement

Few commands use `\xspace` command.

```

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

II.3 Package’s options

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

```

8 \RequirePackage{xkeyval}
9 %
```

II.4 Package's options

II.4.1 Synchronization's options

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

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

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

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

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

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

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

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

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


```

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

```

II.4.2 Position option

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

```

39 \newif\ifmovecolumnpositiononrightpage%
40 \DeclareOptionX{movecolumnpositiononrightpage}{%
41   \movecolumnpositiononrightpagetrue%
42 }%
43 %

```

II.4.3 Other options

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

```

44 \DeclareOptionX{parledgroup}{\parledgrouptrue}
45 %

```

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

```

46 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
47 \DeclareOptionX{continuousnumberingwithcolumns}{\
48   continuousnumberingwithcolumnstrue}%
49 %

```

`\ifsameparallelpagenumber` Options related to page numbering

```

\ifprevpgnotnumbered49 \newif\ifsameparallelpagenumber
50 \newif\ifprevpgnotnumbered
51 \DeclareOptionX{sameparallelpagenumber}{\sameparallelpagenumbertrue}
52 \DeclareOptionX{prevpgnotnumbered}{\prevpgnotnumberedtrue}
53 %

```

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

```

54 \DeclareOptionX{prevpgstyle}{\gdef\prevpgstyle{#1}}%
55 %

```

```

56 \ProcessOptionsX%
57 %

```

II.5 Determining side and category of parallel processing

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

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

II.6 Text’s width

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

```
\Rcolwidth
58 \newdimen\Lcolwidth
59 \Lcolwidth=0.45\textwidth
60 \newdimen\Rcolwidth
61 \Rcolwidth=0.45\textwidth
62 %
```

II.7 Messages

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

```
\reledpar@error 63 \newcommand{\reledpar@error}[2]{\PackageError{reledpar}{#1}{#2}}
64 %
```

```
\reledpar@warning 65 \newcommand{\reledpar@warning}[1]{\PackageWarning{reledpar}{#1}}%
66 %
```

```
\led@err@TooManyPstarts 67 \newcommand*{\led@err@TooManyPstarts}{%
68 \reledpar@error{Too many \string\pstart\space without printing.
69 Some text will be lost}{\@ehc}}
70 %
```

```
\led@err@BadLeftRightPstarts 71 \newcommand*{\led@err@BadLeftRightPstarts}[2]{%
72 \reledpar@error{The numbers of left (#1) and right (#2)
73 \string\pstart s do not match}{\@ehc}}
74 %
```

```

\led@err@LeftOnRightPage75 \providebool{syntax@}
\led@err@RightOnLeftPage76 \newcommand*{\led@err@LeftOnRightPage}{%
77   \notbool{syntax@}%
78   {\reledpar@error{The left page has ended on a right page}{\@ehc}}%
79   {}%
80 }
81 \newcommand*{\led@err@RightOnLeftPage}{%
82   \notbool{syntax@}%
83   {\reledpar@error{The right page has ended on a left page}{\@ehc}}
84   {}%
85 }%
86 %

\led@err@Leftside@PreviousNotPrinted77 \newcommand*{\led@err@Leftside@PreviousNotPrinted}{%
\led@err@Rightside@PreviousNotPrinted78 \reledpar@error{You call a new Leftside environment while the previous
one has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}
89 \newcommand*{\led@err@Rightside@PreviousNotPrinted}{%
90 \reledpar@error{You call a new Rightside environment while the previous
one has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}
91 %

\led@err@Pages@InsideEnv92 \newcommand*{\led@err@Pages@InsideEnv}{%
\led@err@Columns@InsideEnv93 \reledpar@error{\string\Pages\space must be called *outside* of the `
pages` environment}{\@ehc}}
94 \newcommand*{\led@err@Columns@InsideEnv}{%
95 \reledpar@error{\string\Columns\space must be called *outside* of the `
pairs` environment}{\@ehc}}
96 %

\led@err@Pages@WithoutEnv97 \newcommand*{\led@err@Pages@WithoutEnv}{%
\led@err@Columns@WithoutEnv98 \reledpar@error{\string\Pages\space called without previous `pages`
environment}{\@ehc}}
99 \newcommand*{\led@err@Columns@WithoutEnv}{%
100 \reledpar@error{\string\Columns\space called without previous `pairs`
environment}{\@ehc}}
101 %

\led@err@Pages@WithoutRightside92 \newcommand*{\led@err@Pages@WithoutRightside}{%
\led@err@Pages@WithoutLeftside93 \reledpar@error{\string\Pages\space called without any preceding `
Rightside` environment (or environment without content)}{\@ehc}}%
\led@err@Columns@WithoutRightside94 \newcommand*{\led@err@Pages@WithoutLeftside}{%
105 \reledpar@error{\string\Pages\space called without any preceding `
Leftside` environment (or environment without content)}{\@ehc}}%
106 \newcommand*{\led@err@Columns@WithoutRightside}{%
107 \reledpar@error{\string\Columns\space called without any preceding `
Rightside` environment (or environment without content)}{\@ehc}}%

```

```

108 \newcommand*\led@err@Columns@WithoutLeftside}{%
109   \reledpar@error{\string\Columns\space called without any preceding `
Leftside` environment (or environment without content)}{\@ehc}}%
110 %

```

```

\led@error@fail@patch@thepage11 \newcommand{\led@error@fail@patch@thepage}{%
112   \reledpar@error{Fail to patch \string\@thepage\space command.}{\@ehc}%
113 }%
114 %

```

```

\led@error@fail@patch@pagenumbering15 \newcommand{\led@error@fail@patch@pagenumbering}{%
116   \reledpar@error{Fail to patch \string\pagenumbering\space command.}{\@ehc
}%
117 }%
118 %

```

```

\led@error@note@called@onrightside19 \newcommand{\led@error@note@called@onrightside}[1]{%
\led@error@note@called@onleftside20   \reledpar@error{#1 called on right side, despite your configuring it to
be for the left side only}{\@ehc}%
121 }%
122 \newcommand{\led@error@note@called@onleftside}[1]{%
123   \reledpar@error{#1 called on left side, despite your configuring it to be
for the right side only}{\@ehc}%
124 }%
125 %

```

```

\led@error@fail@patch@mempnum26 \newcommand{\led@error@fail@patch@mempnum}{%
127   \reledpar@error{Fail to patch \string@mempnum\space command.}{\@ehc}%
128 }%
129 %

```

```

\led@error@fail@patch@outputpage30 \newcommand{\led@error@fail@patch@outputpage}{%
131   \reledpar@error{Fail to patch \string\@outputpage\space command.}{\@ehc}%
132 }%
133 %

```

```

\led@error@edtext@later@now34 \newcommand{\led@error@edtext@later@now}{%
135   \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?)
136 }%
137 %

```

```

\led@warn@ChangeSyncOption38 \newcommand*{\led@warn@ChangeSyncOption}[1]{%
139 \reledpar@warning{You have changed synchronization's options since the
last run. We have not read line-list file #1. Please run LaTeX again.}%
140 }%
141 %

\warn@setting@in@rightside42 \newcommand{\led@warn@setting@in@rightside}[1]{%
143 \reledpar@warning{You use #1 inside rightside environment.\MessageBreak%
144 Such behavior is deprecated.\MessageBreak%
145 Use instead #1R or #1* in your preamble.}%
146 }
147 %

\led@error@missing@numbering48 \newcommand{\led@error@missing@numbering}[1]{%
149 \reledpar@error{Missing \string\...pstart\string\pend\space inside `#1`
environment}{\@ehc}%
150 }%
151 %

```

II.8 Naming macros

The \LaTeX kernel provides $\backslash\@namedef$ and $\backslash\@namuse$ 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.

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

$\backslash\text{setnamebox}$

$\backslash\text{unhnamebox}$

$\backslash\text{unvnamebox}$

$\backslash\text{namebox}$

```

152 \providecommand*{\newnamebox}[1]{%
153 \expandafter\newbox\csname #1\endcsname}
154 \providecommand*{\setnamebox}[1]{%
155 \expandafter\setbox\csname #1\endcsname}
156 \providecommand*{\unhnamebox}[1]{%
157 \expandafter\unhbox\csname #1\endcsname}
158 \providecommand*{\unvnamebox}[1]{%
159 \expandafter\unvbox\csname #1\endcsname}
160 \providecommand*{\namebox}[1]{%
161 \csname #1\endcsname}
162 %
163 %

```

$\backslash\text{ifcsboxvoid}$ The $\backslash\text{ifcsboxvoid}$ command comes on top of the `etoolbox` package. The first argument is a box name, the second is the $\langle true \rangle$ value, the third the $\langle false \rangle$ value. If the first argument refers to a non-existing box, or if it refers to a box that is void, the $\langle true \rangle$ value is returned. otherwise, $\langle false \rangle$ is returned.

```

164 \newcommand*{\ifcsboxvoid}[1]{%
165   \ifcsname#1\endcsname%
166   \expandafter\ifvoid\csname #1\endcsname%
167   \expandafter\expandafter%
168   \expandafter\@firstoftwo%
169   \else%
170   \expandafter\expandafter%
171   \expandafter\@secondoftwo%
172   \fi%
173   \else%
174   \expandafter\@firstoftwo%
175   \fi%
176 }%
177 %

```

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

```

\usenamecount
178 \providecommand*\newnamecount}[1]{%
179   \expandafter\newcount\csname #1\endcsname}
180 \providecommand*\usenamecount}[1]{%
181   \csname #1\endcsname}
182
183 %

```

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.

```

184 \newcount\section@numR
185 \section@numR=\z@
186 %

```

\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`.

```

187 \pst@rtedLfalse
188 \newif\ifpst@rtedR
189
190 %

```

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

```

191 \newcommand*{\beginnumberingR}{%
192   \ifnumberingR
193     \led@err@NumberingStarted
194   \endnumberingR
195 \fi
196 \global\l@dnumpstartsR \z@
197 \global\pst@rtedRfalse
198 \global\numberingRtrue
199 \global\advance\section@numR \@ne
200 \global\absline@numR \z@
201 \gdef\normal@page@breakR{}
202 \gdef\l@prev@pbR{}
203 \gdef\l@prev@nopbR{}
204 \global\line@numR \z@
205 \global\@lockR \z@
206 \global\sub@lockR \z@
207 \global\sublines@false
208 \global\let\next@page@numR\relax
209 \global\let\sub@change\relax
210 \global\stopmsdata@inserted@true%
211 \global\let\@msdata@list\relax%
212 \global\csundef{\@msdata@\add@msd@cR @dataR}%
213 \set@continuousnumberingforR%
214 \message{Section \the\section@numR R }%
215 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
216 \l@dend@stuff
217 \setcounter{pstartR}{1}
218 \begingroup
219 \initnumbering@sectcountR
220 \gdef\eled@sectionsR@{}%
221 \if@noeled@sec\else%
222   \makeatletter%
223   \InputIfFileExists%
224     {\l@auxdir\jobname.eledsec\the\section@numR R}%
225     {}%
226     {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@numR R}}%
227   \makeatother%
228   \immediate\openout\eled@sectioningR@out=\l@auxdir\jobname.eledsec\the\
section@numR R\relax%
229 \fi%
230 }
231 %

```

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

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

```

232 \def\endnumberingR{%
233   \ifnumberingR
234     \global\numberingRfalse
235     \normal@pars
236     \ifnum\l@dnumpstartsR=0%
237       \led@err@NumberingWithoutPstart%
238     \fi%
239     \ifl@dpairing
240       \global\pst@rtedRfalse
241     \else
242       \ifx\insertlines@listR\empty\else
243         \global\noteschanged@true
244       \fi
245       \ifx\line@listR\empty\else
246         \global\noteschanged@true
247       \fi
248     \fi
249     \ifnoteschanged@
250       \led@mess@NotesChanged
251     \fi
252   \else
253     \led@err@NumberingNotStarted
254   \fi
255   \endgroup
256   \if@noeled@sec\else%
257     \immediate\closeout\eled@sectioningR@out%
258   \fi%
259 }
260
261 %

```

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

`\save@section@number`

`\set@sectcountR`

```

262 \newcounter{chapterR}
263 \newcounter{sectionR}
264 \newcounter{subsectionR}
265 \newcounter{subsubsectionR}
266
267 \newcount\old@chapter%
268 \newcount\old@section%
269 \newcount\old@subsection%
270 \newcount\old@subsubsection%
271 \newcommand{\save@section@number}{%
272   \ifdefined\c@chapter%
273     \global\old@chapter\value{chapter}%
274   \fi%
275   \global\old@section\value{section}%
276   \global\old@subsection\value{subsection}%

```



```

277 \global\old@subsubsection\value{subsubsection}%
278 }%
279 \newcommand{\initnumbering@sectcountR}{
280 \ifdefined\c@chapter%
281 \setcounter{chapterR}{\old@chapter}%
282 \fi%
283 \setcounter{sectionR}{\old@section}%
284 \setcounter{subsectionR}{\old@subsection}%
285 \setcounter{subsubsectionR}{\old@subsubsection}%
286 \set@sectcountR%
287 }
288 \newcommand{\set@sectcountR}{%
289 \let\c@chapter\c@chapterR%
290 \let\c@section\c@sectionR%
291 \let\c@subsection\c@subsectionR%
292 \let\c@subsubsection\c@subsubsectionR%
293 }%
294 %

```

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

```

295 \newcommand*{\pausenumberingR}{%
296 \endnumberingR\global\numberingRtrue}
297 \newcommand*{\resumenumberingR}{%
298 \ifnumberingR
299 \global\pst@rtedRtrue
300 \global\advance\section@numR \@ne
301 \led@mess@SectionContinued{\the\section@numR R}%
302 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
303 \ld@end@stuff
304 \begingroup%
305 \initnumbering@sectcountR%
306 \set@continuousnumberingforR%
307 \else
308 \led@err@numberingShouldHaveStarted
309 \endnumberingR
310 \beginnumberingR
311 \fi}
312 %
313 %

```

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

```

314 \newcommand*{\memorydumpL}{%
315 \endnumbering
316 \numberingtrue
317 \global\pst@rtedLtrue
318 \global\advance\section@num \@ne
319 \led@mess@SectionContinued{\the\section@num}%

```

```

320 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
321 \l@dend@stuff}
322
323 \newcommand*{\memorydumpR}{%
324 \endnumberingR
325 \numberingRtrue
326 \global\pst@rtedRtrue
327 \global\advance\section@numR \@ne
328 \led@mess@SectionContinued{\the\section@numR R}%
329 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
330 \l@dend@stuff}
331
332 %

```

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.

```

333 \newcommand*{\lineationR}[1]{%
334 \ifnumbering
335 \led@err@LineationInNumbered
336 \else
337 \def\@tempa{#1}\def\@tempb{page}%
338 \ifx\@tempa\@tempb
339 \global\bypage@Rtrue
340 \global\bypstart@Rfalse
341 \unless\ifnocritical@%
342 \Xpstart[] [false]%
343 \fi%
344 \else
345 \def\@tempb{pstart}%
346 \ifx\@tempa\@tempb
347 \global\bypage@Rfalse
348 \global\bypstart@Rtrue
349 \unless\ifnocritical@%
350 \Xpstart%
351 \fi%
352 \else
353 \def\@tempb{section}
354 \ifx\@tempa\@tempb
355 \global\bypage@Rfalse%

```

```

356         \global\bystart@Rfalse%
357         \unless\ifnocritical@%
358             \Xpstart[] [false]%
359         \fi%
360     \else
361         \led@warn@BadLineation
362     \fi%
363 \fi
364 \fi
365 \fi}}
366 %

```

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

```

367 \newcommand{\set@continuousnumberingforR}{%
368     \ifcontinuousnumberingwithcolumns%
369         \ifnum\line@numR<\line@num%
370             \expandafter\setlinenum\expandafter{\the\line@num}%
371         \fi%
372         \ifnum\last@page@num>\last@page@numR%
373             \global\last@page@numR=\last@page@num%
374         \fi%
375     \fi%
376 }
377 %

```

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

```

378 \WithSuffix\newcommand\lineation*[1]{%
379     \lineation{#1}%
380     \lineationR{#1}%
381 }%
382 %

```

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`.

```

383 \newcount\line@marginR
384 %

```

By default put right text numbers at the right.

```

385 \line@marginR=\@ne
386
387 %

```

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

```

388 \newcommand{\linenummarginR}[1]{%
389   \l@getline@margin{#1}%
390   \ifnum\@l@dttempcntb>\m@ne%
391     \global\line@marginR=\@l@dttempcntb%
392   \fi%
393 }
394 \WithSuffix\newcommand\linenummargin*[1]{%
395   \l@getline@margin{#1}%
396   \ifnum\@l@dttempcntb>\m@ne%
397     \global\line@marginR=\@l@dttempcntb%
398     \global\line@margin=\@l@dttempcntb%
399   \fi%
400 }
401 \ifmovecolumnspostiononrightpage%
402   \linenummargin{inner}%
403   \linenummarginR{outer}%
404 \fi%
405 %

```

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

```

406
407 \newcount\line@margin@columnsR%
408 \line@margin@columnsR=\m@ne%
409
410 \newcommand{\linenummarginColumns}[1]{%
411   \l@getline@margin{#1}%
412   \ifnum\@l@dttempcntb>\m@ne%
413     \global\line@margin@columns=\@l@dttempcntb%
414   \fi%
415 }%
416
417 \WithSuffix\newcommand\linenummarginColumns*[1]{%
418   \l@getline@margin{#1}%
419   \ifnum\@l@dttempcntb>\m@ne%
420     \global\line@margin@columns=\@l@dttempcntb%
421     \global\line@margin@columnsR=\@l@dttempcntb%
422   \fi%

```

```

423 }%
424
425 \newcommand{\linenummarginColumnsR}[1]{%
426   \l@getline@margin{#1}%
427   \ifnum\@l@tempcntb>\m@ne%
428     \global\line@margin@columnsR=\@l@tempcntb%
429   \fi%
430 }%
431
432 %

```

IV.3 Setting lineation start and step

`\c@firstlinenumR`
`\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.

```

433 \newcounter{firstlinenumR}
434   \setcounter{firstlinenumR}{5}
435 \newcounter{linenumincrementR}
436   \setcounter{linenumincrementR}{5}
437 %

```

`\c@firstsublinenumR`
`\c@sublinenumincrementR`

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

```

438 \newcounter{firstsublinenumR}
439   \setcounter{firstsublinenumR}{5}
440 \newcounter{sublinenumincrementR}
441   \setcounter{sublinenumincrementR}{5}
442
443 %

```

`\firstlinenum`
`\linenumincrement`
`\firstsublinenum`
`\sublinenumincrement`

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

```

444 \WithSuffix\newcommand\firstlinenum*[1]{%
445   \setcounter{firstlinenumR}{#1}%
446   \setcounter{firstlinenum}{#1}%
447 }
448 \firstlinenum*
449 \WithSuffix\newcommand\linenumincrement*[1]{%
450   \setcounter{linenumincrementR}{#1}%
451   \setcounter{linenumincrement}{#1}%
452 }
453 \linenumincrement*
454 \WithSuffix\newcommand\firstsublinenum*[1]{%
455   \setcounter{firstsublinenumR}{#1}%
456   \setcounter{firstsublinenum}{#1}%
457 }
458 \firstsublinenum*
459 \sublinenumincrement*
460 \WithSuffix\newcommand\sublinenumincrement*[1]{%
461   \setcounter{sublinenumincrementR}{#1}%
462   \setcounter{sublinenumincrement}{#1}%
463 }
464 \sublinenumincrement*

```

```

455 }
456 \WithSuffix\newcommand\sublinenumincrement*[1]{%
457   \setcounter{sublinenumincrementR}{#1}%
458   \setcounter{sublinenumincrement}{#1}%
459 }
460 %

```

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

```

\linenumincrementR
\firstsublinenumR
\sublinenumincrementR
461 \newcommand\firstlinenumR[1]{%
462   \setcounter{firstlinenumR}{#1}%
463 }
464 \newcommand\linenumincrementR[1]{%
465   \setcounter{linenumincrementR}{#1}%
466 }
467 \newcommand\firstsublinenumR[1]{%
468   \setcounter{subfirstlinenumR}{#1}%
469 }
470 \newcommand\sublinenumincrementR[1]{%
471   \setcounter{sublinenumincrementR}{#1}%
472 }
473 %

```

IV.4 Setting line flag

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

```

474 \newcommand{\setRlineflag}[1]{%
475   \gdef\Rlineflag{#1}%
476 }
477 \setRlineflag{R}
478 %

```

IV.5 Setting line number style

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

```

479 \newcommand*\linenumrepR[1]{\@arabic{#1}}
480 \newcommand*\sublinenumrepR[1]{\@arabic{#1}}
481
482 %

```

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

`\sublinenumnumberstyleR`

```

483 \newcommand*\linenumnumberstyleR[1]{%
484   \def\linenumrepR##1{\@nameuse{@#1}{##1}}
485 \newcommand*\sublinenumnumberstyleR[1]{%
486   \def\sublinenumrepR##1{\@nameuse{@#1}{##1}}
487 %

```

```

\linenumberstyle* And for both side.
\sublinenumberstyle*
488 \WithSuffix\newcommand\linenumberstyle*[1]{%
489   \linenumberstyle{#1}%
490   \linenumberstyleR{#1}%
491 }%
492
493 \WithSuffix\newcommand\sublinenumberstyle*[1]{%
494   \sublinenumberstyle{#1}%
495   \sublinenumberstyleR{#1}%
496 }%
497 %
498 %

```

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.

```

499 \newif\iflinenumberLevenifblank
500 \newif\iflinenumberRevenifblank
501 %

```

`\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`.

```

502 \newcommand*{\leftlinenumR}{%
503   \l@dlinenumR
504   \kern\linenumsep}
505 \newcommand*{\rightlinenumR}{%
506   \kern\linenumsep
507   \l@dlinenumR}
508 \newcommand*{\l@dlinenumR}{%
509   \numlabfont\linenumrepR{\line@numR}\@Rlineflag%
510   \ifsublines@
511     \ifnum\subline@num>\z@
512       \unskip\fullstop\sublinenumrepR{\subline@numR}%
513     \fi
514   \fi}
515
516 %

```

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.

```
517 \newcount\line@numR
518 \newcount\subline@numR
519 \newcount\absline@numR
520
521 %
```

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

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

`\actionlines@listR`

`\actions@listR`

```
522 \list@create{\line@listR}
523 \list@create{\insertlines@listR}
524 \list@create{\actionlines@listR}
525 \list@create{\actions@listR}
526
527 %
```

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

```
528 \newcount\page@numR
529
530 %
```

IV.7.2 Specific to `reledpar`

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

`\linesinpar@listR`

`\maxlinesinpar@list`

```
531 \list@create{\linesinpar@listL}
532 \list@create{\linesinpar@listR}
533 \list@create{\maxlinesinpar@list}
534
535 %
```

IV.8 Reading the line-list file

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

```
536 \newcommand{\list@clearing@regR}{%
537   \list@clear{\line@listR}%
538   \list@clear{\insertlines@listR}%
```



```

539 \list@clear{\actionlines@listR}%
540 \list@clear{\actions@listR}%
541 \list@clear{\linesinpar@listR}%
542 \list@clear{\linesonpage@listR}
543 }
544 %

```

\@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.

```

545 \newcommand{\@par@sync@option}[1]{%
546 \IfStrEq{#1}{\@par@this@sync@option}%
547 {}%
548 {\ifledRcol%
549 \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
550 %
551 \else%
552 \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
553 %
554 \fi%
555 \endinput%
556 }%

```

\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.

\@nl

```

557 \newcommand{\@nl@regR}{%
558 \ifx\l@dchset@num\relax \else
559 \advance\absline@numR \@ne
560 \set@line@action
561 \let\l@dchset@num\relax
562 \advance\absline@numR \m@ne
563 \advance\line@numR \m@ne% % do we need this?
564 \fi
565 \advance\absline@numR \@ne

```

```

566 \ifx\next@page@numR\relax \else
567 \page@action
568 \let\next@page@numR\relax
569 \fi
570 \ifx\sub@change\relax \else
571 \ifnum\sub@change>\z@
572 \sublines@true
573 \else
574 \sublines@false
575 \fi
576 \sub@action
577 \let\sub@change\relax
578 \fi
579 \ifcase\@lockR
580 \or
581 \@lockR \tw@
582 \or\or
583 \@lockR \z@
584 \fi
585 \ifcase\sub@lockR
586 \or
587 \sub@lockR \tw@
588 \or\or
589 \sub@lockR \z@
590 \fi
591 \ifsublines@
592 \ifnum\sub@lockR<\tw@
593 \advance\subline@numR \@ne
594 \fi
595 \else
596 \ifnum\@lockR<\tw@
597 \advance\line@numR \@ne \subline@numR \z@
598 \fi
599 \fi}
600
601
602 %

```

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

```

603 \newcount\last@page@numR
604 \last@page@numR=-10000
605
606 %

```

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

- `\@set` The `\@set{<num>}` macro sets the current visible line number to the value specified as its argument. This is used to implement `\setline`. It is defined in `reledmac`.
- `\l@d@set` The `\l@d@set{<num>}` macro sets the line number for the next `\pstart...` to the value specified as its argument. This is used to implement `\setlinenum`. It is defined in `reledmac`.
- `\page@action` `\page@action` adds an entry to the action-code list to change the page number. It is defined in `reledmac`.
- `\set@line@action` `\set@line@action` adds an entry to the action-code list to change the visible line number. It is defined in `reledmac`.
- `\sub@action` `\sub@action` adds an entry to the action-code list to turn sub-lineation on or off, according to the current value of the `\ifsublines@` flag. It is defined in `reledmac`.
- `\do@lockon` `\lock@on` adds an entry to the action-code list to turn line number locking on. The current setting of the sub-lineation flag tells us whether this applies to line numbers or sub-line numbers. It is defined in `reledmac`, however the code specific to right side is defined here, in `\do@lockonR`.

```

607 \newcount \@lockR
608 \newcount \sub@lockR
609
610 \newcommand*{\do@lockonR}{%
611   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
612   \ifsublines@
613     \xright@appenditem{-1005}\to\actions@listR
614     \ifnum\sub@lockR=\z@
615       \sub@lockR \@ne
616     \else
617       \ifnum\sub@lockR=\thr@@
618         \sub@lockR \@ne
619       \fi
620     \fi
621   \else
622     \xright@appenditem{-1003}\to\actions@listR
623     \ifnum\@lockR=\z@
624       \@lockR \@ne
625     \else
626       \ifnum\@lockR=\thr@@
627         \@lockR \@ne
628       \fi
629     \fi
630   \fi}
631
632 %

```

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

```

633
634
635 \newcommand{\do@lockoffR}{%
636   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
637   \ifsublines@
638     \xright@appenditem{-1006}\to\actions@listR
639     \ifnum\sub@lockR=\tw@
640       \sub@lockR \thr@@
641     \else
642       \sub@lockR \z@
643     \fi
644   \else
645     \xright@appenditem{-1004}\to\actions@listR
646     \ifnum\@lockR=\tw@
647       \@lockR \thr@@
648     \else
649       \@lockR \z@
650     \fi
651   \fi}
652
653
654 %

```

`\n@num`

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

`\@ref@regR`
`\insert@countR`

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

```

655 \newcount\insert@countR
656 %

```

- #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.

```

657 \newcommand*{\@ref@regR}[2]{%
658   \global\advance\@edtext@level by 1%
659   \global\insert@countR=#1\relax
660   \loop\ifnum\insert@countR>\z@
661     \xright@appenditem{\the\absline@numR}\to\insertlines@listR

```

```

662 \global\advance\insert@countR \m@ne
663 \repeat
664 %

```

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.

```

665 \begingroup
666 \let\@ref=\dummy@ref
667 \let\@lopR\@gobble
668 \let\page@action=\relax
669 \let\sub@action=\relax
670 \let\set@line@action=\relax
671 \let\@lab=\relax
672 \let\@lemma=\relax
673 \let\@sw\@gobblethree%
674 #2
675 \global\endpage@num=\page@numR
676 \global\endline@num=\line@numR
677 \global\endsubline@num=\subline@numR
678 \endgroup
679 %

```

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

```

680 \xright@appenditem%
681 {\the\page@numR|\the\line@numR|}%
682 \ifsublines@ \the\subline@numR \else 0\fi}%
683 \the\endpage@num|\the\endline@num|}%
684 \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@listR
685 %

```

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

```

686 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
@edtext@level\endcsname}%
687 %

```

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

```

688 \@ref@reg@parseR{#2}%
689 %

```

Decrease edtext level counter.

```

690 \global\advance\@edtext@level by -1%
691 }
692 %

```

```

\@ref@reg@parseR93 \newcommand{\@ref@reg@parseR}[1]{%
694 %
    Declare and init boolean for lemma in this level.
695     \providebool{lemmacommand@the\@edtext@level}%
696     \boolfalse{lemmacommand@the\@edtext@level}%
697 %
    Execute the second argument of \@ref again, to perform for real all the commands
    within it.
698     #1%
699 % Now, we store the list of \protect\cs{sw} of this current \protect\cs{
    edtext} as an element of
700 % the global list of list of \protect\cs{sw} for a \protect\cs{edtext}
    depth.
701 % \begin{macrocode}
702     \ifnum\@edtext@level>0%
703     \def\create@this@edtext@level{\expandafter\list@create\expandafter{
    csname sw@list@edtextR@the\@edtext@level\endcsname}}%
704     \ifcsundef{sw@list@edtextR@the\@edtext@level}{\
    create@this@edtext@level}{}%
705     \letcs{\@tmp}{sw@list@edtextR@the\@edtext@level}%
706     \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}%
707     \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
708     \global\cslet{sw@list@edtextR@the\@edtext@level}{\@tmp}%
709     \fi%
710 }%
711 %

```

\@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.

```

712 \renewcommand*{\@pend}[1]{%
713     \ifbypstart@global\line@num=0\fi%
714     \xright@appenditem{#1}\to\linesinpar@listL}
715 \renewcommand*{\@pendR}[1]{%
716     \ifbypstart@R\global\line@numR=0\fi
717     \xright@appenditem{#1}\to\linesinpar@listR}
718
719 %

```

\@pstart \@pstart and cs@pstartR allows us to know, when using \nomaxlines option in which
\@pstartR 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

```

720 \newcommand{\@pstart}[3]{%
721     \ifcsdef{minpage@pstart@#1}%
722     {\ifnumgreater{#2}{\csuse{minpage@pstart@#1}}%

```

```

723     {\csnumgdef{minpage@pstart@#1}{#2}}%
724     {}%
725     }%
726     {\csnumgdef{minpage@pstart@#1}{#2}}
727     \csnumgdef{afterlines@pstart@#1L}{#3}%
728     }%
729
730 \newcommand{\@pstartR}[3]{%
731   \numdef{\@tmp}{#2-1}%Because we have not to know in which page the pstart
732   starts, but in which pair of facing page
733   \ifcsdef{minpage@pstart@#1}%
734     {\ifnumgreater{\@tmp}{\csuse{minpage@pstart@#1}}}%
735     {\csnumgdef{minpage@pstart@#1}{\@tmp}}%
736     {}%
737     }%
738     {\csnumgdef{minpage@pstart@#1}{\@tmp}}
739     \csnumgdef{afterlines@pstart@#1R}{#3}%
740 }%
741 %

```

`\@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`.

```

741 \renewcommand*{\@lopL}[1]{%
742   \xright@appenditem{#1}\to\linesonpage@listL}
743 \renewcommand*{\@lopR}[1]{%
744   \xright@appenditem{#1}\to\linesonpage@listR}
745
746 %

```

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`.

```

747 \newwrite\linenum@outR
748 %

```

`\iffirst@linenum@outR` 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
749 \newif\iffirst@linenum@out@R
750 \first@linenum@out@Rtrue
751 %

```

\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.

```

752 \newcommand*{\line@list@stuffR}[1]{%
753   \read@linelist{#1}%
754   \iffirst@linenum@out@R
755     \immediate\closeout\linenum@outR
756     \global\first@linenum@out@Rfalse
757     \immediate\openout\linenum@outR=\l@auxdir#1%
758     \immediate\write\linenum@outR{\string\line@list@version{\
this@line@list@version}}}%
759   \ifl@dpaging%
760     \immediate\write\linenum@outR{\string\@par@sync@option{\
@par@this@sync@option}}}%
761   \fi%
762   \else
763     \if@minipage%
764       \leavevmode%
765       \fi%
766     \closeout\linenum@outR%
767     \openout\linenum@outR=\l@auxdir#1%
768   \fi}
769
770 %

```

\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.

```

771 \newcommand*{\new@lineL}{%
772   \write\linenum@out{\string\@nl[\the\c@page][\thepage]}}
773 %

```

\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.

```

774 \newcommand*{\new@lineR}{%
775   \write\linenum@outR{\string\@nl[\the\c@page][\thepage]}}
776 %

```

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

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

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

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

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

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

`\endlock`

`\skipnumbering`

V Marking text for notes

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

`\critext`

`\edtext`

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

V.1 Specific hooks and commands for notes

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

```
\newseries@par77 \newcommand{\newseries@par}[1]{%
778 %
```

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.

```
779 \unless\ifnofamiliar@%
780 \csgdef{onlysideX@#1}{}%
781 \newnamebox{footins#1@kept}%
782 \fi%
```

```

783 \unless\ifnocritical@%
784 \csgdef{Xonlyside@#1}{}%
785 \newnamebox{#1footins@kept}%
786 \fi%
787 %

```

V.1.2 Tools specific to familiar footnotes

```

788 \unless\ifnofamiliar@%
789 %

```

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 files to be used on the next run for the main text.

So here, we only defined the new counter.

```

790 \newcounter{footnote#1@typeset}%
791 %

```

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.

```

792 \expandafter\list@create\csname footnote#1@mk\endcsname%
793 %

```

Then, declare the `\footnoteXnomk` command.

```

794 \expandafter\newcommand\csname footnote#1nomk\endcsname[1]{%
795 %

```

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

```

796 \toggletrue{nomk0}%
797 \csuse{footnote#1}{##1}%
798 \togglefalse{nomk0}%
799 %

```

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`.

```

800 \letcs{\@tmp}{footnote#1@mk}%
801 \numdef\@tmpa{\csuse{c@footnote#1}}%
802 \global\xright@appenditem{\@tmpa}\to\@tmp%
803 \global\cslet{footnote#1@mk}{\@tmp}%
804 }%
805 %

```

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

```

806 \expandafter\newcommand\csname footnote#1mk\endcsname{%
807 %

```

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

```

808 \letcs{\@tmp}{footnote#1@mk}%
809 \gl@p\@tmp\to\@tmpa%
810 \global\cslet{footnote#1@mk}{\@tmp}%
811 %

```

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

```

812 \letcs{\old@footnote}{c@footnote#1}%
813 \setcounter{footnote#1}{\@tmpa}%
814 %

```

Define the footnote mark and print it

```

815 \protected@csxdef{\thefnmark#1}{\csuse{thefootnote#1}}%
816 \csuse{@footnotemark#1}%
817 %

```

Restore previous footnote counter and finally add space.

```

818 \setcounter{footnote#1}{\old@footnote}%
819 \xspace%
820 }%
821 %

```

End of tools specific to familiar notes.

```

822 \fi
823 %

```

End of `\newseries@par`.

```
824 }%
825 %
```

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.

```
826 \newcommand{\save@familiarfootnote@number}{%
827   \unless\ifnofamiliar@%
828     \def\do##1{\csxdef{saved@footnote##1}{\the\csname c@footnote##1\endcsname}}%
829   \dolistloop{\@series}%
830   \fi%
831   \xdef\saved@footnote{\the\c@footnote}%
832 }
833 \newcommand{\get@familiarfootnote@number}{%
834   \unless\ifnofamiliar@%
835     \def\do##1{\setcounter{footnote##1@typeset}{\csuse{saved@footnote##1}}}%
836   \dolistloop{\@series}%
837   \fi%
838   \setcounter{footnote@typeset}{\saved@footnote}%
839 }
840 %
```

V.2 Create hooks

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

```
841 \unless\ifnocritical@%
842   \newhookcommand@series{Xonlyside}%
843 \fi%
844 \unless\ifnofamiliar@%
845   \newhookcommand@series{onlysideX}%
846 \fi
847
848
849 %
```

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`.

```
850 \newcommand{\init@series@par}{%
851   \def\do##1{\newseries@par{##1}}%
```

```

852 \dolistloop{\@series}%
853 }%
854 \init@series@par%
855 %

```

V.4 Tools specific to L^AT_EX's classical footnotes

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

```

\footnote@reading56 \newcount\footnote@reading%
\footnote@typeset57 \newcounter{footnote@typeset}%
858 %

```

VI Pstart numbers dumping and restoration

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

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

```

\list@pstartL@pc59 \list@create{\list@pstartL@pc}%
\list@pstartR@pc60 \list@create{\list@pstartR@pc}%
861 %

```

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

```

\dump@pstartL@pc62 \def\dump@pstartL@pc{%
\dump@pstartR@pc63 \xright@appenditem{\the\c@pstartL}\to\list@pstartL@pc%
864 \global\cslet{numberpstart@L\the\l@dnumpstartsL}{\ifnumberpstart}%
865 }%
866
867 \def\dump@pstartR@pc{%
868 \xright@appenditem{\the\c@pstartR}\to\list@pstartR@pc%
869 \global\cslet{numberpstart@R\the\l@dnumpstartsR}{\ifnumberpstart}%
870 }%
871
872 %

```

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

`\restore@pstartR@pc`

```

873 \def\restore@pstartL@pc{%
874   \ifx\list@pstartL@pc\empty\else%
875     \gl@p\list@pstartL@pc\to\@temp%
876     \global\c@pstartL=\@temp%
877   \fi%
878 }%
879 \def\restore@pstartR@pc{%
880   \ifx\list@pstartR@pc\empty\else%
881     \gl@p\list@pstartR@pc\to\@temp%
882     \global\c@pstartR=\@temp%
883   \fi%
884 }%
885 %

```

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.

```

886 \newenvironment{pairs}{%
887   \l@dpairingtrue
888   \l@dpagingfalse
889   \initnumbering@quote
890   \save@familiarfootnote@number%
891   \if@ledgroup%
892     \get@familiarfootnote@number%
893   \fi%
894   \save@section@number%
895   \at@begin@pairs%
896 }{%
897   \l@dpairingfalse
898 }
899 %
900 %

```

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

```

901 \newcommand{\AtBeginPairs}[1]{\gdef\at@begin@pairs{#1}}%
902 \def\at@begin@pairs{}%
903 %
904 %

```

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.

```

905 \newenvironment{pages}{%
906   \l@dpairingtrue
907   \l@dpagingtrue
908   \initnumbering@quote
909   \save@familiarfootnote@number%
910   \if@ledgroup%
911     \get@familiarfootnote@number%
912   \fi%
913   \save@section@number%
914   \setlength{\Lcolwidth}{\textwidth}%
915   \setlength{\Rcolwidth}{\textwidth}%
916 }{%
917   \l@dpairingfalse
918   \l@dpagingfalse
919 }
920
921 %

```

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

```

922 \newif\ifinstanzaL
923 \newif\ifinstanzaR
924 %

```

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.

```

925 \newenvironment{Leftside}{%
926   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname\else%
927     \led@err@Leftside@PreviousNotPrinted%
928   \fi%
929   \ledRcolfalse
930   \setcounter{pstartL}{1}
931   \let\pstart\pstartL
932   \let\thepstart\thepstartL
933   \let\pend\pendL
934   \let\memorydump\memorydumpL
935   \Leftsidehook
936   \let\old@startstanza\@startstanza
937   \def\@startstanza[##1]{\global\instanzaLtrue\old@startstanza[##1]}
938 }{
939   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname%
940     \led@error@missing@numbering{Leftside}%

```

```

941 \fi%
942 \Leftsidehookend}
943 %

```

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

```

\Leftsidehookend
\Rightsidehook
\Rightsidehookend
944 \newcommand*{\Leftsidehook}{}
945 \newcommand*{\Leftsidehookend}{}
946 \newcommand*{\Rightsidehook}{}
947 \newcommand*{\Rightsidehookend}{}
948
949 %

```

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.

```

950 \newenvironment{Rightside}{%
951 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else%
952 \led@err@Rightside@PreviousNotPrinted%
953 \fi%
954 \ledRcoltrue
955 \let\beginnumbering\beginnumberingR
956 \let\endnumbering\endnumberingR
957 \let\pausenumbering\pausenumberingR
958 \let\resumenumbering\resumenumberingR
959 \let\memorydump\memorydumpR
960 \let\thepstart\thepstartR
961 \let\pstart\pstartR
962 \let\pend\pendR
963 \let\ledpb\ledpbR
964 \let\lednopb\lednopbR
965 \let\lineation\lineationR
966 \Rightsidehook
967 \let\old@startstanza\@startstanza
968 \def\@startstanza[##1]{\global\instanzaRtrue\old@startstanza[##1]}
969 }{%
970 \ledRcolfalse
971 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
972 \led@error@missing@numbering{Rightside}%
973 \fi%
974 \Rightsidehookend
975 }
976
977 %

```


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.

```
978 \newcount\num@linesR
979 \newbox\one@lineR
980 \newcount\par@lineR
981 %
```

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

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

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

```
982
983 \newcounter{pstartL}
984 \renewcommand{\thepstartL}{\bfseries\@arabic\c@pstartL}. }
985 \newcounter{pstartR}
986 \renewcommand{\thepstartR}{\bfseries\@arabic\c@pstartR}. }
987
988 \newcommandx*{\pstartL}[1][1]{%
989   \if@nobreak%
990     \let\@oldnobreak\@nobreaktrue%
991   \else%
992     \let\@oldnobreak\@nobreakfalse%
993   \fi%
994   \@nobreaktrue%
995   \ifluatex%
```

```

996 \xdef\l@luatexttextdir@L{\the\textdir}%
997 \xdef\l@luatexpardir@L{\the\pardir}%
998 \xdef\l@luatexbodydir@L{\the\bodydir}%
999 \fi%
1000 \ifnumbering \else%
1001 \led@err@PstartNotNumbered%
1002 \beginnumbering%
1003 \fi%
1004 \ifnumberedpar@%
1005 \led@err@PstartInPstart%
1006 \pend%
1007 \fi%
1008 %

```

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

```

1009 \ifpstart@rtedL\else%
1010 \list@clear{\inserts@list}%
1011 \global\let\next@insert=\empty%
1012 \global\pstart@rtedLtrue%
1013 \fi%
1014 \begingroup\everypar{}%
1015 %

```

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.

```

1016 \global\advance\l@dnumpstartsL \one%
1017 \ifnum\l@dnumpstartsL>\l@dc@maxchunks%
1018 \led@err@TooManyPstarts%
1019 \global\l@dnumpstartsL=\l@dc@maxchunks%
1020 \fi%
1021 \global\setnamebox{\l@dLcolrawbox\the\l@dnumpstartsL}=\vbox\bgroup%
1022 %

```

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.

```

1023 \l@dzeropenalties%
1024 \ifautopar\else%
1025 \ifnumberpstart%
1026 \ifsidepstartnum%
1027 \else%
1028 \thepstartL%
1029 \fi%
1030 \fi%
1031 \fi%
1032 \hsize=\Lcolwidth%
1033 \numberedpar@true%
1034 \iflabelpstart\protected@edef\@currentlabel%

```

```

1035     {\p@pstartL\thepstartL}\fi%
1036 %

```

Dump the optional arguments

```

1037 \ifstrempy{#1}%
1038   {\csgdef{before@pstartL@the\l@dnumpstartsL}{\at@every@pstart}}%
1039   {\csgdef{before@pstartL@the\l@dnumpstartsL}{\noindent#1}}%
1040   \at@every@pstart@call%
1041 %

```

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

```

1042 \ignorespaces%
1043 %

```

```

1044 }
1045 %

```

The same for right side.

```

1046 \newcommandx*{\pstartR}[1][1]{%
1047   \if@nbreak%
1048     \let\@oldnbreak\@nbreaktrue%
1049   \else%
1050     \let\@oldnbreak\@nbreakfalse%
1051   \fi%
1052   \@nbreaktrue%
1053   \ifluatex%
1054     \xdef\l@luatextextdir@R{\the\textdir}%
1055     \xdef\l@luatexpardir@R{\the\pardir}%
1056     \xdef\l@luatexbodydir@R{\the\bodydir}%
1057   \fi%
1058   \ifnumberingR \else%
1059     \led@err@PstartNotNumbered%
1060     \beginnumberingR%
1061   \fi%
1062   \ifnumberedpar@%
1063     \led@err@PstartInPstart%
1064     \pendR%
1065   \fi%
1066   \ifpstart@rtedR\else%
1067     \list@clear{\inserts@listR}%
1068     \global\let\next@insertR=\empty%
1069     \global\pstart@rtedRtrue%
1070   \fi%
1071   \begingroup\everypar{}%
1072   \global\advance\l@dnumpstartsR \@ne%
1073   \ifnum\l@dnumpstartsR>\l@dc@maxchunks%
1074     \led@err@TooManyPstarts%
1075     \global\l@dnumpstartsR=\l@dc@maxchunks%
1076   \fi%

```

```

1077 \global\setnamebox{l@dRcolrawbox\the\l@dnumstartR}=\vbox\bgroup%
1078 \l@dzeropenalties%
1079 \ifautopar\else%
1080 \ifnumberpstart%
1081 \ifsidepstartnum\else%
1082 \thepstartR%
1083 \fi%
1084 \fi%
1085 \fi%
1086 \hsize=\Rcolwidth%
1087 \numberedpar@true%
1088 \iflabelpstart\protected@edef\@currentlabel%
1089 {\p@pstartR\thepstartR}\fi%
1090 \ifstrempy{#1}%
1091 {\csgdef{before@pstartR@the\l@dnumstartR}{\at@every@pstart}}%
1092 {\csgdef{before@pstartR@the\l@dnumstartR}{\noindent#1}}%
1093 \at@every@pstart@call%
1094 \ignorespaces%
1095 }
1096 %

```

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

```

1097 \newcommand*{\pendL}[1][1]{%
1098 \ifnumbering\else%
1099 \led@err@PendNotNumbered%
1100 \fi%
1101 \ifnumberedpar@\else%
1102 \led@err@PendNoPstart%
1103 \fi%
1104 %

```

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.

```

1105 \endgraf\global\num@lines=\prevgraf\egroup%
1106 \global\par@line=0%
1107 %

```

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

```

1108 \endgroup%
1109 \ignorespaces%
1110 \@oldnobreak%
1111 \dump@pstartL@pc%
1112 \ifnumberpstart%
1113 \addtocounter{pstartL}{1}%
1114 \fi
1115 \parledgroup@beforenotes@save{L}%
1116 %

```

Dump content of the optional argument.

```

1117 \ifstrempy{#1}%
1118   {\csgdef{after@pendL@the\l@dnumpstartsL}{\at@every@pend}}%
1119   {\csgdef{after@pendL@the\l@dnumpstartsL}{\noindent#1}}%
1120 }
1121 %

```

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

```

1122 \newcommand*{\pendR}[1][1]{%
1123   \ifnumberingR \else%
1124     \led@err@PendNotNumbered%
1125   \fi%
1126   \ifnumberedpar@ \else%
1127     \led@err@PendNoPstart%
1128   \fi%
1129   \endgraf\global\num@linesR=\prevgraf\egroup%
1130   \global\par@lineR=0%
1131   \endgroup%
1132   \ignorespaces%
1133   \@oldnobreak%
1134   \dump@pstartR@pc%
1135   \ifnumberpstart%
1136     \addtocounter{pstartR}{1}%
1137   \fi%
1138   \parledgroup@beforenotes@save{R}%
1139   \ifstrempy{#1}%
1140     {\csgdef{after@pendR@the\l@dnumpstartsR}{\at@every@pend}}%
1141     {\csgdef{after@pendR@the\l@dnumpstartsR}{\noindent#1}}%
1142 }
1143 %
1144 %

```

\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.

```

1145 \newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
1146 \gdef\at@every@pstart@call{}%
1147 %

```

\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

```

1148 \newif\ifprint@last@after@pendL%
1149 \newif\ifprint@last@after@pendR%
1150 %

```

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`

```
1151 \newbox\l@dleftbox
1152 \newbox\l@drightbox
1153
1154 %
```

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

```
\countRline
1155 \newcount\countLline
1156 \countLline \z@
1157 \newcount\countRline
1158 \countRline \z@
1159
1160 %
```

`\@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`

```
1161 \newcount\@donereallinesL
1162 \newcount\@donetotallinesL
1163 \newcount\@donereallinesR
1164 \newcount\@donetotallinesR
1165
1166 %
```

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

```
1167 \newcommand*{\do@lineL}{%
1168 \letcs{\ifnumberpstart}{numberpstart@L\the\l@dpscl}%
1169 \advance\countLline \one%
1170 \ifvbox\namebox{\l@dLcolrawbox\the\l@dpscl}%
1171 {\vbadness=10000%
1172 \splittopskip=\z@%
1173 \do@lineLhook%
1174 \l@demtyd@ta%
1175 \global\setbox\one@line=\vsplit\namebox{\l@dLcolrawbox\the\l@dpscl}%
1176 to\baselineskip}%
1177 \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startL}{}%
1178 \unvbox\one@line \global\setbox\one@line=\lastbox%
1179 \@writepageofparL%
```

```

1180 \getline@numL%
1181 \ifnum\@lock>\@ne%
1182   \inserthangingsymboltrue%
1183 \else%
1184   \inserthangingsymbolfalse%
1185 \fi%
1186 \setbox\l@dleftbox%
1187 \hb@xt@ \Lcolwidth{%
1188   \ifl@dhidenumber%
1189     \global\l@dhidenumberfalse%
1190     \f@x@l@cks%
1191   \else%
1192     \affixline@num%
1193   \fi%
1194   \xifinlist{\the\l@dpscL}{\eled@sections@@}%
1195   {%
1196     \if@firstlineofpage%
1197       \set@Xtxtbeforenotes%
1198       \global\@firstlineofpagefalse%
1199     \fi%
1200     \ifdefstring{ms@data@position}{msdata-regular}{%
1201       \insert@msdata%
1202       \add@inserts%
1203       \add@Xgroupbyline%
1204     }{%
1205       \add@inserts%
1206       \add@Xgroupbyline%
1207       \insert@msdata%
1208     }%
1209     \affixside@note%
1210   }%
1211   {\print@lineL}%
1212 }%
1213 \add@penaltiesL%
1214 \global\advance\@donereallinesL\@ne%
1215 \global\advance\@donetotallinesL\@ne%
1216 \else%
1217   \iflinenumberLevenifblank
1218     \new@lineL%
1219     \l@emptyd@ta%
1220     \getline@numL%
1221     \affixline@num%
1222     \setbox\l@dleftbox \hb@xt@ \Lcolwidth{%
1223       \l@dld@ta%
1224       \hspace*{\Lcolwidth}%
1225       \ledrlfill\l@drd@ta%
1226     }%
1227   \else%
1228     \setbox\l@dleftbox \hb@xt@ \Lcolwidth{\hspace*{\Lcolwidth}}%
1229   \fi%

```

```

1230 \global\advance\@donetotallinesL\@ne%
1231 \fi%
1232 }%
1233
1234
1235 %

```

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

```

1236 \def\print@lineL{%
1237   \affixpstart@numL%
1238   \l@dld@ta%
1239   \if@firstlineofpage%
1240     \set@Xtxtbeforenotes%
1241     \global\@firstlineofpagefalse%
1242   \fi%
1243   \ifdefstring{\ms@data@position}{msdata-regular}{%
1244     \insert@msdata%
1245     \add@inserts%
1246     \add@Xgroupbyline%
1247   }{%
1248     \add@inserts%
1249     \add@Xgroupbyline%
1250     \insert@msdata%
1251   }%
1252   \affixside@note%
1253   \l@dlsn@te%
1254   \do@insidelineLhook%
1255   \hb@xt@ \Lcolwidth{\ledllfill\hb@xt@ \wd\one@line{%
1256     \ifluatex%
1257       \texdir\l@luatextextdir@L%
1258     \fi%
1259     \new@lineL%
1260     \inserthangingsymbolL%
1261     \l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta%
1262   \l@drsn@te}}%
1263
1264 %

```

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

```

1265 \def\print@eledsectionL{%%
1266   \addtocounter{pstartL}{-1}%
1267   \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{%
1268     \ifdefstring{\@eledsectmark}{L}{\ledsectnomark}%
1269     \numdef{\temp@}{\l@dpscL-1}%
1270     \xifinlist{\temp@}{\eled@sections@}{\@nbreaktrue}{\@nbreakfalse}%
1271     \@eled@sectioningtrue%
1272   \bgroup%

```



```

1273 \ifluatex%
1274 \texdir\l@luatextextdir@L%
1275 \pardir\l@luatexpardir@L%
1276 \bodydir\l@luatexbodydir@L%
1277 \ifdefstring{\l@luatextextdir@L}{TRT}{\@RTLtrue}{}%
1278 \fi%
1279 \csuse{eled@sectioning@the\l@dpscl}%
1280 \egroup%
1281 \@eled@sectioningfalse%
1282 \global\csundef{eled@sectioning@the\l@dpscl}%
1283 \if@RTL%
1284 \hspace{-3\paperwidth}%
1285 {\hbox{\l@dunhbox@line{one@line}} \new@line}%
1286 \else%
1287 \hspace{3\paperwidth}%
1288 {\new@line \hbox{\l@dunhbox@line{one@line}}}%
1289 \fi%
1290 \vskip\eledsection@correcting@skip%
1291 }
1292 %
1293 %

```

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

```

\dolinehook
\dolinehook
\doinssidehook
\doinssidehook
1294 \newcommand*\dolinehook[1]{\gdef\do@linehook{#1}}%
1295 \newcommand*\dolinehook[1]{\gdef\do@linehook{#1}}%
1296 \newcommand*\doinssidehook[1]{\gdef\do@insidelinehook{#1}}%
1297 \newcommand*\doinssidehook[1]{\gdef\do@insidelinehook{#1}}%
1298 %
1299 %

```

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

```

\do@linehook
\do@linehook
1300 \newcommand*\do@linehook{}
\doinssidehook
1301 \newcommand*\do@linehook{}
\doinssidehook
1302 \newcommand*\do@insidelinehook{}
1303 \newcommand*\do@insidelinehook{}
1304 %
1305 %

```

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

```

1306 \newcommand*\do@lineR{%
1307 \let\linenumrepL\linenumrep%
1308 \let\sublinenumrepL\sublinenumrep%
1309 \let\linenumrep\linenumrepR%
1310 \let\sublinenumrep\sublinenumrepR%
1311 \letcs{ifnumberpstart}{numberpstart@R\the\l@dpscl}%

```

```

1312 \ledRcol@true%
1313 \advance\countRline \@ne%
1314 \ifvbox\namebox{1@dRcolrawbox\the\1@dpscR}%
1315 {\vbadness=10000%
1316 \splittopskip=\z@%
1317 \do@lineRhook%
1318 \l@emptyd@ta%
1319 \global\setbox\one@lineR=\vsplit\namebox{1@dRcolrawbox\the\1@dpscR}%
1320 to\baselineskip}%
1321 \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startR}{}%
1322 \unvbox\one@lineR \global\setbox\one@lineR=\lastbox%
1323 \@writepageofparR%
1324 \getline@numR%
1325 \ifnum\@lockR>\@ne%
1326 \inserthangingsymbolRtrue%
1327 \else%
1328 \inserthangingsymbolRfalse%
1329 \fi%
1330 \setbox\l@drightrightbox%
1331 \hb@xt@ \Rcolwidth{%
1332 \ifl@dhiddenumber%
1333 \global\l@dhiddenumberfalse%
1334 \f@x@l@cksR%
1335 \else%
1336 \affixline@numR%
1337 \fi%
1338 \xifinlist{\the\1@dpscR}{\eled@sectionsR@@}%
1339 {%
1340 \if@firstlineofpageR%
1341 \set@Xtxtbeforenotes%
1342 \global\@firstlineofpageRfalse%
1343 \fi%
1344 \ifdefstring{\ms@data@position}{msdata-regular}{%
1345 \insert@msdata%
1346 \add@insertsR%
1347 \add@Xgroupbyline%
1348 }{%
1349 \add@insertsR%
1350 \add@Xgroupbyline%
1351 \insert@msdata%
1352 }%
1353 \affixside@noteR%
1354 }%
1355 {\print@lineR}%
1356 }%
1357 \add@penaltiesR%
1358 \global\advance\@donereallinesR\@ne%
1359 \global\advance\@donetotallinesR\@ne%
1360 \else%

```

```

1361 \iflinenumberRevenifblank%
1362 \new@lineR
1363 \l@emptyd@ta%
1364 \getline@numR%
1365 \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1366 \affixline@numR%
1367 \l@dld@ta%
1368 \hspace*{\Rcolwidth}%
1369 \ledrlfill\l@drd@ta%
1370 }%
1371 \else%
1372 \setbox\l@drightbox \hb@xt@ \Rcolwidth{\hspace*{\Rcolwidth}}%
1373 \fi%
1374 \global\advance\@donetotallinesR\@ne%
1375 \fi%
1376 \ledRcol@false%
1377 \let\linenumrep\linenumrepL%
1378 \let\sublinenumrep\sublinenumrepL%
1379 }
1380
1381
1382 %

```

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

VIII.3 Line and page number computation

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

```

1383 \newcommand*{\getline@numR}{%
1384 \global\advance\absline@numR \@ne
1385 \do@actionsR
1386 \do@ballastR
1387 \ifledgroupnotesR\else
1388 \ifnumberline
1389 \ifsublines@
1390 \ifnum\sub@lockR<\tw@
1391 \global\advance\subline@numR \@ne
1392 \fi
1393 \else
1394 \ifnum\@lockR<\tw@
1395 \global\advance\line@numR \@ne
1396 \global\subline@numR \z@
1397 \fi
1398 \fi
1399 \fi
1400 \fi
1401 }
1402 \newcommand*{\getline@numL}{%

```

```

1403 \global\advance\absline@num \@ne
1404 \do@actions
1405 \do@ballast
1406   \ifledgroupnotesL@\else
1407     \ifnumberline
1408       \ifsublines@
1409         \ifnum\sub@lock<\tw@
1410           \global\advance\subline@num \@ne
1411         \fi
1412       \else
1413         \ifnum\@lock<\tw@
1414           \global\advance\line@num \@ne
1415           \global\subline@num \z@
1416         \fi
1417       \fi
1418     \fi
1419   \fi
1420 }
1421
1422 %
1423 %

```

`\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.

```

1424 \newcommand*{\do@ballastR}{\global\ballast@count=\z@
1425   \begingroup
1426     \advance\absline@numR \@ne
1427     \ifnum\next@actionlineR=\absline@numR
1428       \ifnum\next@actionR>-1001
1429         \global\advance\ballast@count by -\c@ballast
1430       \fi
1431     \fi
1432   \endgroup}
1433 %

```

`\l@dskipversenumberR` The `\do@actionsR` macro looks at the list of actions to take at particular right text absolute line numbers, and does everything that is specified for the current line.
`\do@actionsR`
`\do@actions@fixedcodeR` It may call itself recursively and we use tail recursion, via `\do@actions@nextR` for this.
`\do@actions@nextR`

```

1434
1435 \newif\ifl@dskipversenumberR
1436 \newcommand*{\do@actions@fixedcodeR}{%
1437   \ifcase\@l@dttempcnta%
1438     \or% % 1001 = starting sublineation
1439       \global\sublines@true
1440     \or% % 1002 = ending sublineation
1441       \global\sublines@false
1442     \or% % 1003 = starting locking number

```

```

1443 \global\@lockR=\@ne
1444 \or% % 1004 = ending locking number
1445 \ifnum\@lockR=\tw@
1446 \global\@lockR=\thr@@
1447 \else
1448 \global\@lockR=\z@
1449 \fi
1450 \or% % 1005 = starting locking subnumber
1451 \global\sub@lockR=\@ne
1452 \or% % 1006 = ending locking subnumber
1453 \ifnum\sub@lockR=\tw@
1454 \global\sub@lockR=\thr@@
1455 \else
1456 \global\sub@lockR=\z@
1457 \fi
1458 \or% % 1007 = skipping numbering
1459 \l@dskipnumbertrue
1460 \or% % 1008 = skipping numbering in stanza
1461 \l@dskipversenumberRtrue%
1462 \or% % 1009 = hiding number
1463 \l@dhidenumbertrue%
1464 \or% % 1010 = inserting msdata
1465 \add@msdata%
1466 \else%
1467 \led@warn@BadAction
1468 \fi%
1469 }
1470
1471
1472 \newcommand*{\do@actionsR}{%
1473 \global\let\do@actions@nextR=\relax
1474 \@l@tempcntb=\absline@numR
1475 \ifnum\@l@tempcntb<\next@actionlineR\else
1476 \ifnum\next@actionR>-1001\relax
1477 \@firstlineofpageRtrue%
1478 \global\page@numR=\next@actionR
1479 \ifbypage@R
1480 \global\line@numR \z@ \global\subline@numR \z@
1481 \fi
1482 \add@msdata@firstlineofpage%
1483 \else
1484 \ifnum\next@actionR<-4999\relax % 9/05 added relax here
1485 \@l@tempcnta=-\next@actionR
1486 \advance\@l@tempcnta by -5001\relax
1487 \ifsublines@
1488 \global\subline@numR=\@l@tempcnta
1489 \else
1490 \global\line@numR=\@l@tempcnta
1491 \fi
1492 \else

```

```

1493     \@l@tempcnta=-\next@actionR
1494     \advance\@l@tempcnta by -1000\relax
1495     \do@actions@fixedcodeR
1496     \fi
1497   \fi
1498   \ifx\actionlines@listR\empty
1499     \gdef\next@actionlineR{1000000}%
1500   \else
1501     \glp\actionlines@listR\to\next@actionlineR
1502     \glp\actions@listR\to\next@actionR
1503     \global\let\do@actions@nextR=\do@actionsR
1504   \fi
1505 \fi
1506 \do@actions@nextR}
1507
1508 %

```

VIII.4 Line number printing

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

```

1509 \ch@cksub@l@ckR
1510 \ch@ck@l@ckR
1511 \f@x@l@ckR
1512 \affixline@numR
1513
1514 \newcommand*{\l@dcalcnm}[3]{%
1515   \ifnum #1 > #2\relax
1516     \@l@tempcnta = #1\relax
1517     \advance\@l@tempcnta by -#2\relax
1518     \divide\@l@tempcnta by #3\relax
1519     \multiply\@l@tempcnta by #3\relax
1520     \advance\@l@tempcnta by #2\relax
1521   \else
1522     \@l@tempcnta=#2\relax
1523   \fi}
1524
1525 \newcommand*{\ch@cksub@l@ckR}{%
1526   \ifcase\sub@lockR
1527   \or
1528     \ifnum\sublock@disp=\@ne
1529       \@l@tempcntb \z@ \@l@tempcnta \@ne
1530     \fi
1531   \or
1532     \ifnum\sublock@disp=\tw@
1533     \else
1534       \@l@tempcntb \z@ \@l@tempcnta \@ne
1535     \fi
1536   \or
1537     \ifnum\sublock@disp=\z@
1538       \@l@tempcntb \z@ \@l@tempcnta \@ne
1539     \fi
1540   \fi}

```

```

1537 \newcommand*{\ch@ck@l@ckR}{%
1538   \ifcase\@lockR
1539   \or
1540   \ifnum\lock@disp=\@ne
1541     \@l@tempcntb \z@ \l@tempcnta \@ne
1542   \fi
1543   \or
1544   \ifnum\lock@disp=\tw@
1545   \else
1546     \@l@tempcntb \z@ \l@tempcnta \@ne
1547   \fi
1548   \or
1549   \ifnum\lock@disp=\z@
1550     \@l@tempcntb \z@ \l@tempcnta \@ne
1551   \fi
1552 \fi}
1553
1554 \newcommand*{\f@x@l@cksR}{%
1555   \ifcase\@lockR
1556   \or
1557   \global\@lockR \tw@
1558   \or \or
1559   \global\@lockR \z@
1560   \fi
1561   \ifcase\sub@lockR
1562   \or
1563   \global\sub@lockR \tw@
1564   \or \or
1565   \global\sub@lockR \z@
1566   \fi}
1567
1568
1569
1570 \let\linenumberlistR\empty%
1571 \newcommand*{\affixline@numR}{%
1572   \ifledgroupnotesR\else\ifnumberline
1573   \ifl@dskipnumber
1574     \global\l@dskipnumberfalse
1575   \else
1576     \ifsublines@
1577       \@l@tempcntb=\subline@numR
1578       \l@dcalcnnum{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR}
1579     }%
1580     \ch@cksub@lockR
1581   \else
1582     \@l@tempcntb=\line@numR
1583     \ifx\linenumberlistR\empty%
1584       \l@dcalcnnum{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%
1585     \else
1586       \@l@tempcnta=\line@numR

```

```

1586 \edef\rem@inderR{\linenumberlistR,\number\line@numR,}%
1587 \edef\sc@n@list{\def\noexpand\sc@n@list
1588   ###1,\number\@l@tempcnta,###2|{\def\noexpand\rem@inderR{###2}}}%
%
1589 \sc@n@list\expandafter\sc@n@list\rem@inderR|
1590 \ifx\rem@inderR\empty\advance\@l@tempcnta\@ne\fi%
1591 \fi
1592 \ch@ck@l@ckR
1593 \fi
1594 \ifnum\@l@tempcnta=\@l@tempcntb
1595 \ifl@dskipversenumberR\else
1596 \if@twocolumn
1597 \if@firstcolumn
1598 \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1599 \else
1600 \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1601 \fi
1602 \else
1603 \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columnsR}{\m@ne}}}%
1604 {\@l@tempcntb=\line@margin@columnsR}%
1605 {\@l@tempcntb=\line@marginR}%
1606 \ifnum\@l@tempcntb>\@ne
1607 \advance\@l@tempcntb by\page@numR
1608 \fi
1609 \ifodd\@l@tempcntb
1610 \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1611 \else
1612 \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1613 \fi
1614 \fi
1615 \fi
1616 \fi
1617 \f@x@l@cksR
1618 \fi
1619 \fi
1620 \fi}
1621 %

```

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@numL1622
\affixpstart@numR1623 \newcommand*{\affixpstart@numL}{%
\leftpstartnum1624 \ifsidepstartnum
\rightpstartnum1625 \if@twocolumn
\leftpstartnum1626 \if@firstcolumn
\rightpstartnum1627 \gdef\l@dld@ta{\llap{\leftpstartnumL}}}%
\leftpstartnum1628 \else
\rightpstartnum1629 \gdef\l@dld@ta{\rlap{\rightpstartnumL}}}%
1630 \fi
1631 \else
1632 \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\line@margin@columns}{\m@ne}}}%
1633 {\@l@tempcntb=\line@margin@columns}%
1634 {\@l@tempcntb=\line@margin}%
1635 \ifnum\@l@tempcntb>\@ne
1636 \advance\@l@tempcntb \page@num
1637 \fi
1638 \ifodd\@l@tempcntb
1639 \gdef\l@dld@ta{\rlap{\rightpstartnumL}}}%
1640 \else
1641 \gdef\l@dld@ta{\llap{\leftpstartnumL}}}%
1642 \fi
1643 \fi
1644 \fi
1645 }
1646 \newcommand*{\affixpstart@numR}{%
1647 \ifsidepstartnum
1648 \if@twocolumn
1649 \if@firstcolumn
1650 \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
1651 \else
1652 \gdef\l@dld@ta{\rlap{\rightpstartnumR}}}%
1653 \fi
1654 \else
1655 \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\line@margin@columnsR}{\m@ne}}}%
1656 {\@l@tempcntb=\line@margin@columnsR}%
1657 {\@l@tempcntb=\line@marginR}%
1658 \ifnum\@l@tempcntb>\@ne
1659 \advance\@l@tempcntb \page@numR
1660 \fi
1661 \ifodd\@l@tempcntb
1662 \gdef\l@dld@ta{\rlap{\rightpstartnumR}}}%
1663 \else
1664 \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
1665 \fi
1666 \fi
1667 \fi
1668 }

```

```

1669 \newcommand*\leftpstartnumL){
1670 \ifpstartnum
1671 \thepstartL
1672 \kern\linenumsep\global\pstartnumfalse\fi
1673 }
1674 \newcommand*\rightpstartnumL){
1675 \ifpstartnum\kern\linenumsep
1676 \thepstartL
1677 \global\pstartnumfalse\fi
1678 }
1679 \newif\ifpstartnumR
1680 \pstartnumRtrue
1681 \newcommand*\leftpstartnumR){
1682 \ifpstartnumR
1683 \thepstartR
1684 \kern\linenumsep\global\pstartnumRfalse\fi
1685 }
1686 \newcommand*\rightpstartnumR){
1687 \ifpstartnumR\kern\linenumsep
1688 \thepstartR
1689 \global\pstartnumRfalse\fi
1690 }
1691 }
1692 %

```

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.

```

1693 \list@create{\inserts@listR}
1694 %

```

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

```

1695 \newcommand*\add@insertsR){%
1696 \global\let\add@inserts@nextR=\relax
1697 \ifx\inserts@listR\empty \else
1698 \ifx\next@insertR\empty
1699 \ifx\insertlines@listR\empty
1700 \global\noteschanged@true
1701 \gdef\next@insertR{100000}%
1702 \else
1703 \gl@p\insertlines@listR\to\next@insertR
1704 \fi
1705 \fi
1706 \ifnum\next@insertR=\absline@numR
1707 \gl@p\inserts@listR\to\@insertR
1708 \@insertR

```

```

1709 \global\let\@insertR=\undefined
1710 \global\let\next@insertR=\empty
1711 \global\let\add@inserts@nextR=\add@insertsR
1712 \fi
1713 \fi
1714 \add@inserts@nextR}
1715
1716 %

```

VIII.7 Penalties

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

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

```

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

```

This is for a single chunk. However, as we are probably dealing with several chunks

at a time, the above is nor 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.

```
1717 \newcommand*{\add@penaltiesL}{%
1718 \newcommand*{\add@penaltiesR}{%
1719
1720 %
```

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.

```
1721 \newcommand*{\flush@notesR}{%
1722   \@xloop
1723   \ifx\inserts@listR\empty \else
1724     \glp\inserts@listR\to\@insertR
1725     \@insertR
1726     \global\let\@insertR=\undefined
1727   \repeat}
1728
1729 %
```

IX Footnotes

IX.1 Footnotes output specific to `\Pages`

`\print@Xnotes@forpages` 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 `\correct@Xfootins@box` `\print@notesX@forpages` `\print@Xnotes@forpages`, which replaces `\print@Xnotes` inside `\Pages`. Here is how we proceed⁶:

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

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

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

```
1730 \newcommand\print@Xnotes@forpages[1]{%
1731 %
```

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

```
1732 \ifcsempy{Xonlyside@#1}{%
1733 \csuse{#1footstart}{#1}%
1734 \csuse{#1footgroup}{#1}%
1735 }%
1736 %
```

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

```
1737 {%
1738 \ifboolexpr{%
1739 ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page}}))%
1740 or%
1741 (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}}))%
1742 }%
1743 %
```

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

```
1744 {%
1745 \csuse{#1footstart}{#1}%
1746 \csuse{#1footgroup}{#1}%
1747 %
```

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`.

```
1748 \global\count\csuse{#1footins}=\z@%
1749 \global\skip\csuse{#1footins}=\z@%
1750 \global\dimen\csuse{#1footins}=\maxdimen%
1751 }%
1752 %
```

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

```

1753 {%
1754   \global\count\csuse{#1footins}=\csuse{default@#1footins}%
1755   \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
1756   \bgroup%
1757     \csuse{Xnotefontsize@#1}%
1758     \global\dimen\csuse{#1footins}=\csuse{Xmaxhnotes@#1}%
1759   \egroup%
1760 %

```

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

```

1761   \global\setnamebox{#1footins@kept}=\box\namebox{#1footins}%
1762 }%
1763 %

```

End of \print@Xnotes@forpages.

```

1764 }%
1765 }%
1766 %

```

And now, the same for familiar footnotes.

```

1767 \newcommand\print@notesX@forpages[1]{%
1768   \ifcempty{onlysideX@#1}{%
1769     \csuse{footstart#1}{#1}%
1770     \csuse{footgroup#1}{#1}%
1771   }%
1772   {%
1773     \ifboolexpr{%
1774       ((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page
1775       }%
1776       (test {\ifcsstring{onlysideX@#1}{R}} and test{\ifnumodd{\c@page}}))%
1777     }%
1778     {%
1779       \csuse{footstart#1}{#1}%
1780       \csuse{footgroup#1}{#1}%
1781       \global\count\csuse{footins#1}=\z@%
1782       \global\skip\csuse{footins#1}=\z@%
1783       \global\dimen\csuse{footins#1}=\maxdimen%
1784     }%
1785     {%
1786       \global\count\csuse{footins#1}=\csuse{default@footins#1}%
1787       \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
1788       \bgroup%
1789         \csuse{Xnotefontsize@#1}%
1790         \global\dimen\csuse{footins#1}=\csuse{maxhnotesX@#1}%
1791       \egroup%
1792     }%

```

```

1793 \global\setnamebox{footins#1@kept}=\box\namebox{footins#1}%
1794 }%
1795 }%
1796 }%
1797 %

```

`\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.

```

1798 \newcommand{\insert@notes@for@onlyside}{%
1799 \def\do##1{%
1800 \unless\ifnocritical@%
1801 \ifvoid\csuse{##1footins@kept}\else%
1802 \expandafter\insert\csname ##1footins\endcsname%
1803 \bgroup%
1804 \unvnamebox{##1footins@kept}%
1805 \egroup%
1806 \fi%
1807 \fi%
1808 \unless\ifnofamiliar@%
1809 \ifvoid\csuse{footins##1@kept}\else%
1810 \expandafter\insert\csname footins##1\endcsname%
1811 \bgroup%
1812 \unvnamebox{footins##1@kept}%
1813 \egroup%
1814 \fi%
1815 \fi%
1816 }%
1817 \dolistloop{\@series}%
1818 }%
1819 %

```

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
1820 \newcount\edtext@now%
1821 \newcount\edtext@later%
1822 %

1823 \newcommand{\edtextlater}[2]{%#1 lemma, #2 critical notes
1824 %

```

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

```

1825 \global\advance\edtext@later by \@ne%
1826 %

```

```

1827 % 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}.
1828 \advance\@edtext@level by \@ne%
1829 \ifcsvoid{sw@list@edtext@\the\@edtext@level}%
1830     {\global\let\sw@inthisedtext\empty}%
1831     {\expandafter\gl@p\csname sw@list@edtext@\the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1832 %

```

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

```

1833 %
1834 \csxdef{edtext@later@\the\edtext@later}{%
1835 %

```

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

```

1836 \noexpand\edtext{%
1837 }%
1838 %

```

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

```

1839 {%
1840 \unexpanded{%
1841 \lemma{\no@expands #1}%
1842 }%
1843 \noexpand\xxref{start:edtext:later:\the\edtext@later}{end:edtext:
later:\the\edtext@later}%
1844 \noexpand\linenum{||||\edfont@info}%
1845 \unexpanded{\def\sw@inthisedtext}{\expandonce{\sw@inthisedtext}}%
1846 %

```

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.

```

1847 \ifledRcol%
1848 \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@true}}%
1849 \else%
1850 \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@false}}%
1851 \fi%
1852 %

```

And the footnote command of this `\edtextlater`.

```

1853 \unexpanded{#2}%
1854 }%
1855 }%
1856 %

```

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`


```

1857 \edlabel{start:edtext:later:\the\edtext@later}%
1858 \flag@start@later%
1859 \bgroup%
1860 \def\lemma##1{%
1861     \ifledRcol%
1862         \write\linenum@outR{\string\@lemma}%
1863     \else%
1864         \write\linenum@out{\string\@lemma}%
1865     \fi%
1866 }%
1867 \renewcommand{\do}[1]{\expandafter\renewcommandx\csname ##1footnote\
endcsname[2][1,usedefault]{}\unskip}%\unskip because of a spurious space in
\newcommandx
1868 \dolistloop{\@series}%
1869 #2%
1870 \egroup%
1871 \showlemma{#1}%
1872 \edlabel{end:edtext:later:\the\edtext@later}%
1873 \flag@end@later%
1874 %

```

We decrease the counter increased at the beginning.

```

1875 \advance\@edtext@level by -\@ne%
1876 }%
1877 %

```

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

```

1878 \newcommand{\edtextnow}[0]{%
1879     \global\advance\edtext@now by \@ne
1880     \csuse{edtext@later@\the\edtext@now}%
1881 }%
1882 %

```

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.

```

1883 \list@create{\labelref@listR}
1884
1885 %

```

\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.

```

1886 \def\l@make@labelsR#1|#2|#3|#4|#5{%
1887   \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
else
1888     \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
1889   \fi
1890   \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
{#1|#2|#3|#4|\@Rlineflag}%
1891   \global\providetoggle{label@#5@ledRcol}%False is the default value of
this toggle, which tell us if a label is linked to a right or a left side
1892   \global\toggletrue{label@#5@ledRcol}%
1893   \ignorespaces}
1894 \AtBeginDocument{%
1895   \def\l@make@labelsR#1|#2|#3|#4|#5{%
1896   }
1897
1898 %

```

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

It is defined on reledmac.

XI Side notes

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

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

```

\sidenotemargin*
\sidenotemarginR
1899 \WithSuffix\newcommand\sidenotemargin*[1]{%
1900   \l@dgetsidenote@margin{#1}
1901   \global\sidenote@marginR=\@l@tempcntb
1902   \global\sidenote@margin=\@l@tempcntb
1903 }
1904 \newcommand{\sidenotemarginR}[1]{%
1905   \l@dgetsidenote@margin{#1}%
1906   \global\sidenote@marginR=\@l@tempcntb%
1907 }%
1908 \newcount\sidenote@marginR
1909 \global\sidenote@marginR=\@ne%
1910
1911 \ifmovecolumnspostiononrightpage%
1912   \sidenotemargin{inner}%
1913   \sidenotemarginR{outer}%
1914 \fi%
1915
1916 %

```

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

```

1917 \newcommand*{\affixside@noteR}{%
1918   \prepare@edindex@for@note{\the\page@numR|\the\line@numR|\the\
subline@numR|\the\page@numR|\the\line@numR|\the\subline@numR|}%
1919   \def\sidenotecontent@{%
1920     \numgdef{\itemcount@}{0}%
1921     \def\do##1{%
1922       \ifnumequal{\itemcount@}{0}%
1923         {%
1924           \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
1925         {\appto\sidenotecontent@{\sidenotessep ##1}%
1926         }%
1927       \numgdef{\itemcount@}{\itemcount@+1}%
1928     }%
1929     \dolistloop{\l@dcsnotetext}%
1930     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
1931   \gdef\@templ@d{%
1932     \gdef\@templ@n{\l@dcsnotetext\l@dcsnotetext@l\l@dcsnotetext@r}%
1933     \ifx\@templ@d\@templ@n \else%
1934       \if@twocolumn%
1935         \if@firstcolumn%
1936           \setl@dlp@rbox{##1}{\sidenotecontent@}%
1937         \else%
1938           \setl@drp@rbox{\sidenotecontent@}%
1939         \fi%
1940       \else%
1941         \l@dttempcntb=\sidenote@marginR%
1942         \ifnum\l@dttempcntb>\@ne%
1943           \advance\l@dttempcntb by\page@numR%
1944         \fi%
1945         \ifodd\l@dttempcntb%
1946           \setl@drp@rbox{\sidenotecontent@}%
1947           \gdef\sidenotecontent@{%
1948             \numdef{\itemcount@}{0}%
1949             \dolistloop{\l@dcsnotetext@l}%
1950             \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
1951             \setl@dlp@rbox{\sidenotecontent@}%
1952           \else%
1953             \setl@dlp@rbox{\sidenotecontent@}%
1954             \gdef\sidenotecontent@{%
1955               \numdef{\itemcount@}{0}%
1956               \dolistloop{\l@dcsnotetext@r}%
1957               \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
1958               \setl@drp@rbox{\sidenotecontent@}%
1959             \fi%
1960           \fi%
1961         \fi%
1962         \advance\@edindex@for@note@m@ne%

```

```

1963 }
1964
1965 %

```

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.

```

\inserthangingsymbolL966 \newif\ifinserthangingsymbolR
\inserthangingsymbolR967 \newcommand{\inserthangingsymbolL}{%
1968   \ifinserthangingsymbol%
1969     \ifinstanzaL%
1970       \hskip \ifundefined{sza@00}{0}{\expandafter%
1971         \noexpand\csname sza@00\endcsname}\stanzaindentbase%
1972       \@hangingsymbol%
1973     \fi%
1974   \fi%
1975 }%
1976 \newcommand{\inserthangingsymbolR}{%
1977   \ifinserthangingsymbolR%
1978     \ifinstanzaR%
1979       \hskip \ifundefined{sza@00}{0}{\expandafter%
1980         \noexpand\csname sza@00\endcsname}\stanzaindentbase%
1981       \@hangingsymbol%
1982     \fi%
1983   \fi%
1984 }%
1985 %

```

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.

```

1986 \chardef\next=\catcode`&
1987 \catcode`&=\active
1988
1989 %

```

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

```

1990 \newenvironment{astanza}[1][\]{%
1991   \catcode`&\active
1992   \global\stanza@count\ne\stanza@modulo\ne
1993   \ifnum\usernamecount{sza@00}=\z@
1994     \let\stanza@hang\relax
1995     \let\endlock\relax

```

```

1996 \else
1997   \rightskip\z@ plus 1fil\relax
1998 \fi
1999 \ifnum\usernamecount{szp@0@}=\z@
2000   \let\sza@penalty\relax
2001 \fi
2002 \def&{%
2003   \endlock\mbox{}%
2004   \sza@penalty
2005   \global\advance\stanza@count\@ne
2006   \@astanza@line}%
2007 \def\&{\@stopastanza}%
2008 \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}} and test{\ifstrempy
2009   {\pstart[\at@every@stanza]}}}%
2010   {\pstart[#1]}}%
2011   \@astanza@line
2012   \let\par\relax\ignorespaces%No paragraph in verses
2013 }{}
2014
2015 %

```

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

```

2016 \newcommandx{\@stopastanza}[1][1,usedefault]{%
2017   \endlock\mbox{}%
2018   \ifboolexpr{not test{\ifdefvoid{\at@every@stop@stanza}} and test{\
2019     ifstrempy{#1}}}%
2020     {\pend[\at@every@stop@stanza]}}%
2021     {\pend[#1]}}%
2022 }%

```

\@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.

```

2023 \newcommand*{\@astanza@line}{%
2024   \ifnum\value{stanzaindentrepetition}=0
2025     \ifcsdef{sza@\number\stanza@count @}{%
2026       {%
2027         \parindent=\csname sza@\number\stanza@count @\endcsname\
2028         stanzaindentbase%
2029       }{%
2030         \led@err@StanzaIndentNotDefined%
2031       }%
2032     }%
2033   \else
2034     \ifcsdef{sza@\number\stanza@modulo @}{%
2035       \parindent=\csname sza@\number\stanza@modulo @\endcsname\
2036       stanzaindentbase%
2037     }%
2038     \managestanza@modulo%

```

```

2035     }%
2036     {%
2037     \led@err@StanzaIndentNotDefined%
2038     }%
2039     \fi
2040     \endgraf
2041     \stanza@hang%
2042     \ignorespaces}
2043
2044 %

```

Lastly reset the modified category codes.

```

2045 \catcode`\&=\next
2046
2047 %

```

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

```

\thestanzaR
2048 \newcounter{stanzaL}
2049 \newcounter{stanzaR}
2050 \renewcommand{\thestanzaL}{%
2051   \textbf{\arabic{stanzaL}}%
2052 }
2053 \renewcommand{\thestanzaR}{%
2054   \textbf{\arabic{stanzaR}}%
2055 }
2056 %
2057 %

```

XIII Fixing babel and polyglossia

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

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

```

\ifl@dusedbabel A flag for checking if babel has been used as a package.
\l@dusedbabelfalse
\l@dusedbabeltrue
2058 \newif\ifl@dusedbabel
2059 %

```

`\l@dcchecklang`

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

```
2060 \patchcmd{\bbl@set@language}%
2061   {\select@language{\language\language}}%
2062   {\edef\language{#1}\select@language{\language}}%
2063   {}%
2064   {}%
2065
2066 %
```

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` `\l@duselanguage` `\theledlanguageL` `\theledlanguageR`
`\l@duselanguage` are the names of the languages of the left and right texts. `\l@duselanguage` is similar
`\theledlanguageL` to `\selectlanguage`.
`\theledlanguageR`

```
2067 \newcommand*{\l@duselanguage}[1]{%
2068   \gdef\theledlanguageL{#1}%
2069   \gdef\theledlanguageR{#1}%
2070
2071 %
```

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

```
2072 \AtBeginDocument{%
2073   \ifundefined{xpg@main@language}{%
2074     \ifundefined{bbl@main@language}{%
2075       %
```

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

```
2076   \l@dusedbabelfalse
2077   }{%
2078   %
```

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.

```
2079   \l@dusedbabeltrue
2080   \let\l@doldselectlanguage\selectlanguage
2081   \let\l@doldbbl@set@language\bbl@set@language
2082   \renewcommand{\selectlanguage}[1]{%
2083     \l@doldselectlanguage{#1}%
```

```

2084 \ifledRcol \gdef\theledlanguageR{#1}%
2085 \else      \gdef\theledlanguageL{#1}%
2086 \fi}
2087 %

```

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

```

2088 \renewcommand*{\l@duselanguage}[1]{%
2089   \expandafter\l@doldselectlanguage\expandafter{#1}}
2090 %

```

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

```

2091 \gdef\theledlanguageL{\bbl@main@language}%
2092 \gdef\theledlanguageR{\bbl@main@language}%
2093 }%
2094 }
2095 %

```

If use polyglossia

```

2096 { \let\old@otherlanguage\otherlanguage%
2097   \renewcommand{\otherlanguage}[2][]{%
2098     \selectlanguage{#1}{#2}%
2099     \ifledRcol \gdef\theledlanguageR{#2}%
2100     \else      \gdef\theledlanguageL{#2}%
2101     \fi}%
2102   \renewcommand{\l@duselanguage}[1]{%
2103     \csuse{no\language@name @numbers}\select@language{#1}%
2104   }%
2105   \gdef\theledlanguageL{\xpg@main@language}%
2106   \gdef\theledlanguageR{\xpg@main@language}%
2107 %

```

That is it.

```

2108 }}
2109 %

```

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
\l@dc@maxchunks 5120 chunk pairs.

```

2110 \newcount\l@dc@maxchunks
2111 \newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}

```



```

2112 \maxchunks{5120}
2113
2114 %

```

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

```

\l@dnumpstartsR
2115 \newcount\l@dnumpstartsR
2116
2117 %

```

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

```

\l@pscR
2118 \newcount\l@dpscL
2119 \newcount\l@dpscR
2120
2121 %

```

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

```

2122 \newcommand*{\l@dsetuprawboxes}{%
2123 \l@l@tempcntb=\l@dc@maxchunks
2124 \loop\ifnum\l@l@tempcntb>\z@
2125 \newnamebox{\l@dLcolrawbox\the\l@l@tempcntb}
2126 \newnamebox{\l@dRcolrawbox\the\l@l@tempcntb}
2127 \advance\l@l@tempcntb \m@ne
2128 \repeat}
2129
2130 %

```

\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.

```

2131 \newcommand*{\l@dsetupmaxlinecounts}{%
2132 \l@l@tempcntb=\l@dc@maxchunks
2133 \loop\ifnum\l@l@tempcntb>\z@
2134 \newnamecount{\l@dmaxlinesinpar\the\l@l@tempcntb}
2135 \advance\l@l@tempcntb \m@ne
2136 \repeat}
2137 \newcommand*{\l@dzeromaxlinecounts}{%
2138 \begingroup
2139 \l@l@tempcntb=\l@dc@maxchunks
2140 \loop\ifnum\l@l@tempcntb>\z@
2141 \global\usenamecount{\l@dmaxlinesinpar\the\l@l@tempcntb}=\z@
2142 \advance\l@l@tempcntb \m@ne
2143 \repeat
2144 \endgroup}
2145
2146 %

```

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

```

2147 \AtBeginDocument{%
2148   \l@dssetuprawboxes
2149   \l@dssetupmaxlinecounts
2150   \l@dszeromaxlinecounts
2151   \l@dnumpstartsL=\z@
2152   \l@dnumpstartsR=\z@
2153   \l@dpscL=\z@
2154   \l@dpscR=\z@}
2155
2156 %

```

XV Checking text to be processed

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

`\@pstartstrue`

`\@pstartsfalse`

`\check@pstarts`

```

2157 \newif\if@pstarts
2158 \newcommand*{\check@pstarts}{%
2159   \@pstartsfalse
2160   \ifnum\l@dnumpstartsL>\l@dpscL
2161     \@pstartstrue
2162   \else
2163     \ifnum\l@dnumpstartsR>\l@dpscR
2164       \@pstartstrue
2165     \fi
2166   \fi
2167 }
2168
2169 %

```

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

`\araw@texttrue`

`\araw@textfalse`

`\checkraw@text`

```

2170 \newif\ifaraw@text
2171 \newcommand*{\checkraw@text}{%
2172   \araw@textfalse
2173   \ifvbox\namebox{\l@dLcolrawbox\the\l@dpscL}
2174     \araw@texttrue
2175   \else
2176     \ifvbox\namebox{\l@dRcolrawbox\the\l@dpscR}
2177       \araw@texttrue
2178     \fi
2179   \fi
2180 }
2181
2182 %

```

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

```

2183 \newcommand*{\@writelinesinparL}{%
2184   \edef\next{%
2185     \write\linenum@out{\string\@pend[\the\@donereallinesL]}}%
2186   \next
2187   \global\@donereallinesL \z@}
2188 \newcommand*{\@writelinesinparR}{%
2189   \edef\next{%
2190     \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}%
2191   \next
2192   \global\@donereallinesR \z@}
2193
2194 %

```

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

```

\@writepageofparR
2195 \newcommand*{\@writepageofparL}[0]{%
2196   \ifnum\@donereallinesL=\z@%
2197     \edef\next{%
2198       \write\linenum@out{\string\@pstart{\the\l@dpscl}{\the\c@page}{\the\
2199         numpagelinesL}}%
2200     }%
2201     \next%
2202   \fi%
2203 }%
2204 \newcommand*{\@writepageofparR}[0]{%
2205   \ifnum\@donereallinesR=\z@%
2206     \edef\next{%
2207       \write\linenum@outR{\string\@pstartR{\the\l@dpscR}{\the\c@page}{\the\
2208         numpagelinesR}}%
2209     }%
2210     \next%
2211   \fi%
2212 }%
2213 %
2214 %

```

XVI Parallel columns

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

```

\@eledsectionR
2212 \newsavebox{\@eledsectionL}%
2213 \newsavebox{\@eledsectionR}%
2214 %

```

`\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.

```

2215 \newcommand*{\Columns}{%
2216   \ifl@dpairing%
2217     \led@err@Columns@InsideEnv%
2218   \fi%
2219   \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{
l@dLcolrawbox1}}}{%
2220     \ifcsboxvoid{l@dRcolrawbox1}{%
2221       \ifcsboxvoid{l@dLcolrawbox1}%
2222       {\led@err@Columns@WithoutEnv}%
2223       {\led@err@Columns@WithoutRightside}%
2224     }%
2225     {\led@err@Columns@WithoutLeftside}%
2226   }{%
2227     \global\l@dprintingcolumnstrue%
2228     \eledsection@correcting@skip=-\baselineskip% Correction for sections'
titles
2229     \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else
2230       \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
2231     \fi
2232   }%

```

Start a group and zero counters, etc.

```

2233 \begingroup
2234   \l@dzeropenalties
2235   \endgraf\global\num@lines=\prevgraf
2236   \global\num@linesR=\prevgraf
2237   \global\par@line=\z@
2238   \global\par@lineR=\z@
2239   \global\l@dpscL=\z@
2240   \global\l@dpscR=\z@
2241   \get@familiarfootnote@number%
2242 %

```

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

```

2243 \check@pstarts
2244 \loop\if@pstarts
2245   \global\pstartnumtrue
2246   \global\pstartnumRtrue
2247 %

```

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

```

2248 \global\advance\l@dpscL \@ne
2249 \global\advance\l@dpscR \@ne
2250 \restore@pstartL@pc%
2251 \restore@pstartR@pc%
2252 %

```

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

```

2253 \Columns@print@before@pstart%
2254 %

```

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

```

2255 \checkraw@text
2256 { \loop\ifaraw@text
2257 %

```

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

```

2258 \l@duselanguage{\theledlanguageL}%
2259 \do@lineL
2260 \xifinlist{\the\l@dpscl}{\eled@sections@@}
2261 {%
2262 \ifdefstring{\@eledsectmark}{L}%
2263 {\csuse{eled@sectmark@\the\l@dpscl}%
2264 }{}%
2265 \global\csundef{eled@sectmark@\the\l@dpscl}%
2266 \savebox{\@eledsectionL}{\parbox[t]{}[t]{\Lcolwidth}{\vbox
2267 {} \print@eledsectionL}}}%\vbox{}-> prevent alignment troubles with RTL
2268 language
2269 {%
2270 \l@duselanguage{\theledlanguageR}%
2271 \do@lineR
2272 \xifinlist{\the\l@dpscr}{\eled@sectionsR@@}
2273 {%
2274 \ifdefstring{\@eledsectmark}{R}%
2275 {\csuse{eled@sectmark@\the\l@dpscr R}%
2276 }{}%
2277 \global\csundef{eled@sectmark@\the\l@dpscr R}%
2278 \savebox{\@eledsectionR}{\parbox[t]{}[t]{\Rcolwidth}{\vbox
2279 {} \print@eledsectionR}}}%\vbox{}-> prevent alignment troubles with RTL
2280 language
2281 {%
2282 \hb@xt@ \hsize{%
2283 \ifdefstring{\columns@position}{L}{\hfill }%
2284 \print@leftcolumn%
2285 \print@columnseparator%
2286 \print@rightcolumn%
2287 \ifdefstring{\columns@position}{R}{\hfill }%
2288 }%
2289 \checkraw@text
2290 \checkverseL
2291 \checkverseR
2292 \checkpb@columns
2293 \repeat}
2294 %

```

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.

```

2292 \@writelinesinparL
2293 \@writelinesinparR
2294 \check@pstarts
2295 \ifbypstart@%
2296 \write\linenum@out{\string\@set[1]}
2297 \resetprevline@
2298 \fi
2299 \ifbypstart@R
2300 \write\linenum@outR{\string\@set[1]}
2301 \resetprevline@
2302 \fi
2303 \Columns@print@after@pend%
2304 \repeat
2305 %

```

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.

```

2306 \flush@notes
2307 \flush@notesR
2308 \endgroup
2309 %

2310 \global\l@dpscl=\z@
2311 \global\l@dpscR=\z@
2312 \global\l@dnpstartsL=\z@
2313 \global\l@dnpstartsR=\z@
2314 \global\l@dprintingcolumnsfalse%
2315 \ignorespaces
2316 \global\instanzaLfalse%
2317 \global\instanzaRfalse%
2318 }%
2319 }%
2320
2321 %

```

\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).

```

2322 \def\print@columnseparator{%
2323 \ifdim\beforecolumnseparator<0pt%
2324 \hfill%
2325 \else%
2326 \hspace{\beforecolumnseparator}%
2327 \fi%
2328 \columnseparator%

```

```

2329 \ifdim\aftercolumnseparator<0pt%
2330 \hfill%
2331 \else%
2332 \hspace{\aftercolumnseparator}%
2333 \fi%
2334 }%
2335 %

```

`\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 `\movecolumnsonrightpage` is set to TRUE, it also depends on the page number.

```

2336 \newcommand{\print@leftcolumn}{%
2337 \if\page@num>\page@numR%
2338 \l@tempcntb=\page@num%
2339 \else%
2340 \l@tempcntb=\page@numR%
2341 \fi%
2342 \ifboolexpr{%
2343 not bool {movecolumnsonrightpage}%
2344 or test {\ifnumodd{\l@tempcntb}}}%
2345 }%
2346 {%
2347 \unhbox\l@leftbox%
2348 \ifhbox\@eledsectionL%
2349 \usebox{\@eledsectionL}%
2350 \fi%
2351 }%
2352 {%
2353 \unhbox\l@rightbox%
2354 \ifhbox\@eledsectionR%
2355 \usebox{\@eledsectionR}%
2356 \fi%
2357 }%
2358 }%
2359
2360
2361 \newcommand{\print@rightcolumn}{%
2362 \if\page@num>\page@numR%
2363 \l@tempcntb=\page@num%
2364 \else%
2365 \l@tempcntb=\page@numR%
2366 \fi%
2367 \ifboolexpr{%
2368 not bool {movecolumnsonrightpage}%
2369 or test {\ifnumodd{\l@tempcntb}}}%
2370 }%
2371 {%
2372 \unhbox\l@rightbox%

```

```

2373 \ifhbox\@eledsectionR%
2374 \usebox{\@eledsectionR}%
2375 \fi%
2376 }%
2377 {%
2378 \unhbox\l@dleftbox%
2379 \ifhbox\@eledsectionL%
2380 \usebox{\@eledsectionL}%
2381 \fi%
2382 }%
2383 }%
2384 %

```

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

```

2385
2386 \newcommand{\checkpb@columns}{%
2387 \newif\if@pb
2388 \newif\if@nopb
2389 \IfStrEq{\led@pb@setting}{before}{
2390 \numdef{\next@absline}{\the\absline@num+1}%
2391 \numdef{\next@abslineR}{\the\absline@numR+1}%
2392 \xifinlistcs{\next@absline}{l@prev@pb}{\@pbtrue}{}%
2393 \xifinlistcs{\next@abslineR}{l@prev@pbR}{\@pbtrue}{%
2394 \xifinlistcs{\next@absline}{l@prev@nopb}{\@nopbtrue}{}%
2395 \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\@nopbtrue}{%
2396 }{}
2397 \IfStrEq{\led@pb@setting}{after}{
2398 \xifinlistcs{\the\absline@num}{l@prev@pb}{\@pbtrue}{}%
2399 \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\@pbtrue}{%
2400 \xifinlistcs{\the\absline@num}{l@prev@nopb}{\@nopbtrue}{}%
2401 \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\@nopbtrue}{%
2402 }{}
2403 \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
2404 \if@pb\pagebreak[4]\fi
2405 }
2406 %

```

\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).

```

2407 \newcommand*{\columnseparator}{%
2408 \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}}
2409 \newdimen\columnrulewidth
2410 \columnrulewidth=\z@
2411
2412 %

```


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

```
2413 \newcommand*\columnspan[1]{%
2414   \xdef\columns@position{#1}%
2415 }%
2416 \xdef\columns@position{R}%
2417 %
```

`\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`.

```
2418 \newlength{\beforecolumnseparator}%
2419 \setlength{\beforecolumnseparator}{-2pt}%
2420
2421 \newlength{\aftercolumnseparator}%
2422 \setlength{\aftercolumnseparator}{-2pt}%
2423
2424 %
```

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

```
2425 \newcommand*\setwidthliketwocolumns@L{%
2426   % Temporary dimension, initially equal to the standard hsize, i.e. text
width
2427   % \begin{macrocode}
2428   \newdimen\temp%
2429   \temp=\hsize%
2430   %
```

Hsize : Left + Right width

```
2431   \hsize=\Lcolwidth%
2432   \advance\hsize\Rcolwidth%
2433   %
```

Now, calculating the remaining space

```
2434   \advance\temp-\hsize%
2435   %
```

And multiply the hsize by 2/3 of this space

```
2436   \multiply\temp by 2%
2437   \divide\temp by 3%
2438   \advance\hsize\temp%
2439 }%
```

```

2440 \newcommand{\setpositionliketwocolumns@L}{%
2441 \renewcommand{\ledrlfill}{\hfill}%
2442 }%
2443
2444 \newcommand{\setnotespositionliketwocolumns@L}{%
2445 }%
2446
2447
2448
2449 %

```

```

2450 \newcommand{\setwidthliketwocolumns@C}{%
2451 % Temporary dimension, initially equal to the standard hsize, i.e. text
2452 % width
2453 }%

```

```

2453 \newdimen\temp%
2454 \temp=\hsize%
2455 % Hsize : Left + Right width
2456 %

```

```

2457 \hsize=\Lcolwidth%
2458 \advance\hsize\Rcolwidth%
2459 % Now, calculating the remaining space
2460 %

```

```

2461 \advance\temp-\hsize%
2462 %

```

And multiply the hsize by 1/2 of this space

```

2463 \divide\temp by 2%
2464 \advance\hsize\temp%
2465 }%
2466
2467 \newcommand{\setpositionliketwocolumns@C}{%
2468 \doinsidelinehook{\hfill}%
2469 \renewcommand{\ledrlfill}{\hfill}%
2470 }%
2471
2472 \newcommand{\setnotespositionliketwocolumns@C}{%
2473 \newdimen\temp%
2474 \newdimen\tempa%
2475 \temp=\hsize%
2476 \tempa=\Lcolwidth%
2477 \advance\tempa\Rcolwidth%
2478 \advance\temp-\tempa%
2479 \divide\temp by 2%
2480 \leftskip=\temp%
2481 \rightskip=-\temp%
2482 }%

```

```

2483 \newcommand{\setwidthliketwocolumns@R}{%
2484 %
2485 %

```

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

```

2486 \newdimen\temp%
2487 \temp=\hsize%
2488 %

```

Hsize : Left + Right width

```

2489 \hsize=\Lcolwidth%
2490 \advance\hsize\Rcolwidth%
2491 %

```

Now, calculating the remaining space

```

2492 \advance\temp-\hsize%
2493 %

```

And multiply the hsize by 2/3 of this space

```

2494 \multiply\temp by 2%
2495 \divide\temp by 3%
2496 \advance\hsize\temp%
2497 }%
2498
2499 \newcommand{\setpositionliketwocolumns@R}{%
2500 \doinsidelinehook{\hfill}%
2501 }%
2502
2503 \newcommand{\setnotespositionliketwocolumns@R}{%
2504 \newdimen\temp%
2505 \newdimen\tempa%
2506 \temp=\hsize%
2507 \tempa=\Lcolwidth%
2508 \advance\tempa\Rcolwidth%
2509 \advance\temp-\tempa%
2510 \divide\temp by 2%
2511 \leftskip=\temp%
2512 \rightskip=-\temp%
2513 }%
2514
2515 %

```

`\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.

```

2516 \newcommand{\Columns@print@before@pstart}{%
2517 \ifboolexpr{%
2518 test{\ifcsstring{before@pstartL@the\l@dpscl}{\at@every@pstart}}}%

```

```

2519 and test {\ifcsstring{before@pstartR@the\l@dpscR}{\at@every@pstart}}%
2520 and test {\ifdefempty{\at@every@pstart}}}%
2521 {}%
2522 {%
2523   \hb@xt@ \hsize{%
2524     \ifdefstring{\columns@position}{L}{\hfill }%
2525     \par\parbox[t] [] [t]{\Lcolwidth}{%
2526       \csuse{before@pstartL@the\l@dpscL}%
2527     }%
2528     \print@columnseparator%
2529     \parbox[t] [] [t]{\Rcolwidth}{%
2530       \set@sectcountR%
2531       \csuse{before@pstartR@the\l@dpscR}%
2532     }%
2533     \ifdefstring{\columns@position}{R}{\hfill}%
2534   }%
2535 }%
2536 \global\csundef{before@pstartL@the\l@dpscL}%
2537 \global\csundef{before@pstartR@the\l@dpscR}%
2538 }%
2539 \newcommand{\Columns@print@after@pend}{%
2540   \ifboolexpr{%
2541     test{\ifcsstring{after@pendL@the\l@dpscL}{\at@every@pend}}%
2542     and test {\ifcsstring{after@pendR@the\l@dpscR}{\at@every@pend}}%
2543     and test {\ifdefempty{\at@every@pend}}}%
2544   {%
2545     {%
2546       \hb@xt@ \hsize{%
2547         \ifdefstring{\columns@position}{L}{\hfill }%
2548         \parbox[t] [] [t]{\Lcolwidth}{%
2549           \csuse{after@pendL@the\l@dpscL}%
2550         }%
2551         \print@columnseparator%
2552         \parbox[t] [] [t]{\Rcolwidth}{%
2553           \set@sectcountR%
2554           \csuse{after@pendR@the\l@dpscR}%
2555         }%
2556         \ifdefstring{\columns@position}{R}{\hfill}%
2557       }%
2558     }%
2559     \global\csundef{after@pendL@the\l@dpscL}%
2560     \global\csundef{after@pendR@the\l@dpscR}%
2561   }%
2562 }%

```

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
`\numpagelinesR` lines on a pair of facing pages.
`\l@dminpagelines`

```

2563 \newcount\numpagelinesL
2564 \newcount\numpagelinesR
2565 \newcount\l@dminpagelines
2566
2567 %

```

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.

```

2568 \newcommand*{\Pages}[1][1,usedefault]{%
2569   \ifl@dpairing%
2570     \led@err@Pages@InsideEnv%
2571   \fi%
2572   \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{
l@dLcolrawbox1}}}{%
2573     \ifcsboxvoid{l@dRcolrawbox1}{%
2574       \ifcsboxvoid{l@dLcolrawbox1}%
2575       {\led@err@Pages@WithoutEnv}%
2576       {\led@err@Pages@WithoutRightside}%
2577     }%
2578     {\led@err@Pages@WithoutLeftside}%
2579   }{%
2580     \ifstrequal{#1}{mainmatter}{\Pages@mainmattertrue}{\Pages@mainmatterfalse
}%
2581     \eledsection@correcting@skip=-2\baselineskip% line correcting for section
titles.
2582     \parledgroup@notespacing@set@correction%
2583     \typeout{}%
2584     \typeout{***** PAGES *****}%
2585     \ifnum\l@dnumstartsL=\l@dnumstartsR\else%
2586       \led@err@BadLeftRightPstarts{\the\l@dnumstartsL}{\the\l@dnumstartsR}%
2587     \fi%
2588   }%

```

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

```

2589 \cleartol@devenpage%
2590 \global\l@dprintingpagetrue%
2591 \begingroup%
2592 %

```

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

```

2593 \setlength{\Lcolwidth}{\textwidth}%
2594 \setlength{\Rcolwidth}{\textwidth}%
2595 %

2596 \l@dzeropenalties%
2597 \endgraf\global\num@lines=\prevgraf%
2598 \global\num@linesR=\prevgraf%
2599 \global\par@line=\z@%
2600 \global\par@lineR=\z@%
2601 \global\l@dpscL=\z@%
2602 \global\l@dpscR=\z@%
2603 \writtenlinesLfalse%
2604 \writtenlinesRfalse%
2605 \get@familiarfootnote@number%
2606 %

```

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

```

2607 \let\print@Xnotes\print@Xnotes@forpages%
2608 \let\print@notesX\print@notesX@forpages%
2609 %

```

Check if there are chunks to be processed.

```

2610 \check@pstarts%
2611 \loop\if@pstarts%
2612 %

```

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

```

2613 \global\advance\l@dpscL \@ne%
2614 \global\advance\l@dpscR \@ne%
2615 %

```

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

```

2616 \getlinesfromparlistL%
2617 \getlinesfromparlistR%
2618 \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}%
2619 {\usernamecount{l@dmaxlinesinpar\the\l@dpscL}}%
2620 \check@pstarts%
2621 \repeat%
2622 %

```

Zero the counts again, ready for the next bit.

```

2623 \global\l@dpscL=\z@%
2624 \global\l@dpscR=\z@%
2625 %

```

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

```

2626 \getlinesfrompagelistL%
2627 \getlinesfrompagelistR%
2628 \l@dcalcm@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
2629 {\l@dminpagelines}%
2630 %

```

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.

```

2631 \check@pstarts%
2632 \if@pstarts%
2633 %

```

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

```

2634 \global\advance\l@dpscL \@ne%
2635 \global\advance\l@dpscR \@ne%
2636 \restore@pstartL@pc%
2637 \restore@pstartR@pc%
2638 %

```

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

```

2639 \global\@donereallinesL=\z@%
2640 \global\@donetotallinesL=\z@%
2641 \global\@donereallinesR=\z@%
2642 \global\@donetotallinesR=\z@%
2643 %

```

Start a loop over the boxes (chunks).

```

2644 \checkraw@text%
2645 %
2646 % \begingroup
2647 { \loop\ifaraw@text%
2648 %

```

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

```

2649 \checkpageL%
2650 \l@duselanguage{\theledlanguageL}%
2651 { \loop\ifl@dsamepage%
2652 %

```

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

```

2653 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}%
2654 \csuse{before@pstartL@the\l@dpscL}%
2655 \global\csundef{before@pstartL@the\l@dpscL}%
2656 \do@lineL%
2657 \xifinlist{\the\l@dpscL}{\eled@sections@@}

```

```

2658         {\print@eledsectionL}%
2659         {}%
2660         \advance\numpagelinesL \@ne%
2661 %

```

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

```

2662         \ifshiftedpstarts%
2663             \ifdim\ht\l@dleftbox>0pt%
2664                 \parledgroup@correction@notespacing{L}%
2665                 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2666             \else%
2667                 \unless\ifadvancedshiftedpstarts%
2668                     \dimen0=\pagetotal%
2669                     \advance\dimen0 by \baselineskip%
2670                     \global\pagetotal=\dimen0%
2671                 \else%
2672                     \ifnomaxlines%
2673                         \numdef{\@tmp}{\the\l@dpscL+1}%
2674                         \ifcsdef{minpage@pstart@\@tmp}{%
2675                             \ifnumless{\the\c@page}{\cuse{
minpage@pstart@\@tmp}}}%
2676                             {\dimen0=\pagetotal%
2677                                 \advance\dimen0 by \baselineskip%
2678                                 \global\pagetotal=\dimen0%
2679                                 }%
2680                             {}%
2681                         }{}%
2682                     \fi%
2683                 \fi%
2684             \fi%
2685         \else%
2686             \parledgroup@correction@notespacing{L}%
2687             \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2688         \fi%
2689 %

```

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

```

2690         \get@nextboxL%
2691         \global\l@dskipversenumberfalse%
2692         \ifprint@last@after@pendL%

```



```

2693         \csuse{after@pendL@the\l@dpscL}%
2694         \global\csundef{after@pendL@the\l@dpscL}%
2695         \fi%
2696         \checkpageL%
2697         \checkverseL%
2698         \checkpbL%
2699         \repeat%
2700 %

```

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.

```

2701         \ifl@dpagfull%
2702         \@writelinesonpageL{\the\numpagelinesL}%
2703         \else%
2704         \@writelinesonpageL{1000}%
2705         \fi%
2706 %

```

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.

```

2707         \numpagelinesL \z@%
2708         \parledgroup@correction@notespacing@init%
2709         \clearl@dleftpage }%
2710 %

```

Now do the same for the right text.

```

2711         \checkpageR%
2712         \l@duselanguage{\theledlanguageR}%
2713 {
2714         \loop\ifl@dsamepage%
2715         \set@sectcountR%
2716         \ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{}%
2717         \csuse{before@pstartR@the\l@dpscR}%
2718         \global\csundef{before@pstartR@the\l@dpscR}%
2719         \do@lineR%
2720         \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
2721         {\print@eledsectionR}%
2722         {}%
2723         \advance\numpagelinesR \@ne%
2724         \ifshiftedpstarts%
2725         \ifdim\ht\l@drightbox>0pt%
2726         \parledgroup@correction@notespacing{R}%
2727         \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2728         \else%
2729         \unless\ifadvancedshiftedpstarts%
2730         \dimen0=\pagetotal%
2731         \advance\dimen0 by \baselineskip%

```

```

2731         \global\pagetotal=\dimen0%
2732     \else%
2733         \ifnomaxlines%
2734             \numdef{\@tmp}{\the\l@dpscR+1}%
2735             \ifcsdef{minpage@pstart@\@tmp}{%
2736                 \ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}%
2737                 {\dimen0=\pagetotal%
2738                  \advance\dimen0 by \baselineskip%
2739                  \global\pagetotal=\dimen0%
2740                  }%
2741                 }{}%
2742             }{}%
2743         \fi%
2744     \fi%
2745 \fi%
2746 \else%
2747     \parledgroup@correction@notespacing{R}%
2748     \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2749 \fi%
2750 \get@nextboxR%
2751 \global\l@dskipversenumberRfalse%
2752 \ifprint@last@after@pendR%
2753     \csuse{after@pendR@\the\l@dpscR}%
2754 \global\csundef{after@pendR@\the\l@dpscR}%
2755 \fi%
2756 \checkpageR%
2757 \checkverseR%
2758 \checkpbR%
2759 \repeat%
2760 \ifl@dpagefull%
2761     \@writelinesonpageR{\the\numpagelinesR}%
2762 \else%
2763     \@writelinesonpageR{1000}%
2764 \fi%
2765 \numpagelinesR=\z@%
2766 \parledgroup@correction@notespacing@init%
2767 %

```

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

```

2768     \clearl@drightpage}%
2769 %

```

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.

```

2770     \checkraw@text%
2771 \ifaraw@text%
2772     \getlinesfrompagelistL%
2773     \getlinesfrompagelistR%
2774     \l@dcalc@minoftwo{\cs@linesonpageL}{\cs@linesonpageR}%

```

```

2775             {\l@dminpagelines}%
2776         \fi%
2777     \repeat}%
2778 %

```

We have now output the text from all the chunks.

```

2779     \fi%
2780 %

```

Make sure that there are no inserts hanging around.

```

2781     \flush@notes%
2782     \flush@notesR%
2783     \endgroup%
2784 %

```

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

```

2785     \global\l@dpscL=\z@%
2786     \global\l@dpscR=\z@%
2787     \global\l@dnumstartsL=\z@%
2788     \global\l@dnumstartsR=\z@%
2789     \global\instanzaLfalse%
2790     \global\instanzaRfalse%
2791     \global\l@dprintingpagesfalse%
2792 %

```

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

```

2793     \ifnum\edtext@later=\edtext@now%
2794     \else%
2795         \led@error@edtext@later@now%
2796     \fi%
2797     \global\edtext@later=\z@%
2798     \global\edtext@now=\z@%
2799 %

```

Prevent final notes from overlapping the line number

```

2800     \finish@Pages@notes%
2801     \ignorespaces}}}%
2802
2803
2804 %

```

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.

```

2805 \newcommand{\finish@Pages@notes}{%
2806     \def\do##1{%
2807 %

```

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

```

2808 \ifnocritical@%
2809 \global\newnamebox{##1footins}%
2810 \fi
2811 \ifnofamiliar@%
2812 \global\newnamebox{footins##1}%
2813 \fi
2814 %

```

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

```

2815 \ifvoid\csuse{##1footins}%
2816 \ifvoid\csuse{footins##1}\else%
2817 \newpage\null%
2818 \listbreak%
2819 \fi%
2820 \else%
2821 \newpage\null%
2822 \listbreak%
2823 \fi%
2824 }%
2825 \dolistloop{@series}%
2826 }%
2827 %

```

XVII.4 Struts

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

```

\ledstrutR
2828 \newcommand*{\ledstrutL}{}
2829 \newcommand*{\ledstrutR}{}
2830
2831 %

```

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.

```

2832 \providecommand{\cleartoevenpage}[1][\@empty]{%
2833 \clearpage
2834 \ifodd\c@page\hbox{##1}\clearpage\fi}
2835
2836 \newcommand*{\cleartol@devenpage}{%
2837 \ifdim\pagetotal<\topskip% on an empty page
2838 \else
2839 \clearpage

```

```

2840 \Pages@mainmatter%
2841 \fi
2842 \ifodd\c@page%
2843 \ifprevpgnotnumbered%
2844 \addtocounter{par@page}{-1}%
2845 \ifdef{\prevpgstyle}{\thispagestyle{\prevpgstyle}}{}%
2846 \fi%
2847 \hbox{}\clearpage%
2848 \fi%
2849 }%
2850 %

```

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

```

2851 \newcommand*{\clearl@leftpage}{%
2852 \ifdim\pagetotal=0pt\hbox{}\fi%
2853 \newpage%
2854 \insert@notes@for@onlyside%
2855 \ifodd\c@page\else
2856 \led@err@LeftOnRightPage
2857 \hbox{}%
2858 \cleardoublepage
2859 \fi}
2860
2861 \newcommand*{\clearl@rightpage}{%
2862 \ifdim\pagetotal=0pt\hbox{}\fi%
2863 \newpage%
2864 \insert@notes@for@onlyside%
2865 \ifodd\c@page
2866 \led@err@RightOnLeftPage
2867 \hbox{}%
2868 \cleartoevenpage
2869 \fi}
2870
2871 %

```

XVII.6 Lines managing

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

```

2872 \newcommand*{\getlinesfromparlistL}{%
2873 \ifx\linesinpar@listL\empty

```

```

2874 \gdef\@cs@linesinparL{0}%
2875 \else
2876 \gl@p\linesinpar@listL\to\@cs@linesinparL
2877 \fi}
2878 \newcommand*\getlinesfromparlistR}{%
2879 \ifx\linesinpar@listR\empty
2880 \gdef\@cs@linesinparR{0}%
2881 \else
2882 \gl@p\linesinpar@listR\to\@cs@linesinparR
2883 \fi}
2884
2885 %

```

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

```

2886 \newcommand*\getlinesfrompagelistL}{%
2887 \ifx\linesonpage@listL\empty
2888 \gdef\@cs@linesonpageL{1000}%
2889 \else
2890 \gl@p\linesonpage@listL\to\@cs@linesonpageL
2891 \fi}
2892 \newcommand*\getlinesfrompagelistR}{%
2893 \ifx\linesonpage@listR\empty
2894 \gdef\@cs@linesonpageR{1000}%
2895 \else
2896 \gl@p\linesonpage@listR\to\@cs@linesonpageR
2897 \fi}
2898
2899 %

```

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

```

2900 \newcommand*\@writelinesonpageL}[1]{%
2901 \edef\next{\write\linenum@out{\string\@lopL{#1}}}%
2902 \next}
2903 \newcommand*\@writelinesonpageR}[1]{%
2904 \edef\next{\write\linenum@outR{\string\@lopR{#1}}}%
2905 \next}
2906
2907 %

```

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

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

```

2908 \newcommand*{\l@dcalc@maxoftwo}[3]{%
2909   \ifnum #2>#1\relax
2910     #3=#2\relax
2911   \else
2912     #3=#1\relax
2913   \fi}
2914 \newcommand*{\l@dcalc@minoftwo}[3]{%
2915   \ifnum #2<#1\relax
2916     #3=#2\relax
2917   \else
2918     #3=#1\relax
2919   \fi}
2920
2921 %

```

XVII.7 Page break managing

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

```

\checkpageL
\checkpageR
2922 \newif\ifl@dsamepage
2923 \l@dsamepagetrue
2924 \newif\ifl@dpagetrue
2925
2926 \newcommand*{\checkpageL}{%
2927   \l@dpagetrue
2928   \l@dsamepagetrue
2929   \check@goal
2930   \ifdim\pagetotal<\ledthegoal
2931     \ifnum\umpagelinesL<\l@minpagelines
2932       \else
2933         \ifnomaxlines%
2934         \else%
2935           \l@dsamepagefalse%
2936           \l@dpagetruefalse%
2937         \fi%
2938       \fi
2939     \else
2940       \l@dsamepagefalse
2941       \l@dpagetrue
2942     \fi%
2943   \ifprint@last@after@pendL%
2944     \l@dpagetruefalse%
2945     \l@dsamepagefalse%
2946     \print@last@after@pendLfalse%

```

```

2947 \fi%
2948 }%
2949
2950 \newcommand*\checkpageR{%
2951 \l@dpagetrue
2952 \l@dsamepagetrue
2953 \check@goal
2954 \ifdim\pagetotal<\ledthegoal
2955 \ifnum\numpagelinesR<\l@dmminpagelines
2956 \else
2957 \ifnomaxlines%
2958 \else%
2959 \l@dsamepagefalse%
2960 \l@dpagetruefalse%
2961 \fi%
2962 \fi
2963 \else
2964 \l@dsamepagefalse
2965 \l@dpagetrue
2966 \fi%
2967 \ifprint@last@after@pendR%
2968 \l@dpagetruefalse%
2969 \l@dsamepagefalse%
2970 \print@last@after@pendRfalse%
2971 \fi%
2972 }%
2973
2974 %

```

`\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`.

```

2975 \newcommand*\checkpbL{
2976 \IfStrEq{\ledpb@setting}{after}{
2977 \xifinlistcs{\the\absline@num}{l@prev@pb}{\l@dpagetrue\
l@dsamepagefalse}{
2978 \xifinlistcs{\the\absline@num}{l@prev@nopb}{\l@dpagetruefalse\
l@dsamepagetrue}{
2979 }{
2980 \IfStrEq{\ledpb@setting}{before}{
2981 \numdef{\next@absline}{\the\absline@num+1}
2982 \xifinlistcs{\next@absline}{l@prev@pb}{\l@dpagetrue\
l@dsamepagefalse}{
2983 \xifinlistcs{\next@absline}{l@prev@nopb}{\l@dpagetruefalse\
l@dsamepagetrue}{
2984 }{
2985 }
2986
2987 \newcommand*\checkpbR{
2988 \IfStrEq{\ledpb@setting}{after}{

```



```

2989 \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\l@dpagfulltrue\
l@dsamepagefalse}{\}
2990 \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\l@dpagfullfalse\
l@dsamepagetrue}{\}
2991 }{\}
2992 \IfStrEq{\led@pb@setting}{before}{
2993 \numdef{\next@abslineR}{\the\absline@numR+1}
2994 \xifinlistcs{\next@abslineR}{l@prev@pbR}{\l@dpagfulltrue\
l@dsamepagefalse}{\}
2995 \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\l@dpagfullfalse\
l@dsamepagetrue}{\}
2996 }{\}
2997 }
2998 %

```

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

\checkverseR

```

2999 \newcommand{\checkverseL}{
3000 \ifinstanzaL
3001 \iflednopbinverse
3002 \ifinserthangingsymbol
3003 \numgdef{\prev@abslineverse}{\the\absline@num-1}
3004 \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{\}
3005 \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesL<3\ledpbnum{\
prev@abslineverse}\fi}{\}
3006 \fi
3007 \fi
3008 \fi
3009 }
3010 \newcommand{\checkverseR}{
3011 \ifinstanzaR
3012 \iflednopbinverse
3013 \ifinserthangingsymbolR
3014 \numgdef{\prev@abslineverse}{\the\absline@numR-1}
3015 \IfStrEq{\led@pb@setting}{after}{\lednopbnumR{\prev@abslineverse}}{\}
3016 \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesR<3\ledpbnumR{\
prev@abslineverse}\fi}{\}
3017 \fi
3018 \fi
3019 \fi
3020 }
3021 %

```

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

\check@goal \newdimen\ledthegoal
 \ifshiftedpstarts

```

3024 \newcommand*{\@goalfraction}{0.95}
3025 \else
3026 \newcommand*{\@goalfraction}{0.9}
3027 \fi
3028
3029 \newcommand*{\check@goal}{%
3030 \ledthegoal=\@goalfraction\pagegoal}
3031 \newcommand{\setgoalfraction}[1]{%
3032 \xdef\@goalfraction{#1}%
3033 }
3034 %

```

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

```

\ifwrittenlinesL
3035 \newif\ifwrittenlinesL
3036 \newif\ifwrittenlinesR
3037
3038 %

```

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.

```

3039 \newif\if@getnextbox%
3040 %

```

\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.

```

3041 \newcommand*{\get@nextboxL}{%
3042 \ifvbox\namebox{1@dLcolrawbox\the\1@dpscL}% box is not empty
3043 %

```

The current box is not empty; do nothing.

```

3044 \else% box is empty
3045 %

```

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

```

3046 \@getnextboxtrue%Should be local, but be cautious
3047 %

```

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

```

3048 \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscl}>\@donetotallinesL
3049 \parledgroup@notes@endL%
3050 \unless\ifnosyncpstarts%
3051 \getnextboxfalse%
3052 %

```

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.

```

3053 \ifnomaxlines%
3054 \ifdim\pagetotal<\ledthegoal%
3055 \numdef{\@tmp}{\l@dpscl+1}%
3056 \ifcsdef{afterlines@pstart@\@tmp R}{%
3057 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp R}}
3058 %
3059 {}%
3060 {\ifcsdef{minpage@pstart@\@tmp}%
3061 {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
3062 {\ifnum\numpagelinesL=1@dminpagelines%
3063 \getnextboxtrue%
3064 \fi%
3065 }%
3066 {\@getnextboxtrue}}%
3067 {\@getnextboxtrue}}%
3068 }%
3069 {}%
3070 \fi%
3071 \fi%
3072 \fi%
3073 \else%
3074 \ifnomaxlines%
3075 \numdef{\@tmp}{\the\l@dpscl+1}%
3076 \ifcsdef{minpage@pstart@\@tmp}{%
3077 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
3078 {\ifdimgreater{\pagetotal}{\ledthegoal}%
3079 {\@getnextboxtrue}%
3080 {\@getnextboxfalse}%
3081 }%
3082 {\@getnextboxtrue}%
3083 }{}%
3084 \fi%
3085 \fi%
3086 %

```

Sufficient lines have been output.

```

3087 \if@getnextbox%
3088 \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscl}=\@donetotallinesL
3089 \parledgroup@notes@endL

```

```

3090 \fi
3091 \ifwrittenlinesL\else
3092 %

```

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

```

3093 \@writelinesinparL
3094 \writtenlinesLtrue
3095 \fi
3096 \ifnum\l@dnumstartsL>\l@dpscl
3097 %

```

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.

```

3098 \writtenlinesLfalse
3099 \ifbypstart@
3100 \global\line@num=0%
3101 \resetprevline%
3102 \fi
3103 % Add the content of the optional argument of the previous \protect\cs{pend
3104 % \begin{macrocode}
3105 \csuse{after@pendL@the\l@dpscl}%
3106 \global\csundef{after@pendL@the\l@dpscl}%
3107 %

```

Check the number of lines

```

3108 \l@dcalc@maxoftwo{\the\usernamecount{l@dmaxlinesinpar\the\l@dpscl}}%
3109 {\the\@donetotallinesL}%
3110 {\usernamecount{l@dmaxlinesinpar\the\l@dpscl}}%
3111 \global\@donetotallinesL \z@
3112 %

```

Go to the next pstart

```

3113 \global\advance\l@dpscl \@ne
3114 \global\pstartnumtrue%
3115 \restore@pstartL@pc%
3116 %

```

Add notes of parallel ledgroup.

```

3117 \parledgroup@notes@endL
3118 \parledgroup@correction@notes@spacing@final{L}
3119 \else
3120 %

```

```

3121 \print@last@after@pendLtrue%
3122 \fi
3123 \fi
3124 \fi}
3125 %

```

```

3126 \newcommand*{\get@nextboxR}{%
3127   \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}% box is not empty
3128   \else% box is empty
3129     \@getnextboxtrue%
3130     \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscR}>\@donetotallinesR
3131       \parledgroup@notes@endR
3132       \unless\ifnosyncpstarts%
3133         \@getnextboxfalse%
3134         \ifnomaxlines%
3135           \ifdim\pagetotal<\ledthegoal%
3136             \numdef{\@tmp}{\l@dpscR+1}%
3137             \ifcsdef{afterlines@pstart@\@tmp L}{%
3138               \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp L}}
3139               {}%
3140               {\ifcsdef{minpage@pstart@\@tmp}%
3141                 {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3142                   {\ifnum\numpagelinesR=1@dminpagelines%
3143                     \@getnextboxtrue%
3144                     \fi%
3145                     }%
3146                     {\@getnextboxtrue}}%
3147                     {\@getnextboxtrue}}%
3148                     }%}%
3149                     }%
3150                     {}%
3151                     \fi%
3152                     \fi%
3153                     \fi%
3154                   \else%
3155                     \ifnomaxlines%
3156                     \numdef{\@tmp}{\the\l@dpscR+1}%
3157                     \ifcsdef{minpage@pstart@\@tmp}{%
3158                     \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3159                     {\ifdimgreater{\pagetotal}{\ledthegoal}%
3160                     {\@getnextboxtrue}%
3161                     {\@getnextboxfalse}%
3162                     }%
3163                     {\@getnextboxtrue}%
3164                     }{}
3165                     \fi%
3166                   \fi%
3167                 \if@getnextbox%
3168                   \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscR}=\@donetotallinesR
3169                     \parledgroup@notes@endR
3170                     \fi
3171                     \ifwrittenlinesR\else
3172                     \@writelinesinparR
3173                     \writtenlinesRtrue
3174                     \fi

```

```

3175 \ifnum\l@dnumpstartsR>\l@dpscR
3176 \writtenlinesRfalse
3177 \ifbypstartR
3178 \global\line@numR=0%
3179 \resetprevline@%
3180 \fi
3181 \csuse{after@pendR@the\l@dpscR}%
3182 \global\csundef{after@pendR@the\l@dpscR}%
3183 \l@dcalc@maxoftwo{\the\usernamecount{l@maxlinesinpar\the\l@dpscR}}%
3184 \the\@donetotallinesR}%
3185 \usernamecount{l@maxlinesinpar\the\l@dpscR}}%
3186 \global\@donetotallinesR \z@
3187 \global\advance\l@dpscR \@ne
3188 \global\pstartnumRtrue%
3189 \restore@pstartR@pc%
3190 \parledgroup@notes@endR
3191 \parledgroup@correction@notes@spacing@final{R}
3192 \else
3193 \print@last@after@pendRtrue%
3194 \fi
3195 \fi
3196 \fi}
3197
3198 %

```

XVIII Page numbering

XVIII.1 Global options

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

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

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

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

counter restart in `\par@patch@pagenumbering`. We also call `\par@patch@thepage` it at the beginning of the document.

```

3199
3200 \newcommand{\par@patch@thepage}{%
3201   \ifboolexpr{%
3202     bool{sameparallelpagenumber}%
3203     or bool{prevpgnotnumbered}%
3204   }%
3205   {%
3206     \patchcmd{\thepage}%
3207       {page}{par@page}%
3208       {}%
3209       {\led@error@fail@patch@thepage}%
3210   }{}%
3211 }%
3212
3213 \newcommand{\par@patch@pagenumbering}{%
3214   \ifboolexpr{%
3215     bool{sameparallelpagenumber}%
3216     or bool{prevpgnotnumbered}%
3217   }%
3218   {%
3219     \setcounter{par@page}{1}%
3220   }%
3221   {}%
3222   \par@patch@thepage%
3223 }%
3224
3225 \ifl@dmemoir%
3226   \apptocmd{\@mempnum}%
3227     {\par@patch@pagenumbering}%
3228     {}%
3229     {\led@error@fail@patch@%mempnum}%
3230
3231 \else%
3232   \apptocmd{\pagenumbering}%
3233     {\par@patch@pagenumbering}%
3234     {}%
3235     {\led@error@fail@patch@pagenumbering}%
3236 \fi%
3237
3238 \AtBeginDocument{\par@patch@thepage}%
3239 %

```

\@outputpage As its name says, `\@outputpage` is a \LaTeX 's macro called in the output routine. It is this macro which increases the page counter.. We patch it in order to increase, conditionally, the `par@page` counter.

```

3240 \AtBeginDocument{%

```

```

3241 \apptocmd{\@outputpage}{%
3242   \ifsameparallelpagenumber%
3243     \ifl@dprintingpages%
3244       \ifodd\c@page\else%
3245         \stepcounter{par@page}%
3246       \fi%
3247     \else%
3248       \stepcounter{par@page}%
3249     \fi%
3250   \else%
3251     \stepcounter{par@page}%
3252   \fi%
3253 }%
3254 {}%
3255 {\led@error@fail@patch@@@outputpage}%
3256 }
3257 %

```

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

```

3258 \newcounter{par@page}%
3259 \setcounter{par@page}{1}%
3260 %

```

XVIII.2 mainmatter option of \Pages

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

```

\ifPages@mainmatter 3261 \newif\ifPages@mainmatter
\Pages@mainmatter 3262 \newcommand{\Pages@mainmatter}{%
3263   \ifPages@mainmatter%
3264     \pagenumbering{arabic}%
3265     \addtocounter{page}{1}%
3266     \addtocounter{par@page}{-1}%
3267     \patchcmd{\thepage}{page}{par@page}{}{}%
3268   \fi%
3269 }
3270 %

```

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 3271 \pretocmd{\chapter}{%
3272 \ifl@dprintingpages%
3273 \vbox{}}%
3274 \fi%
3275 }%
3276 {}%
3277 {}%
3278 %

```

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

```

3279 \newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}}
3280 \eledsectnotoc{R}
3281 %

```

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

```

3282 \newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}}
3283 \eledsectmark{L}
3284 %

```

\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.

```

3285 \newskip\eledsection@correcting@skip
3286 %

```

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

```

3287 \newwrite\eled@sectioningR@out
3288 %

```

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).

```

3289 \def\l@prev@pbR{}
3290 \def\l@prev@nopbR{}
3291 %

```

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

```
3292 \newcommand{\ledpbR}{\write\linenum@outR{\string\led@pbR}}
3293 \newcommand{\ledpbnumR}[1]{\write\linenum@outR{\string\led@pbnumR{#1}}}
3294 \newcommand{\lednopbR}{\write\linenum@outR{\string\led@nopbR}}
3295 \newcommand{\lednopbnumR}[1]{\write\linenum@outR{\string\led@nopbnumR{#1}}}
3296 %
```

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

```
3297 \newcommand{\led@pbR}{\listxadd{\l@prev@pbR}{\the\absline@numR}}
3298 \newcommand{\led@pbnumR}[1]{\listxadd{\l@prev@pbR}{#1}}
3299 \newcommand{\led@nopbR}{\listxadd{\l@prev@nopbR}{\the\absline@numR}}
3300 \newcommand{\led@nopbnumR}[1]{\listxadd{\l@prev@nopbR}{#1}}
3301 %
```

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).

```
3302 \newmarks\parledgroup@
3303 \newmarks\parledgroup@series
3304 \newmarks\parledgroup@type
3305 %
```

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

```
3306 \newcommand{\parledgroup@notes@startL}{%
3307   \ifnum\usenamecount{1@dmxlinesinpar\the\l@dpscL}>0%
3308     \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
3309       bhooknoteX@\splitfirstmarks\parledgroup@series}}{}%
3309     \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
3310       bhookXnote@\splitfirstmarks\parledgroup@series}}{}%
3310     \fi%
3311     \global\ledgroupnotesL@true%
3312     \insert@noterule@ledgroup{L}%
3313   }
3314 \newcommand{\parledgroup@notes@startR}{%
3315   \ifnum\usenamecount{1@dmxlinesinpar\the\l@dpscR}>0%
```

```

3316 \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
bhooknoteX@\splitfirstmarks\parledgroup@series}}{ }%
3317 \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
bhookXnote@\splitfirstmarks\parledgroup@series}}{ }%
3318 \fi%
3319 \global\ledgroupnotesR@true%
3320 \insert@noterule@ledgroup{R}%
3321 }
3322 %

```

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

```

3323 \newcommand{\parledgroup@notes@endL}{%
3324 \global\ledgroupnotesL@false%
3325 }
3326 \newcommand{\parledgroup@notes@endR}{%
3327 \global\ledgroupnotesR@false%
3328 }
3329 %

```

`\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`

```

3330 \newcommand{\insert@noterule@ledgroup}[1]{
3331 \IfStrEq{\splitbotmarks\parledgroup@}{begin}{%
3332 \IfStrEq{\splitbotmarks\parledgroup@type}{Xfootnote}{
3333 \csuse{ifledgroupnotes#1@}
3334 \vskip\skip\csuse{mp\splitbotmarks\parledgroup@series footins}
3335 \csuse{\splitbotmarks\parledgroup@series footnoterule}
3336 \fi
3337 }
3338 {}
3339 \IfStrEq{\splitbotmarks\parledgroup@type}{footnoteX}{
3340 \csuse{ifledgroupnotes#1@}
3341 \vskip\skip\csuse{mpfootins\splitbotmarks\parledgroup@series}
3342 \csuse{footnoterule\splitbotmarks\parledgroup@series}
3343 \fi
3344 }{}
3345 }
3346 {}
3347 }
3348 %

```

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

```

3349 \newcommand{\setparledgroupnotespadding}[1]{\gdef\@parledgroupnotespadding
{#1}}
3350 \newcommand{\@parledgroupnotespadding}{}
3351 %

```

`\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`.

```

3352 \dimdef{\parledgroup@notespacing@correction}{0pt}
3353 \newcommand{\parledgroup@notespacing@set@correction}{%
3354   {\@getfirstseries\csuse{Xnotefontsize@}\@firstseries}%We suppose all the
series has the same footnote size setup
3355   \@parledgroupnotespacing\dimgdef{\temp@spacing}{\baselineskip}}%
3356   \dimgdef{\parledgroup@notespacing@correction}{\baselineskip-\temp@spacing}
}%
3357 }
3358 %

```

`\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.

```

3359 \newcommand{\parledgroup@correction@notespacing@init}{
3360   \dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}
3361   \dimdef{\parledgroup@notespacing@correction@modulo}{0pt}
3362 }
3363 \parledgroup@correction@notespacing@init
3364 %

```

`\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.

```

3365 \newcommand{\parledgroup@correction@notespacing@final}[1]{
3366   \ifparledgroup
3367     \vspace{\parledgroup@notespacing@correction@accumulated}
3368     \parledgroup@correction@notespacing@init%
3369     \ifstrequal{#1}{L}{
3370       \numdef{\@checking}{\the\l@dpscl-1}
3371     }{
3372       \numdef{\@checking}{\the\l@dpscr-1}
3373     }
3374     \dimdef{\@beforenotes@current@diff}{\csuse{\parledgroup@beforenotes@
@checking L}-\csuse{\parledgroup@beforenotes@\@checking R}}%
3375     \ifstrequal{#1}{L}%
3376     {% Left
3377       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{-\
@beforenotes@current@diff}}%
3378     }%
3379     {% Right
3380       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{\
@beforenotes@current@diff}}{}
3381     }%

```

```

3382 \fi
3383 }
3384 %

```

`\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.

```

3385 \newcommand{\parledgroup@correction@notespacing}[1]{%
3386   \csuse{ifledgroupnotes#1@}%
3387   \vspace{-\parledgroup@notespacing@correction}%
3388   \dimdef{\parledgroup@notespacing@correction@accumulated}{\parledgroup@notespacing@correction@accumulated+\parledgroup@notespacing@correction}%
3389   \dimdef{\parledgroup@notespacing@correction@modulo}{\parledgroup@notespacing@correction@modulo+\parledgroup@notespacing@correction}%
3390   \ifdimless{\parledgroup@notespacing@correction@modulo}{\baselineskip}{\advance\numpagelinesL -\@ne%
3391     \dimdef{\parledgroup@notespacing@correction@modulo}{\parledgroup@notespacing@correction@modulo-\baselineskip}%
3392   }% mean greater than equal
3393   \fi%
3394 }
3395 %

```

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

```

3396 \dimdef\parledgroup@beforenotesL{0pt}
3397 \dimdef\parledgroup@beforenotesR{0pt}
3398 %

```

The macro `\parledgroup@beforenotes@save` dumps the space before notes of the current parallel ledgroup in a macro named with the current `pstart` number.

```

3399 \newcommand{\parledgroup@beforenotes@save}[1]{
3400   \ifparledgroup
3401     \csdimgdef{@parledgroup@beforenotes@\the\csuse{1@dnumpstarts#1}#1}{\csuse{\parledgroup@beforenotes#1}}

```

```
3402 \csdimgdef{parledgroup@beforenotes#1}{0pt}  
3403 \fi  
3404 }  
3405 %
```

XXII Compatibility with eledmac

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

```
3406 \ifeledmaccompat%  
3407  
3408  
3409 \unless\ifnocritical@  
3410 \let\onlyXside\Xonlyside  
3411 \fi  
3412 \fi  
3413 %
```

XXIII The End

</code>

Appendix A Some things to do when changing version

Appendix A.1 Migration to eledpar 1.4.3

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

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

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

See the following two examples:

With standard `\hangingsymbol`:

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

With the modification of the `hangingsymbol`:

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

Appendix A.2 Migration from eledpar to reledpar

As for migration from eledmac to reledmac:

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

Appendix A.2.1 Deprecated options

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

Appendix A.2.2 `\renewcommand` replaced with command

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

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

Appendix A.2.3 Commands the names of which have changed

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

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

Appendix A.3 Migration to `reledpar` 2.2.0

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

Appendix A.4 Migration to `reledpar` 2.3.0

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

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

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

Appendix A.5 Migration to `reledpar` 2.4.0

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

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

Appendix A.6 Migration to `reledpar` 2.5.0

If you use either `\stanza` or `astanza` environment, please read Appendix A.12 p. 356.

Appendix A.7 Migration to `reledpar` 2.6.0

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

Appendix A.8 Migration to `reledpar` 2.6.1

If you use `perpage` package to control footnote numbering, please read the handbook on 6.3.3 p. 14.

Appendix A.9 Migration to reledpar 2.14.0

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

Appendix 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.

References

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

Index

Symbols

<code>\@adv</code>	1
<code>\@astanza@line</code>	1
<code>\@cs@linesinparL</code>	1
<code>\@cs@linesinparR</code>	1
<code>\@cs@linesonpageL</code>	1
<code>\@cs@linesonpageR</code>	1
<code>\@donereallinesL</code>	1
<code>\@donereallinesR</code>	1
<code>\@donetotallinesL</code>	1
<code>\@donetotallinesR</code>	1
<code>\@eledsectionL</code>	1
<code>\@eledsectionR</code>	1
<code>\@lab</code>	1
<code>\@lopL</code>	1
<code>\@lopR</code>	1
<code>\@nl</code>	1
<code>\@nl@regR</code>	1
<code>\@outputpage</code>	1

\@par@sync@option	1
\@par@this@sync@option	1
\@parledgroupnotespacing	1
\@pend	1
\@pendR	1
\@pstart	1
\@pstartR	1
\@pstartsfalse	1
\@pstartstrue	1
\@ref	1
\@ref@reg@parseR	1
\@ref@regR	1
\@set	1
\@stopastanza	1
\@writelinesinparL	1
\@writelinesinparR	1
\@writelinesonpageL	1
\@writelinesonpageR	1
\@writepageofparL	1
\@writepageofparR	1
CLASSmemoir	118
COMMAND\+	52
COMMAND\@Rlineflag	81, 155
COMMAND\@adv	42, 154
COMMAND\@cs@linesinparL	109
COMMAND\@cs@linesonpageL	110
COMMAND\@eledsectionL	91
COMMAND\@eledsectionR	91
COMMAND\@eledsectmark	121
COMMAND\@eledsectnotoc	121
COMMAND\@goalfraction	12, 113
COMMAND\@l@dtempcnta	75
COMMAND\@lab	41, 82, 154
COMMAND\@lopL	47, 104, 110
COMMAND\@lopR	47, 110
COMMAND\@mempnum	118
COMMAND\@namedef	29
COMMAND\@namuse	29
COMMAND\@nl	41, 48, 82, 154
COMMAND\@nl@regR	41
COMMAND\@outputpage	118, 119
COMMAND\@page	82
COMMAND\@par@sync@option	41
COMMAND\@parledgroupnotespacing	123
COMMAND\@pend	46
COMMAND\@pendR	46
COMMAND\@pstart	46
COMMAND\@pstartstrue	90
COMMAND\@ref	44–46, 48, 154
COMMAND\@ref@later	45

COMMAND\@ref@reg@parseargR	45
COMMAND\@ref@regR	44
COMMAND\@set	43, 154
COMMAND\@sw	45
COMMAND\AtBeginPairs	8, 54, 152, 156
COMMAND\AtEveryPend	152–154
COMMAND\AtEveryPstart	2, 18, 61, 92, 152–154
COMMAND\AtEveryPstartCall	2, 18, 61, 153
COMMAND\AtEveryStanza	156
COMMAND\AtEveryStopStanza	156
COMMAND\Clear the right lines for \read@linelist	40
COMMAND\Columns	8, 18, 53, 91, 97, 149–151, 153, 155
COMMAND\Columns@print@after@pend	99
COMMAND\Columns@print@before@pstart	99
COMMAND\Colwidth	8, 10, 101
COMMAND\Leftsidehook	149
COMMAND\Leftsidehookend	149
COMMAND\MakePerPage	14
COMMAND\Pages	4, 8, 10, 12, 13, 18, 52, 53, 72, 76, 77, 79, 101, 107, 118, 120, 124, 149, 151, 153, 155
COMMAND\Pages@mainmatter	120
COMMAND\Pairs	52
COMMAND\Rcolwidth	8, 10, 101
COMMAND\Rightsidehook	149
COMMAND\Rightsidehookend	149
COMMAND\Rlineflag	127
COMMAND\Xendlineflag	15, 155
COMMAND\Xlineflag	15, 128, 155
COMMAND\Xmaxhnotes	13
COMMAND\Xnoteswidthliketwocolumns	9, 152
COMMAND\Xonlyside	14, 76, 128, 156
COMMAND\Xtxtbeforenotes	156
COMMAND\&	19
COMMAND\absline@numR	40
COMMAND\add@penalties	75
COMMAND\add@penaltiesL	75
COMMAND\advanceline	42, 48, 154
COMMAND\affixline@num	70
COMMAND\affixline@numR	70, 149, 150
COMMAND\affixpstart@num	72
COMMAND\affixpstart@numR	72
COMMAND\affixside@note	83
COMMAND\aftercolumnseparator	9, 97, 151, 157
COMMAND\araw@textfalse	90
COMMAND\araw@texttrue	90
COMMAND\at@begin@pairs	54
COMMAND\autopar	18
COMMAND\ballast@count	75
COMMAND\baselineskip	96, 125
COMMAND\bbl@set@language	87, 154
COMMAND\beforecolumnseparator	9, 97, 151

COMMAND\begin	19
COMMAND\beginnumbering	15, 16, 18, 30, 41, 97, 149–151, 153
COMMAND\beginnumberingR	35, 48
COMMAND\bf	150
COMMAND\bfseries	150
COMMAND\brokenpenalty	75
COMMAND\chapter	120, 149
COMMAND\check@goal	113
COMMAND\check@pstarts	90
COMMAND\checkpageL	104, 111
COMMAND\checkpb@columns	96
COMMAND\checkpbL	112
COMMAND\checkpbR	112
COMMAND\checkraw@text	90
COMMAND\checkverseL	113
COMMAND\checkverseR	113
COMMAND\clear(double)page	108
COMMAND\clearl@dleftpage	109
COMMAND\clearl@drihtpage	109
COMMAND\clearpage	109, 153
COMMAND\cleartoevenpage	108
COMMAND\cleartol@devenpage	108, 120
COMMAND\columnrulewidth	9, 96
COMMAND\columns	157
COMMAND\columns@position	97
COMMAND\columnseparator	9
COMMAND\columnsposition	9, 151
COMMAND\correct@Xfootins@box	153
COMMAND\correct@footinsX@box	153
COMMAND\critext	152
COMMAND\curname	53
COMMAND\displaywidowpenalty	75
COMMAND\do@actions	68
COMMAND\do@actions@fixedcode	149
COMMAND\do@actions@nextR	68
COMMAND\do@actionsR	68, 149
COMMAND\do@ballast	75
COMMAND\do@ballastR	68
COMMAND\do@insidelineLhook	151
COMMAND\do@insidelineRhook	151
COMMAND\do@line	62
COMMAND\do@line(L/R)	65
COMMAND\do@lineL	62, 75, 149, 150
COMMAND\do@lineLhook	149
COMMAND\do@lineR	65, 149, 150, 152
COMMAND\do@lineRhook	149
COMMAND\do@lockoff	154
COMMAND\do@lockoffR	43
COMMAND\do@lockon	154
COMMAND\do@lockonR	43

COMMAND\doinsidelineLhook	19, 129, 152, 157
COMMAND\doinsidelineRhook	19, 129, 152, 157
COMMAND\doinsidelinehook	19
COMMAND\dolineLhook	19, 152
COMMAND\dolineRhook	19, 152
COMMAND\dolinehook	19
COMMAND\edindex	152
COMMAND\edlabel	150, 152, 153
COMMAND\edlineref	157
COMMAND\edtext	15, 44, 48, 49, 80, 152, 156
COMMAND\edtext@later	79, 81, 107
COMMAND\edtext@now	107
COMMAND\edtextlater	15, 79, 80, 156, 157
COMMAND\edtextnow	15, 79–81, 156, 157
COMMAND\eled@sectioningR@out	121
COMMAND\eledchapter	153
COMMAND\eledsection	152–154
COMMAND\eledsection@correcting@skip	121
COMMAND\eledsectmark	21, 121
COMMAND\eledsectnotoc	21, 121
COMMAND\eledxxx	152
COMMAND\end	19
COMMAND\endgraf	60
COMMAND\endlock	49, 154
COMMAND\endnumbering	9, 16, 18, 31, 149, 153
COMMAND\endsub	48, 154
COMMAND\endumbering	16
COMMAND\expandafter	51
COMMAND\extensionchars	30
COMMAND\firstlinenum	17, 151, 154, 155
COMMAND\firstsublinenum	151, 154, 155
COMMAND\fix@page	42, 154
COMMAND\flag@end	48, 149, 152
COMMAND\flag@start	48, 152
COMMAND\flush@notesR	76
COMMAND\footnote	53
COMMAND\footnoteX	50, 51
COMMAND\footnoteX@reading	155
COMMAND\footnoteXmk	14
COMMAND\footnoteXnomk	14, 50
COMMAND\frontmatter	13, 22
COMMAND\get@nextboxL	150
COMMAND\get@nextboxR	150
COMMAND\getline@numL	67
COMMAND\getline@numR	67
COMMAND\getlinesfrompagelistL	110
COMMAND\getlinesfrompagelistR	110
COMMAND\getlinesfromparlistL	109
COMMAND\getlinesfromparlistR	109
COMMAND\gl@p	51

COMMAND\goalfraction	127
COMMAND\hangingsymbol	127, 150
COMMAND\hfill	94, 97
COMMAND\hidenumbering	17, 154
COMMAND\if@getnextbox	114
COMMAND\ifPages@mainmatter	120
COMMAND\ifbypage@	154
COMMAND\ifbypstart@R	154
COMMAND\ifcsboxvoid	29
COMMAND\ifdim	94
COMMAND\ifinserthangingsymbol	84
COMMAND\ifinserthangingsymbolR	84
COMMAND\ifl@dpagefull	111
COMMAND\ifl@dpageing	26, 152
COMMAND\ifl@dpairing	26, 149
COMMAND\ifl@dsamelang	152
COMMAND\ifl@dsamepage	111
COMMAND\ifl@pagefull	104
COMMAND\ifledRcol	26
COMMAND\iflinenumberLevenifblank	39
COMMAND\iflinenumberRevenifblank	39
COMMAND\iflledRcol	150
COMMAND\ifmovecolumnspostiononrightpage	25
COMMAND\ifnumberedpar@	57
COMMAND\ifnumberingR	150
COMMAND\ifnumberpstart	53
COMMAND\ifpst@rtedL	30, 31, 58, 149
COMMAND\ifpst@rtedR	30
COMMAND\ifsublines@	43
COMMAND\insert@countR	44
COMMAND\insert@noterule@ledgroup	123
COMMAND\insert@notes@for@onlyside	79
COMMAND\insertlines@list	44
COMMAND\insertlines@listR	44
COMMAND\inserts@list	57
COMMAND\inserts@listR	74
COMMAND\l@d@nums	49
COMMAND\l@d@set	43, 49, 154
COMMAND\l@dLcolrawbox	57
COMMAND\l@dLcolrawbox1	89
COMMAND\l@dLcolrawbox2	89
COMMAND\l@dRcolrawbox	57
COMMAND\l@dcalc@maxoftwo	110
COMMAND\l@dcalc@minoftwo	110
COMMAND\l@dchecklang	149, 151
COMMAND\l@dcsnote	151
COMMAND\l@dleftbox	62, 104, 153
COMMAND\l@dlinenumR	39, 149
COMMAND\l@dlsnote	151
COMMAND\l@dmake@labels	81

COMMAND\l@dmaxlinesinpar	102
COMMAND\l@dmaxlinesinpar1	89
COMMAND\l@dminpagelines	102, 149
COMMAND\l@dnumpsstartsL	89, 149
COMMAND\l@dprintingcolumnstrue	152
COMMAND\l@dprintingpagestrue	152
COMMAND\l@dpscL	92, 102, 103, 116
COMMAND\l@dpscR	92, 102, 103
COMMAND\l@drsnote	151
COMMAND\l@dsetupmaxlinecounts	89
COMMAND\l@duselanguage	87, 88, 149
COMMAND\l@dzeromaxlinecounts	89
COMMAND\l@prev@nopbR	121
COMMAND\l@prev@pbR	121
COMMAND\labelpstarttrue	150
COMMAND\labelref@list	82
COMMAND\labelref@listR	81
COMMAND\lang	87
COMMAND\last@page@numR	42
COMMAND\led	150
COMMAND\led@nopbR	122
COMMAND\led@nopbnumR	122
COMMAND\led@pbR	122
COMMAND\led@pbnumR	122
COMMAND\ledinnerrote	20
COMMAND\ledleftnote	20
COMMAND\lednopb	20, 96, 112
COMMAND\lednopbR	122
COMMAND\lednopbnumR	122
COMMAND\ledouterote	20
COMMAND\ledpb	96, 112
COMMAND\ledpbR	122
COMMAND\ledpbnumR	122
COMMAND\ledrightnote	20
COMMAND\ledsidenote	20
COMMAND\ledstrutL	149
COMMAND\ledstrutR	149, 155
COMMAND\ledthegoal	113
COMMAND\ledtrutL	149, 155
COMMAND\leftlinenumR	39, 149
COMMAND\lemma	80
COMMAND\let	51
COMMAND\line@list@R	45
COMMAND\line@list@stuff	41, 48
COMMAND\line@margin	35
COMMAND\line@marginR	35, 149
COMMAND\line@numR	40
COMMAND\lineation	17, 153
COMMAND\lineation*	17, 35, 152
COMMAND\lineationR	17, 34, 153, 156

COMMAND\linenum	80
COMMAND\linenum@out	82
COMMAND\linenum@outR	47
COMMAND\linenumberLevenifblanktrue	18, 156
COMMAND\linenumberRevenifblank	156
COMMAND\linenumberRevenifblanktrue	18
COMMAND\linenumberlist	17, 129, 156
COMMAND\linenumberlistR	17, 129, 156
COMMAND\linenumberstyle	17, 128, 157
COMMAND\linenumberstyle*	128
COMMAND\linenumberstyleR	17, 128
COMMAND\linenumincrement	17, 151, 154, 155
COMMAND\linenummargin	17, 35, 149, 154–156
COMMAND\linenummargin*	17, 36, 155
COMMAND\linenummarginColumns	17, 36
COMMAND\linenummarginColumns*	17
COMMAND\linenummarginColumnsR	17
COMMAND\linenummarginR	17, 36, 155
COMMAND\linenumrepR	38, 149
COMMAND\linesinpar@listL	46, 109
COMMAND\linesonpage@listL	47, 110
COMMAND\lock@off	43
COMMAND\lock@on	43
COMMAND\mainmatter	2, 13, 22, 155
COMMAND\makeatletter	65
COMMAND\markboth	129, 157
COMMAND\maxchunks	7, 8, 19, 89, 90
COMMAND\maxdimen	77
COMMAND\maxhnotesX	13
COMMAND\memorydump	16, 33
COMMAND\movecolumnspostiononrightpage	95
COMMAND\msdata	149
COMMAND\n@num	153
COMMAND\new@lineL	48
COMMAND\new@lineR	48
COMMAND\newhookcommand@series	52
COMMAND\newif	153
COMMAND\newpage	108, 109, 153
COMMAND\newseries	52
COMMAND\newseries@	49
COMMAND\newseries@par	49, 51, 52
COMMAND\noeledxxx	152
COMMAND\nomark@	50
COMMAND\nomaxlines	46
COMMAND\notesXwidthliketwocolumns	9, 152
COMMAND\num@lines	75
COMMAND\num@lines(R)	57
COMMAND\numberingR	32
COMMAND\numberlinefalse	7
COMMAND\numberonlyfirstinline	150

COMMAND\numberstartfalse	16
COMMAND\numberstarttrue	16, 150, 155
COMMAND\one@line	57, 84
COMMAND\one@lineR	57
COMMAND\onlyXside	128
COMMAND\onlysideX	14, 76, 79, 153, 156
COMMAND\otherlanguage	154
COMMAND\page@action	43, 154
COMMAND\pagenumbering	118, 154
COMMAND\pages	13
COMMAND\pagetotal	104, 153
COMMAND\par@line	75
COMMAND\par@line(R)	57
COMMAND\par@patch@pagenumbering	119
COMMAND\par@patch@thepage	118, 119
COMMAND\par@sync@option	24
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	20
COMMAND\patchcmd	154
COMMAND\pausenumbering	9, 33, 156
COMMAND\pend	4, 7, 11, 18–20, 53, 57, 60, 61, 88, 99, 151, 152, 154, 155
COMMAND\pendL	151
COMMAND\pendR	151
COMMAND\pends	18
COMMAND\perpage	14
COMMAND\prev@nopbR	122
COMMAND\prev@pbR	122
COMMAND\prevpgstyle	25
COMMAND\print@Xnotes	76
COMMAND\print@Xnotes@forpages	76, 78, 153
COMMAND\print@columnseparator	94, 151
COMMAND\print@eledsectionL	64
COMMAND\print@leftcolumn	95

COMMAND\print@line	64
COMMAND\print@lineL	64
COMMAND\print@notesX@forpages	153
COMMAND\print@rightcolumn	95
COMMAND\printlinenumR	128
COMMAND\printlinesR	155
COMMAND\pstart 4, 7, 8, 11, 16, 18–20, 34, 49, 53, 57, 58, 60, 61, 88, 92, 99, 150, 151, 153–155, 157	
COMMAND\pstartL	61, 151
COMMAND\pstartR	61, 150, 151
COMMAND\pstartinfootnote	153
COMMAND\raw@text	88
COMMAND\read@linelist	40, 41, 131, 154
COMMAND\ref@reg	44
COMMAND\ref@regR	44, 154
COMMAND\relax	128
COMMAND\reledmac	154
COMMAND\renewcommand	127
COMMAND\resumenumbering	9, 33, 152, 156
COMMAND\resumenumberingR	35, 152
COMMAND\rightlinenumR	39, 149
COMMAND\sameword	80
COMMAND\section	149
COMMAND\section@num	30
COMMAND\selectlanguage	18, 87, 88
COMMAND\set@continuousnumberingforR	35
COMMAND\set@line	49, 154
COMMAND\set@line@action	43, 154
COMMAND\setRlineflag	17, 127
COMMAND\setgoalfraction	12, 127
COMMAND\sethangingsymbol	19
COMMAND\setline	43, 49, 154
COMMAND\setlinenum	43, 49, 154
COMMAND\setmsdataposition	157
COMMAND\setnoteposition...	97
COMMAND\setparledgroupnotespacing	127, 155
COMMAND\setposition...	97
COMMAND\setstanzaindents	9, 19, 127
COMMAND\setwidth...	97
COMMAND\sidenotemargin	20, 151
COMMAND\sidenotemargin*	20, 151
COMMAND\sidenotemarginR	20, 157
COMMAND\skipnumbering	16, 153
COMMAND\sloppy	8
COMMAND\stanza	8, 9, 17, 19, 55, 84, 128, 150
COMMAND\stanzanumtrue	20
COMMAND\startlock	49, 154
COMMAND\startsub	48, 154
COMMAND\sub@action	43, 154
COMMAND\sub@off	82
COMMAND\sub@on	82

COMMAND\subline@numR	40
COMMAND\sublinenumberstyle	17, 128
COMMAND\sublinenumberstyle*	128
COMMAND\sublinenumberstyleR	17, 128
COMMAND\sublinenumincrement	151, 154, 155
COMMAND\sublinenumrepR	38, 149
COMMAND\syntaxonly	155
COMMAND\sza@0@	19
COMMAND\textheight	13
COMMAND\textwidth	55
COMMAND\the@labelX	155
COMMAND\theledlanguageL	87, 88
COMMAND\theledlanguageR	87, 88
COMMAND\thepage	21, 118
COMMAND\thepstartL	16, 150
COMMAND\thepstartR	16, 150
COMMAND\thestanzaL	20
COMMAND\thestanzaR	20
COMMAND\ vbox	58
COMMAND\ vskip	123
COMMAND\ vsplit	75
COMMAND\widthliketwocolumns	9
COMMAND\widthliketwocolumnsfalse	9
COMMAND\widthliketwocolumnstrue	9
COMMAND\xflagref	155
COMMAND\xright@appenditem	51
COMMAND\xspace	23
COMMAND\xxxfootstart	97
COMMAND\xxxmatter	118
ENVIRONMENTLeftside	10, 55, 95
ENVIRONMENTRightside	10, 56, 95
ENVIRONMENTastanza	19, 84, 85, 128, 155
ENVIRONMENTcolumns	32, 155
ENVIRONMENTledgroup	6, 156
ENVIRONMENTleft	17
ENVIRONMENTpages	32, 54, 155
ENVIRONMENTpairs	54, 155
PACKAGEEDMAC	129
PACKAGEEDSTANZA	129
PACKAGEEledmac	153
PACKAGEEledpar	153
PACKAGETABMAC	129
PACKAGEbabel	18, 86–88, 155
PACKAGEedmac	129
PACKAGEeledmac	5, 89, 126, 127, 129, 151, 154
PACKAGEeledpar	5, 6, 13, 37, 127, 128, 151, 152
PACKAGEetoolbox	29, 94, 121
PACKAGEledmac	6
PACKAGEledpar	1, 6
PACKAGEMemoir	129

PACKAGEMusixtex	152
PACKAGEperpage	2, 14, 128, 155
PACKAGEpolyglossia	18, 86–88, 156
PACKAGEreledmac	1, 3, 5–9, 13, 17–21, 23, 25, 26, 30, 31, 35, 37, 39–44, 46–50, 52, 53, 64, 72, 81, 82, 84, 102, 108, 127, 128, 154–156
PACKAGEreledpar	1, 3, 5–12, 19–25, 34, 39, 40, 46, 47, 49, 52, 53, 81, 127–129, 154
PACKAGEsetspace	2, 21
PACKAGESyntonly	155
PACKAGExkeyval	23
PACKAGEXR	156

A

\absline@numR	1
\actionlines@listR	1
\actions@listR	1
\add@inserts@nextR	1
\add@insertsR	1
\add@penaltiesL	1
\add@penaltiesR	1
\advanceline	1
\affixline@numR	1
\affixpstart@numL	1
\affixpstart@numR	1
\affixside@noteR	1
\aftercolumnseparator	1, 9
\araw@textfalse	1
\araw@texttrue	1
astanza (environment)	19
\AtBeginPairs	1, 8
\AtEveryPstartCall	1
\autopar	18

B

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

C

\c@firstlinenumR	1
\c@firstsublinenumR	1
\c@linenumincrementR	1
\c@sublinenumincrementR	1
\ch@ck@l@ckR	1
\ch@cksub@l@ckR	1
\chapter	1
\chapterinpages	1
\check@goal	1
\check@pstarts	1
\checkpageL	1

<code>\checkpageR</code>	1
<code>\checkpb@columns</code>	1
<code>\checkpbL</code>	1
<code>\checkpbR</code>	1
<code>\checkraw@text</code>	1
<code>\checkverseL</code>	1
<code>\checkverseR</code>	1
<code>\clearl@dleftpage</code>	1
<code>\clearl@drightpage</code>	1
<code>\cleartoevenpage</code>	1
<code>\cleartol@devenpage</code>	1
<code>\columnrulewidth</code>	1, 9
<code>\Columns</code>	1, 8
<code>\columns@position</code>	1
<code>\Columns@print@after@pend</code>	1
<code>\Columns@print@before@pstart</code>	1
<code>\columnseparator</code>	1, 9
<code>\columnspostion</code>	1, 9
<code>\correct@footinsX@box</code>	1
<code>\correct@Xfootins@box</code>	1
<code>\countLline</code>	1
<code>\countRline</code>	1
<code>\critext</code>	1

D

<code>\do@actions@fixedcodeR</code>	1
<code>\do@actions@nextR</code>	1
<code>\do@actionsR</code>	1
<code>\do@ballastR</code>	1
<code>\do@insidelineLhook</code>	1
<code>\do@insidelineRhook</code>	1
<code>\do@lineL</code>	1
<code>\do@lineLhook</code>	1
<code>\do@lineR</code>	1
<code>\do@lineRhook</code>	1
<code>\do@lockoff</code>	1
<code>\do@lockoffR</code>	1
<code>\do@lockon</code>	1
<code>\do@lockonR</code>	1
<code>\doinsidelineLhook</code>	1, 19
<code>\doinsidelineRhook</code>	1, 19
<code>\dolineLhook</code>	1, 19
<code>\dolineRhook</code>	1, 19
<code>\dump@pstartL@pc</code>	1
<code>\dump@pstartR@pc</code>	1

E

<code>\edlabel</code>	1
<code>\edtext</code>	1
<code>\edtext@later</code>	1

<code>\edtext@now</code>	1
<code>\edtextlater</code>	1, 15
<code>\edtextnow</code>	1, 15
<code>\eled@sectioningR@out</code>	1
<code>\eledsection@correcting@skip</code>	1
<code>\eledsectmark</code>	1, 21
<code>\eledsectnotoc</code>	1, 21
<code>\endlock</code>	1
<code>\endnumbering</code>	1, 15
<code>\endnumberingR</code>	1
<code>\endsub</code>	1
environments:	
<code>astanza</code>	19
<code>Leftside</code>	15
<code>pages</code>	10
<code>pairs</code>	8
<code>Rightside</code>	15

F

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

G

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

H

\hidenumering	17
---------------------	----

I

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

L

\l@d@set	1
\l@dc@maxchunks	1
\l@dcalc@maxoftwo	1
\l@dcalc@minoftwo	1
\l@dcalcnum	1
\l@dchecklang	1

<code>\ldleftbox</code>	1
<code>\ldlinenumR</code>	1
<code>\ldmake@labelsR</code>	1
<code>\ldminpagelines</code>	1
<code>\ldnumpstartsL</code>	1
<code>\ldnumpstartsR</code>	1
<code>\ldpagefullfalse</code>	1
<code>\ldpagefulltrue</code>	1
<code>\ldrightrightbox</code>	1
<code>\ldsamesamepagefalse</code>	1
<code>\ldsamesamepagetrue</code>	1
<code>\ldsetupmaxlinecounts</code>	1
<code>\ldsetupprawboxes</code>	1
<code>\ldskipversenumberR</code>	1
<code>\ldusedbabelfalse</code>	1
<code>\ldusedbabeltrue</code>	1
<code>\lduselanguage</code>	1
<code>\ldzeromaxlinecounts</code>	1
<code>\l@pscL</code>	1
<code>\l@pscR</code>	1
<code>\labelref@listR</code>	1
<code>\last@page@numR</code>	1
<code>\Lcolwidth</code>	1, 8, 10
<code>\led@err@BadLeftRightPstarts</code>	1
<code>\led@err@Columns@InsideEnv</code>	1
<code>\led@err@Columns@WithoutEnv</code>	1
<code>\led@err@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@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>\ledpbnR</code>	1
<code>\ledpbR</code>	1
<code>\ledstrutL</code>	1
<code>\ledstrutR</code>	1
<code>\ledthegoal</code>	1
<code>\leftlinenumR</code>	1
<code>\leftpstartnumL</code>	1
<code>\leftpstartnumR</code>	1
<code>Leftside (environment)</code>	15
<code>\Leftsidehook</code>	1
<code>\Leftsidehookend</code>	1
<code>\line@list@stuffR</code>	1
<code>\line@listR</code>	1
<code>\line@marginR</code>	1
<code>\line@numR</code>	1
<code>\lineation*</code>	1, 17
<code>\lineationR</code>	1, 17
<code>\linenum@outR</code>	1
<code>\linenumberLevenifblanktrue</code>	17
<code>\linenumberRevenifblanktrue</code>	17
<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, 17
<code>\linenummarginColumns</code>	1, 17
<code>\linenummarginColumns*</code>	1, 17
<code>\linenummarginColumnsR</code>	1, 17
<code>\linenummarginR</code>	1, 17
<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>\notesXwidthliketwocolumns</code>	9
<code>\num@linesR</code>	1
<code>\numberpstartfalse</code>	16
<code>\numberpstarttrue</code>	16
<code>\numpagelinesL</code>	1
<code>\numpagelinesR</code>	1

O

<code>\one@lineR</code>	1
<code>\onlysideX</code>	14
<code>optionadvancedshiftedpstarts</code>	11, 12
<code>optionauxdir</code>	156
<code>optioncontinuousnumberingwithcolumns</code>	9, 156
<code>optionmovecolumnspostiononrightpage</code>	10, 157
<code>optionnomaxlines</code>	11, 12, 24
<code>optionnosyncpstarts</code>	12, 24, 114
<code>optionshiftedpstarts</code>	6, 12, 24
<code>optionwidthliketwocolumns</code>	9

P

<code>\page@action</code>	1
<code>\page@numR</code>	1
<code>\Pages</code>	1, 10
<code>pages (environment)</code>	10
<code>\Pages@mainmatter</code>	1
<code>pairs (environment)</code>	8
<code>\par@lineR</code>	1
<code>\par@patch@pagenumbering</code>	1
<code>\par@patch@thepage</code>	1
<code>\parledgroup@</code>	1
<code>\parledgroup@beforenotes@save</code>	1
<code>\parledgroup@beforenotesL</code>	1
<code>\parledgroup@beforenotesR</code>	1
<code>\parledgroup@correction@notespacing</code>	1
<code>\parledgroup@correction@notespacing@final</code>	1
<code>\parledgroup@correction@notespacing@init</code>	1
<code>\parledgroup@notes@startL</code>	1
<code>\parledgroup@notes@startR</code>	1
<code>\parledgroup@notespacing@correction</code>	1
<code>\parledgroup@notespacing@set@correction</code>	1
<code>\parledgroupseries@</code>	1
<code>\parledgrouptype@</code>	1

\pausenumberingR	1
\pend	18
\pendL	1
\pendR	1
\prev@nopbR	1
\prev@pbR	1
\prevpgstyle	1
\print@columnseparator	1
\print@eledsectionL	1
\print@eledsectionR	1
\print@leftcolumn	1
\print@lineL	1
\print@lineR	1
\print@notesX@forpages	1
\print@rightcolumn	1
\print@Xnotes@forpages	1
\pstart	18
\pstartL	1
\pstartR	1

R

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

S

\save@familiarfootnote@number	1
\save@section@number	1
\section@numR	1
\selectlanguage	1
\set@continuousnumberingforR	1
\set@line	1
\set@line@action	1
\set@sectcountR	1
\setgoalfraction	1, 12
\sethangingsymbol	19
\setline	1
\setlinenum	1
\setnamebox	1

<code>\setnotepositionliketwocolumns@C</code>	1
<code>\setnotepositionliketwocolumns@L</code>	1
<code>\setnotepositionliketwocolumns@R</code>	1
<code>\setpositionliketwocolumns@C</code>	1
<code>\setpositionliketwocolumns@L</code>	1
<code>\setpositionliketwocolumns@R</code>	1
<code>\setRlineflag</code>	17
<code>\setwidthliketwocolumns@C</code>	1
<code>\setwidthliketwocolumns@L</code>	1
<code>\setwidthliketwocolumns@R</code>	1
<code>\sidenote@marginR</code>	1
<code>\sidenotemargin*</code>	1
<code>\sidenotemarginR</code>	1
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 16
<code>\startlock</code>	1
<code>\startsub</code>	1
<code>\sub@action</code>	1
<code>\subline@numR</code>	1
<code>\sublinenumberstyle*</code>	1, 17
<code>\sublinenumberstyleR</code>	1, 17
<code>\sublinenumincrement</code>	1, 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>\thepar@page</code>	1
<code>\thepstartL</code>	16
<code>\thepstartR</code>	16
<code>\thestanzaL</code>	1, 20
<code>\thestanzaR</code>	1, 20

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

v.2.15.0.	
General: Fix bug with \msdata when using multiple \beginnumbering...\endnumbering	1
v0.1.0.	
General: First public release	1
v0.2.0.	
\Columns: Added \l@dchecklang and \l@duselanguage to \Columns	93
\Pages: Added \l@duselanguage to \Pages	103
General: Added section of babel related code	86
Fix babel problems	1
v0.3.0.	
\Pages: Added \ledstrutL to \Pages	103
Added \ledstrutR to \Pages	105
\Rightsidehookend: Added \Leftsidehook, \Leftsidehookend, \Rightsidehook and \Rightsidehookend	56
\affixline@numR: Changed \affixline@numR to match new eledmac	70
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR	68
\do@lineL: Added \do@lineLhook to \do@lineL	62
Simplified \do@lineL by using macros for some common code	62
\do@lineR: Changed \do@lineR similarly to \do@lineL	65
\flag@end: Removed extraneous spaces from \flag@end	48
\ifledRcol: Moved \ifl@dpairing 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	89
\ledstrutR: Added \ledstrutL and \ledstrutR	108
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR	38
General: Added \do@lineLhook and \do@lineRhook	65
Added hooks into Leftside environment	55
Reorganize for ledarab	1
v0.3.a.	
\line@marginR: Do not just set \line@marginR in \linenummargin	35
General: Minor \linenummargin fix	1
v0.3.b.	
\Pages: Added \l@dminpagelines calculation for succeeding page pairs	106
General: Improved parallel page balancing	1
v0.3.c.	
General: Compatibilty with Polyglossia	1
v0.4.0.	
General: No more ledparpatch. All patches are now in the main file.	1
v0.5.0.	
General: Corrections about \section and other titles in numbered sections	1
v0.6.0.	
General: Be able to us \chapter in parallel pages.	1
v0.7.0.	
General: Option ‘shiftedverses’ which make 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 <code>\hangingsymbol</code> to define the character.	1
v0.9.0.	
<code>\ifledRcol</code> : Moved <code>\iflledRcol</code> and <code>\ifnumberingR</code> to <code>eledmac</code>	26
General: Possibility to number <code>\pstart</code>	16
Possibility to number the <code>pstart</code> with the commands <code>\numberpstarttrue</code>	1
v0.9.1.	
General: The numbering of the <code>pstarts</code> restarts on each <code>\beginnumbering</code>	1
v0.9.2.	
General: Debug : with <code>\Columns</code> , the hanging indentation now runs on the left columns and the hanging symbol is shown only when <code>\stanza</code> is used.	1
v0.9.3.	
General: <code>\thepstartL</code> and <code>\thepstartR</code> use now <code>\bfseries</code> and not <code>\bf</code> , which is deprecated and makes conflicts with <code>memoir</code> class.	1
v0.10.0.	
General: <code>\edlabel</code> commands on the right side are now correctly indicated.	1
<code>\edlabel</code> commands which start a paragraph are now put in the right place.	1
v0.11.0.	
<code>\Columns</code> : Line numbering by <code>pstart</code>	94
<code>\affixline@numR</code> : Changed <code>\affixline@numR</code> to allow to disable line numbering (like in <code>eledmac</code> 0.15).	70
<code>\get@nextboxR</code> : Change <code>\get@nextboxL</code> and <code>\get@nextboxR</code> to allow to disable line numbering (like in <code>eledmac</code> 0.15).	114
<code>Pstart</code> number can be printed in side	116
<code>\inserthangingsymbolR</code> : Prevent the column separator for hanging verse from shifting	84
General: Change <code>\do@lineL</code> and <code>\do@lineR</code> to allow line numbering by <code>pstart</code> (like in <code>eledmac</code> 0.15).	62
Lineation can be by <code>pstart</code> (like in <code>eledmac</code> 0.15).	34
New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . .	84
v0.12.0.	
General: New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	84
v1.0.0.	
General: Compatibility with <code>eledmac</code> . Change name to <code>eledpar</code>	1
Debug in lineation by <code>pstart</code>	34
v1.0.1.	
General: Correction on <code>\numberonlyfirstinline</code> with lineation by <code>pstart</code> or by page. .	1
v1.1.0.	
<code>\pstartR</code> : Add <code>\labelpstarttrue</code> (from <code>eledmac</code>).	57
General: Shiftedverses becomes shiftedpstarts.	1
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 . .	83
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.	83
v1.3.4.	
General: Allow use of commands in sidenotes, as introduced by <code>eledmac</code> 1.0.	83
v1.4.0.	
General: Added <code>\do@insidelineLhook</code> and <code>\do@insidelineRhook</code>	65
v1.4.1.	
General: Enable the use of <code>stanzaindentsrepetition</code> within <code>astanza</code> environment.	84
v1.4.3.	
<code>\inserthangingsymbolR</code> : Hanging verse is no longer automatically flush right.	84
<code>\pendL</code> : Spurious spaces in <code>\pendL</code>	60
<code>\pendR</code> : Spurious spaces in <code>\pendR</code>	61
<code>\pstartR</code> : Spurious spaces in <code>\pstartL</code> and <code>\pstartR</code>	57
General: Corrects a false hanging verse when a verse is exactly the length of a line.	1
v1.5.0.	
<code>\sublinenumincrement*</code> : Add starred version of <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> to change both Left and Rightside.	37
General: Add, as in <code>eledmac</code> , features to manage page breaks.	1
v1.6.0.	
General: Add tool and documentation for parallel <code>ledgroups</code>	20
v1.7.0.	
General: Add, as in <code>eledmac</code> , features to make crossrefs with <code>pstart</code> numbers.	1
v1.8.0.	
<code>\Columns</code> : Modify <code>\Columns</code> to enable to add section's title.	91
Suppress <code>\l@dchecklang</code> from <code>\Columns</code>	93
<code>\Pages</code> : Modify <code>\Pages</code> to enable to add section's title.	101
<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.	87
<code>\pendL</code> : As in <code>eledmac</code> , <code>\pendL</code> can have an optional argument.	60
<code>\pendR</code> : As in <code>eledmac</code> , <code>\pendR</code> can have an optional argument.	61
<code>\print@columnseparator</code> : Move some code of <code>\Columns</code> to <code>\print@columnseparator</code>	94
<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>	82
Add <code>\sidenotemargin*</code>	82
<code>\theledlanguageR</code> : Correct left/right language setting with <code>polyglossia</code>	88
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>	83
Add <code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code>	9
Add <code>\columnspostion</code>	9
Add, as in <code>eledmac</code> , new system of sectioning commands.	1
Add, as in <code>eledmac</code> , option to insert something after <code>\pends</code> / verses.	1
Add, as in <code>eledmac</code> , option to insert something between <code>\pstarts</code> / verse.	1

Change <code>\do@lineR</code> and <code>\do@lineR</code> to allow new sectioning commands.	62
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>	21
Suppress <code>\ifl@dsamelang</code> which did not work and was not logical, because both columns could have the same language but not the main language of the document.	86
v1.8.1.	
<code>\do@lineL</code> : Fix a bug with critical notes at the beginning of a page, (maybe added by v1.8.0) (?).	62
<code>\do@lineR</code> : Fix a bug with critical notes at the beginning of a page, added by v1.8.0 (?).	65
v1.8.2.	
<code>\flag@end</code> : <code>\flag@start</code> and <code>\flag@end</code> are now defined only one time for <code>eledmac</code> and <code>eledpar</code>	48
<code>\lineation*</code> : Add <code>\lineation*</code>	35
<code>\reledpar@error</code> : Errors specific to <code>eledpar</code> send to <code>eledpar</code> handbook	26
General: Debug <code>\eledxxx</code> with some paper sizes	1
Debug left and side note (bugs added by 1.8.0)	1
v1.8.3.	
<code>\Pages</code> : Debug blank pages when using optional argument in the last <code>\pend</code>	101
<code>\doinsidelineRhook</code> : Added <code>\dolineLhook</code> , <code>\dolineRhook</code> , <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code>	65
<code>\resumenumberingR</code> : Debug <code>\resumenumberingR</code>	33
General: Add <code>\noeledxxx</code> , as in <code>eledmac</code>	1
v1.9.0.	
<code>\ifwidthliketwocolumns</code> : Added <code>widthliketwocolumns</code> option	25
<code>\theledlanguageR</code> : Debug left/right language switching with <code>polyglossia</code> . Do not write in <code>.aux</code> file when setting left/right lines.	88
General: Add <code>\AtBeginPairs</code> macro.	8
Compatibility with <code>\Xnoteswidthliketwocolumns</code> and <code>\notesXwidthliketwocolumns</code>	1
v1.9.1.	
<code>\ifledRcol</code> : Moved <code>\ifl@dpadding</code> to <code>eledmac</code>	26
v1.10.0.	
<code>\Pages</code> : Debug wrong pages splitting when no optional argument is used in last <code>\pend</code> (bug was added in v1.8.3).	101
Debug wrong parallel pages synchronization when an <code>\edtext</code> falls across two pages.	101
General: Compatibility with <code>\AtEveryPstart</code> and <code>\AtEveryPend</code>	1
Restore critical notes in <code>\eledsection</code> in parallel columns (this bug was added in 1.8.2).	1
v1.10.1.	
<code>\line@list@stuffR</code> : Revert modification of 1.4.2, which makes bugs with numbering. Leave vertical mode to solve spurious space before <code>minipage</code>	48
v1.11.0.	
<code>\edtext</code> : <code>\critext</code> and <code>\edtext</code> are now defined only in <code>eledmac</code>	49
General: Compatibility of standard footnotes with some <code>biblatex</code> styles.	1
v1.12.0.	
<code>\Columns</code> : Add <code>\l@dprintingcolumnstrue</code>	91
<code>\Pages</code> : Add <code>\l@dprintingpagestrue</code>	101
<code>\edlabel</code> : <code>\edlabel</code> and <code>\edindex</code> works now with <code>hyperref</code> when using <code>eledpar</code>	81

\edlabel is now defined only one time for both eledmac and eledpar	81
\print@eledsectionL: Compatibility with Lua \TeX RTL languages.	64
\print@eledsectionR: Compatibility with Lua \TeX RTL languages.	67
\print@lineL: Compatibility with Lua \TeX RTL languages.	64
General: Compatibility with Lua \TeX RTL languages.	1
v1.12.1.	
\print@eledsectionL: Fixes bug with Lua \TeX RTL \eledsection.	64
v1.13.0.	
\Pages: Prevent false overfull hboxes when using \Pages outside of pages environment.	101
When using shiftedpstarts option, a \l@leftbox with a null height will advance the \pagetotal in any case.	101
\clearl@drighthouse: Use \newpage instead of \clearpage.	109
\ifledRcol: Remove false boolean settings which are not needed.	26
General: Enable the use of optional argument of & in astanza environment.	84
Fix bug in shiftedpstarts when size difference between pstarts is very important.	1
With parallel pages, long notes can now flow from the Left to the right side and from the Right to the left side.	1
v1.13.1.	
\Pages: Prevent false empty page after \Pages (bug added in 1.13.0)	101
\correct@footinsX@box: Call \correct@footinsX@box and \correct@Xfootins@box directly in \print@notesX@forpages and \print@Xnotes@forpages.	76
Correct \correct@footinsX@box and \correct@Xfootins@box	76
v1.14.0.	
General: Fix bug with line number position when using \eledsection and similar commands for RTL texts with Lua \TeX .	1
The \newifs are not followed by boolean values set to false, because it is the \TeX default setting.	1
v1.15.0.	
\do@actions@nextR: Add actions 1008 and 1009	68
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting when a verse is hanging	84
\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
General: Add \AtEveryPstartCall.	1
Add sameparallelpagenumber option.	12
Fix vertical spurious space before right \eledchapter (bug added in v1.13.0).	1
Prevent vertical space when using \AtEveryPstart or \AtEveryPend with a command which prints nothing	1
v1.16.0.	
\newseries@par: Fix bug with \onlysideX.	49
General: Error message when calling \Pages inside 'pages' environment and \Columns inside 'pairs' environment.	1
Error message when starting a Leftside/a Rightside while the previous one has not been yet typeset.	1
Error message when using \beginnumbering...\endnumbering without \pstart.	1

Fix bug with <code>nofamiliar / nocritical</code> option of <code>eledmac</code> .	1
New package option <code>sameparallelpagenum</code> to have the same page number for both left and right side.	1
v1.16.1.	
General: Write information about line-list file version in the correct file.	1
v1.16.2.	
General: Fix bug when adding empty lines before a <code>\pend</code> in combination with some specific penalties setting.	1
v1.17.0.	
General: Add compatibility of optional argument of <code>\pstart/\pend</code> and <code>\AtEveryPstart/\AtEveryPend</code> with two columns mode.	1
v1.21.0.	
General: Add <code>\hidenum</code>	17
v2.0.0.	
<code>\@adv</code> : <code>\@adv</code> defined only in <code>reledmac</code> .	42
<code>\@lab</code> : <code>\@lab</code> defined only in <code>eledmac</code> .	82
<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> .	48
<code>\bbl@set@language</code> : Patch <code>\bbl@set@language</code> instead of redefining it	87
<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> .	48
<code>\fix@page</code> : <code>\fix@page</code> is defined only once in <code>reledmac</code>	42
<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> .	43
<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
<code>\theledlanguageR</code> : Patch <code>\otherlanguage</code> instead of redefining it.	88
General: <code>\@nl</code> is now defined only in <code>reledmac</code> .	41
<code>\ifbypage@</code> and <code>\ifbypstart@R</code> defined in <code>eledmac</code> .	34
Fix some bugs with ‘ <code>sameparallelpagenum</code> ’ option.	1
Many code refactored and moved to <code>reledmac</code> .	1
Package’s name becomes <code>reledpar</code> .	1
Totally new implementation of ‘ <code>sameparallelpagenum</code> ’ option.	1
<code>chapterinpages</code> : Deleting the old system of managing parallel chapter, keep only the new one with <code>\patchcmd</code> .	55
v2.1.0.	
General: Fix bug when using <code>\eledsection</code> and related on right pages when page width is short.	1
Fix bug when using <code>\pagenumbering</code> with <code>memoir</code> (bug added in v2.0.0).	1

Fix bug with <code>\setparledgroupnotespacing</code> with the <code>shiftedpstarts</code> option.	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code>	1
Options to custom empty right page before <code>\Pages</code>	1
v2.2.0.	
General: <code>astanza</code> environment can take an optional argument, which will be the optional argument of <code>\pstart</code> started by this environment.	1
New tools to number stanza	1
v2.2.1.	
General: Fix bug with optional argument of last left <code>\pend</code>	1
v2.3.0.	
<code>\Pages</code> : Fix bug when calling <code>\Columns</code> after a <code>\Pages</code> (bug added in v1.13.0).	101
General: Change some internal codes in order to provide compatibility with \TeX release of october 2015	1
Fix bug with title number in parallel columns	1
New line setting command suffixed by R to set only the right side.	1
v2.4.0.	
<code>\ledstrutR</code> : Deleted <code>\ledstrutL</code> and <code>\ledstrutR</code>	108
Fix bug with dotted letter	108
General: New way of (not) synchronizing the parallel pages.	1
Option to switch to <code>\mainmatter</code> when calling <code>\Pages</code>	1
v2.5.0.	
General: Disable empty lines as paragraph in <code>astanza</code>	1
Fix bug introduced in v1.15.0 which made hanging indentation in verse not work anymore.	1
New commands <code>\linenummarginR</code> and <code>\linenummargin*</code>	1
v2.5.1.	
General: Fix spurious space when using optional argument of <code>astanza</code> environment (introduced in v2.5.0).	1
v2.5.2.	
General: Fix bug introduced in v2.5.0 with <code>\linenummargin</code> , <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code>	1
v2.6.0.	
<code>\l@dmake@labelsR</code> : <code>\@Rlineflag</code> is not stored directly after the line number, but as a fifth argument of <code>\the@labelX</code> . Can be retrieved by <code>\xflagref</code>	81
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 bug with footnote numbering when using the same series of familiar footnotes on both sides.	1
Fix bug with right side title number when using title commands before <code>pages</code> or <code>columns</code> environments.	1
Fix compatibility with <code>babel</code> (broken in v2.0.0).	1
No error messages about ends of left / right page when using the <code>\syntaxonly</code> command of the <code>syntonly</code> package.	1
v2.6.1.	
General: Fix bug, introduced in v2.6.0, with footnote numbering when using <code>perpage</code> package.	1
v2.6.2.	
<code>\newseries@par</code> : The \TeX counter <code>\footnoteX@reading</code> is defined in <code>reledmac</code> . . .	50

General: Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.6.0).	1
Fix bug (added in v2.6.0) with footnote numbering in parallel typesetting while using <code>polyglossia</code> with specific numbering systems (like Greek).	1
v2.6.3.	
General: Fix spurious dot when using <code>\linenummargin</code> on right side (introduced in v2.5.0).	1
v2.7.0.	
General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package.	1
v2.7.1.	
General: Fix bug added in <code>reledmac</code> 2.8.2, when typesetting parallel text just after a sectioning command	1
v2.8.0.	
General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> options.	1
Add <code>\linenumberLevenifblanktrue</code> and <code>\linenumberRevenifblank</code> commands	1
Fix bug when the right line number style is not the same to the left line number style	1
v2.9.0.	
General: Add <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> .	1
More specific error messages.	1
v2.9.1.	
General: Prevent <code>\Xtxtbeforenotes</code> hook from causing notes to go beyond the bottom margin	1
v2.10.0.	
<code>\do@actions@nextR</code> : Add action 1010	68
General: Add new tools to make apparatuses of manuscripts	1
v2.11.0.	
<code>\correct@footinsX@box</code> : Clarification in the handbook about the use of <code>\Xonlyside</code> and <code>onlysideX</code> and error message if misuse.	76
New implementation of <code>\Xonlyside</code> and <code>\onlysideX</code> hooks, prevent trouble with vertical spacing.	76
General: Compatibility with <code>reledmac</code> 's <code>auxdir</code> option.	1
v2.12.0.	
General: Add <code>\edtextlater</code> and <code>\edtextnow</code> .	1, 79
Fix bug with <code>\AtBeginPairs</code> .	1
Fix bug with <code>\lineationR</code> .	1
v2.13.0.	
General: Compatibility with the new features of <code>reledmac</code> 2.15.0	1
v2.13.1.	
General: Simplification of the code concerning the sectioning command	1
v2.14.0.	
General: Add <code>\linenumberlistR</code> , equivalent to <code>\linenumberlist</code> for the right side.	1
Error message when left / right side are not defined.	1
More explicit error message when the stanza indentation is not defined.	1
New commands to have specific line margins for text running in parallel columns	1
When indexing texts in sidenotes with <code>\edtext</code> , referring to the line number where the sidenote is called.	1

v2.14.1.	
General: Fix bug when using <code>\markboth</code> on <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code>	1
Fix bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.13.1).	1
v2.14.2.	
General: Fix spurious space with <code>\edtextlater</code>	1
v2.14.3.	
General: Fix bug in <code>\edtextnow</code>	1
v2.14.4.	
General: Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code>	1
v2.15.0.	
General: Add <code>movecolumnspostiononrightpage</code> option.	1
Fix bug about space before sectioning commands after a <code>pairs</code> environment (bug added in v. 2.14.4).	1
Fix bug with page break when using <code>\columns</code> (bug added in v. 2.14.1).	1
New macro <code>\sidenotemarginR</code> , fix default value to right.	1
v2.16.0.	
General: Compatibility with <code>\setmsdataposition</code>	1
v2.16.1.	
General: Restore sidenotes on left side, deleted by mistake on v2.16.0	1
v2.16.2.	
General: Change log message when numbered files don't still exist, in order to improve compatibility with <i>latexmk</i>	1
v2.16.3.	
General: Fix bug when setting <code>\aftercolumnseparator</code>	1