

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 v2.9.0, last revised 2016/05/08.

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

II Preliminaries	21
II.1 Package's meta-data	21
II.2 Package's requirement	21
II.3 Package's options	21
II.4 Package's options	22
II.4.1 Synchronization's options	22
II.4.2 Other options	23
II.5 Determining side and category of parallel processing	23
II.6 Text's width	24
II.7 Messages	24
III Sectioning commands	26
IV Line counting	30
IV.1 Setting lineation reset	30
IV.2 Setting line number margin	31
IV.3 Setting lineation start and step	32
IV.4 Setting line flag	33
IV.5 Setting line number style	33
IV.6 Print marginal line number	34
IV.7 Line-number counters and lists	35
IV.7.1 Correspond to those in <code>reledmac</code> for regular or left text	35
IV.7.2 Specific to <code>reledpar</code>	35
IV.8 Reading the line-list file	36
IV.9 Commands within the line-list file	36
IV.10 Writing to the line-list file	42
V Marking text for notes	44
V.1 Specific hooks and commands for notes	44
V.1.1 Notes to be printed on one side only	44
V.2 Tools specific to familiar footnotes	45
V.2.1 Managing correct number	45
V.2.2 Familiar footnotes without marks	45
V.2.3 Get correct footnote number	47
V.3 Create hooks	47
V.4 Init standards series (A,B,C,D,E,Z)	48
V.5 Tools specific to \LaTeX 's classical footnotes	48
VI Pstart numbers dumping and restoration	48
VII Parallel environments	49
VIII Paragraph decomposition and reassembly	52
VIII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	52
VIII.2 Processing one line	57
VIII.3 Line and page number computation	61
VIII.4 Line number printing	64

VIII.5 Pstart number printing in side	66
VIII.6 Add insertions to the vertical list	68
VIII.7 Penalties	69
VIII.8 Printing leftover notes	70
IX Footnotes	70
IX.1 Footnotes output specific to \Pages	70
X Cross referencing	74
XI Side notes	75
XII Familiar footnotes	76
XIII Verse	76
XIV Naming macros	79
XV Fixing babel and polyglossia	79
XVI Counts and boxes for parallel texts	81
XVII Checking text to be processed	83
XVIII Parallel columns	85
XIX Parallel pages	93
XIX.1 Specific counters	93
XIX.2 Main macro	93
XIX.3 Ensure all notes be printed at the end of parallel pages	99
XIX.4 Struts	100
XIX.5 Page clearing	100
XIX.6 Lines managing	101
XIX.7 Page break managing	103
XIX.8 Getting boxes content	106
XX Page numbering	110
XX.1 Global options	110
XX.2 mainmatter option of \Pages	112
XXI Sections' titles' commands	112
XXII Page break/no page break, depending on the specific line	113
XXIII Parallel ledgroup	114
XXIV Compatibility with eledmac	118

XXV The End	118
Appendix A Some things to do when changing version	119
Appendix A.1 Migration to eledpar 1.4.3	119
Appendix A.2 Migration from eledpar to reledpar	119
Appendix A.2.1 Deprecated options	119
Appendix A.2.2 \renewcommandreplaced with command	119
Appendix A.2.3 Commands the names of which have changed	120
Appendix A.3 Migration to reledpar 2.2.0	120
Appendix A.4 Migration to reledpar 2.3.0	120
Appendix A.5 Migration to reledpar 2.4.0	120
Appendix A.6 Migration to reledpar 2.5.0	120
Appendix A.7 Migration to reledpar 2.6.0	120
Appendix A.8 Migration to reledpar 2.6.1	120
References	121
Index	121
Change History	139

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 XXV); and an Index to the source code. As `reledpar` is an adjunct to `reledmac` we assume that you have read the `reledmac` manual. Also `reledpar` requires `reledmac` to be used, in the version distributed with version.

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

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

1.2 Historical overview

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

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

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

2 Options

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

2.1 Synchronization's options

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

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

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

nomaxlines allows facing pages to have different numbers of lines.

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

2.2 Other options

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

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

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

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

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

3 General

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

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

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

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

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

```
\maxchunks{5120}
```

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

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

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

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

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

in this respect.

4 Parallel columns

4.1 Basic use

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

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

```
\begin{pairs}
\begin{Leftside} reledmac numbering text command \end{Leftside}
\begin{Rightside} reledmac numbering text command \end{Rightside}
\end{pairs}
\Columns
\begin{pairs}
\begin{Leftside} reledmac numbering text command \end{Leftside}
...
\end{pairs}
\Columns
```

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

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

There is no required pagebreak before or after the columns.

4.2 Setting

4.2.1 Column's width

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

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

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

4.2.2 Column's separator

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

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

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

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

4.2.3 Column's positions

`\columnspan`

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

5.1 Basic usage

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

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

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

```
\begin{pages}
\begin{Leftside} reledmac numbering text command \end{Leftside}
\begin{Rightside} reledmac numbering text command \end{Rightside}
\begin{Leftside} reledmac numbering text command \end{Leftside}
...
\end{pages}
\Pages
```

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

5.2 Setting

5.2.1 Text width

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

5.2.2 Way of synchronizing⁴

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

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

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

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

The aim of the default setting is twofold:

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

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

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

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

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

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

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

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

5.2.3 Page number

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

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

Please also read advising in 11 p. 20.

5.2.4 Page breaking

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

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

5.2.5 Right page before \Pages

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

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

`prevpgstyle=<style>` in order to set the style of this page. A common value of `<style>` is empty. Use `prevpgstyle=empty` will suppress header and footer in this page. Please also read advising in 11 p. 20.

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

5.2.6 Notes about \mainmatter

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

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

5.3 Critical and familiar footnotes

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

5.3.1 Notes height setting

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

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

So, you MUST change the default setting via `\Xmaxhnotes` (for critical notes) and `\maxhnotesX` (for familiar notes). Both commands are explained in `reledmac` handbook (7.13.5 p. 41). As an advisable setting:

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

5.3.2 About the numbering of familiar footnotes

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

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

5.3.3 Using `perpage` package

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

So do not set:

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

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

```
\MakePerPage{footnoteB}
```

But set:

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

5.3.4 Notes for one side only

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

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

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

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

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

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

5.4 Using line flag

`\Xlineflag` Use `\Xlineflag[⟨s⟩]` to add right line flag (6.3 p. 16) to right critical footnotes and
`\Xendlineflag` `\Xendlineflag[⟨s⟩]` to add it to right critical endnotes.

6 Left and right texts

6.1 Environments

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

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

6.2 Numbering text lines and paragraphs

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

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

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

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

\memorydump The command \memorydump effectively performs an \endnumbering immediately followed by a \beginnumbering while not restarting the numbering sequence. This has the effect of clearing T_EX'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}

```

```

\numberpstarttrue
\numberpstartfalse
  \thepstartL
  \thepstartR
\skipnumbering
\hidenumbering

```

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

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

6.3 Lineation system

```

\firstlinenum
\linenumincrement
\firstsublinenum
\sublinenumincrement

```

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

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

```

\firstlinenum*
\linenumincrement*
\firstsublinenum*
\sublinenumincrement*
\firstlinenumR
\linenumincrementR
\firstsublinenumR
\sublinenumincrementR
\lineationR
\lineation*
\linenumberstyleR
\sublinenumberstyleR
\linenumberstyle*
\sublinenumberstyle*
\linenummarginR
\linenummargin*
\setRlineflag
\linenumberLevenifblanktrue
\linenumberRevenifblanktrue

```

The starred versions change both left and right numbering schemes.

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

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

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

`\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.

`\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`. 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

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

6.4 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).

6.5 \AtEveryPstart and \AtEveryPstartCall

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

Consequently:

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

6.6 Language setting

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

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

7 Verse

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

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

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

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

If you get an error message along the lines of ‘Missing number, treated as zero `\szi@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` `\thestanzaR` When using `\stanzanumtrue` (9.9 p. 45) in parallel typesetting, stanza counter is replaced by `stanzaL` counter in left side and by `stanzaR` counter in right side. Consequently, you can redefine `\thestanzaL` and `\thestanzaR` to change their aspect.

8 Side notes

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

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

9 Parallel ledgroups

9.1 General

You can also make parallel ledgroups (see the documentation of `reledmac` about ledgroups, 10 p. 46). To do it you have:

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

See the following example:

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

```
\end{pages}
\Pages
```

9.2 Parallel ledgroups and setspace package

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

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

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

10 Sectioning commands

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

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

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

11 Notes about page number

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

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

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

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

I Implementation overview

\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}[2016/05/08 v2.9.0 reledmac extension for
  parallel texts]%
4
5 %

```

II.2 Package’s requirement

Few commands use `\xspace` command.

```

6 \RequirePackage{xspace}%
7 %

```

II.3 Package’s options

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

```

8 \RequirePackage{xkeyval}
9 %

```

II.4 Package's options

II.4.1 Synchronization's options

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

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

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

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

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

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

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

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

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

```

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

```

II.4.2 Other options

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

```

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

```

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

```

41 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
42 \DeclareOptionX{continuousnumberingwithcolumns}{\
43   continuousnumberingwithcolumnstrue}%
44 %

```

`\ifsameparallelpagenunder` Options related to page numbering

`\ifprevpgnotnumbered`

```

44 \newif\ifsameparallelpagenunder
45 \newif\ifprevpgnotnumbered
46 \DeclareOptionX{sameparallelpagenunder}{\sameparallelpagenundertrue}
47 \DeclareOptionX{prevpgnotnumbered}{\prevpgnotnumberedtrue}
48 %

```

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

```

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

```

```

51 \ProcessOptionsX%
52 %

```

II.5 Determining side and category of parallel processing

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

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

II.6 Text's width

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

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

II.7 Messages

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

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

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

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

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

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



```

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

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

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

@error@fail@patch@thepage97 \newcommand{\led@error@fail@patch@thepage}{%
98   \reledpar@error{Fail to patch \string\@thepage\space command.}{\@ehc}%
99 }%
100 %

@fail@patch@pagenumbering01 \newcommand{\led@error@fail@patch@pagenumbering}{%
102   \reledpar@error{Fail to patch \string\pagenumbering\space command.}{\@ehc
}%
103 }%
104 %

error@fail@patch@@mempnum05 \newcommand{\led@error@fail@patch@@mempnum}{%
106   \reledpar@error{Fail to patch \string@mempnum\space command.}{\@ehc}%
107 }%
108 %

or@fail@patch@@outputpage09 \newcommand{\led@error@fail@patch@@outputpage}{%
110   \reledpar@error{Fail to patch \string\@outputpage\space command.}{\@ehc}%
111 }%
112 %

```

```

\led@warn@ChangeSyncOption13 \newcommand*\led@warn@ChangeSyncOption}[1]{%
114 \reledpar@warning{You have changed synchronization's options since last
run. We have not read line-list file #1. Please run LaTeX again.}%
115 }%
116 %

```

```

\led@warn@setting@in@rightside17 \newcommand{\led@warn@setting@in@rightside}[1]{%
118 \reledpar@warning{You use #1 inside rightside environment.\MessageBreak%
119 Such behavior is deprecated.\MessageBreak%
120 Use instead #1R or #1* in your preamble.}%
121 }
122 %

```

```

\led@error@missing@numbering23 \newcommand{\led@error@missing@numbering}[1]{%
124 \reledpar@error{Missing \string\...pstart\string\pend\space inside `#1`
environment}{\@ehc}%
125 }%
126 %

```

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\section@num`, where `nn` is the section number. However, for right side texts the file is called `\jobname\section@numR`. The `\extensionchars` applies to the right side files just as it does to the normal files.

```

127 \newcount\section@numR
128 \section@numR=\z@
129 %

```

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

```

130 \pst@rtedLfalse
131 \newif\ifpst@rtedR
132
133 %

```

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

```

134 \newcommand*\beginnumberingR}{%
135 \ifnumberingR
136 \led@err@NumberingStarted

```

```

137 \endnumberingR
138 \fi
139 \global\l@dnumstartsR \z@
140 \global\pst@rtedRfalse
141 \global\numberingRtrue
142 \global\advance\section@numR \@ne
143 \global\absline@numR \z@
144 \gdef\normal@page@breakR{}
145 \gdef\l@prev@pbR{}
146 \gdef\l@prev@nopbR{}
147 \global\line@numR \z@
148 \global\@lockR \z@
149 \global\sub@lockR \z@
150 \global\sublines@false
151 \global\let\next@page@numR\relax
152 \global\let\sub@change\relax
153 \set@continuousnumberingforR%
154 \message{Section \the\section@numR R }%
155 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
156 \l@dend@stuff
157 \setcounter{pstartR}{1}
158 \begingroup
159 \initnumbering@sectcountR
160 \gdef\eled@sectionsR@{ }%
161 \if@noeled@sec\else%
162 \makeatletter\inputIfFileExists{\jobname.eledsec\the\section@numR R
163 }{ }{\makeatother}%
164 \immediate\openout\eled@sectioningR@out=\jobname.eledsec\the\
165 section@numR R\relax%
166 \fi%
167 }
168 %

```

\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.

```

167 \def\endnumberingR{%
168 \ifnumberingR
169 \global\numberingRfalse
170 \normal@pars
171 \ifnum\l@dnumstartsR=0%
172 \led@err@NumberingWithoutPstart%
173 \fi%
174 \ifl@dpairing
175 \global\pst@rtedRfalse
176 \else

```

```

177 \ifx\insertlines@listR\empty\else
178 \global\noteschanged@true
179 \fi
180 \ifx\line@listR\empty\else
181 \global\noteschanged@true
182 \fi
183 \fi
184 \ifnoteschanged@
185 \led@mess@NotesChanged
186 \fi
187 \else
188 \led@err@NumberingNotStarted
189 \fi
190 \endgroup
191 \if@noeled@sec\else%
192 \immediate\closeout\eled@sectioningR@out%
193 \fi%
194 }
195
196 %

```

`\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`

```

197 \newcounter{chapterR}
198 \newcounter{sectionR}
199 \newcounter{subsectionR}
200 \newcounter{subsubsectionR}
201
202 \newcount\old@chapter%
203 \newcount\old@section%
204 \newcount\old@subsection%
205 \newcount\old@subsubsection%
206 \newcommand{\save@section@number}{%
207 \ifdefined\c@chapter%
208 \global\old@chapter\value{chapter}%
209 \fi%
210 \global\old@section\value{section}%
211 \global\old@subsection\value{subsection}%
212 \global\old@subsubsection\value{subsubsection}%
213 }%
214 \newcommand{\initnumbering@sectcountR}{
215 \ifdefined\c@chapter%
216 \setcounter{chapterR}{\old@chapter}%
217 \fi%
218 \setcounter{sectionR}{\old@section}%
219 \setcounter{subsectionR}{\old@subsection}%
220 \setcounter{subsubsectionR}{\old@subsubsection}%
221 \set@sectcountR%

```

```

222 }
223 \newcommand{\set@sectcountR}{%
224   \let\c@chapter\c@chapterR%
225   \let\c@section\c@sectionR%
226   \let\c@subsection\c@subsectionR%
227   \let\c@subsubsection\c@subsubsectionR%
228 }%
229 %

```

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

```

230 \newcommand*{\pausenumberingR}{%
231   \endnumberingR\global\numberingRtrue}
232 \newcommand*{\resumenumberingR}{%
233   \ifnumberingR
234     \global\pst@rtedRtrue
235     \global\advance\section@numR \@ne
236     \led@mess@SectionContinued{\the\section@numR R}%
237     \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
238     \l@dend@stuff
239     \begingroup%
240     \initnumbering@sectcountR%
241     \set@continuousnumberingforR%
242   \else
243     \led@err@numberingShouldHaveStarted
244     \endnumberingR
245     \beginnumberingR
246   \fi}
247
248 %

```

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

```

249 \newcommand*{\memorydumpL}{%
250   \endnumbering
251   \numberingtrue
252   \global\pst@rtedLtrue
253   \global\advance\section@num \@ne
254   \led@mess@SectionContinued{\the\section@num}%
255   \line@list@stuff{\jobname.\extensionchars\the\section@num}%
256   \l@dend@stuff}
257
258 \newcommand*{\memorydumpR}{%
259   \endnumberingR
260   \numberingRtrue
261   \global\pst@rtedRtrue
262   \global\advance\section@numR \@ne
263   \led@mess@SectionContinued{\the\section@numR R}%
264   \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%

```

```

265 \l@dend@stuff}
266
267 %

```

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`.

```

268 \newcommand*{\lineationR}[1]{%
269   \ifnumbering
270     \led@err@LineationInNumbered
271   \else
272     \def\@tempa{#1}\def\@tempb{page}%
273     \ifx\@tempa\@tempb
274       \global\bypage@Rtrue
275       \global\bypstart@Rfalse
276       \unless\ifnocritical@%
277         \Xpstart[] [false]%
278       \fi%
279     \else
280       \def\@tempb{pstart}%
281       \ifx\@tempa\@tempb
282         \global\bypage@Rfalse
283         \global\bypstart@Rtrue
284         \unless\ifnocritical@%
285           \Xpstart%
286         \fi%
287       \else
288         \def\@tempb{section}
289         \ifx\@tempa\@tempb
290           \global\bypage@Rfalse%
291           \global\bypstart@Rfalse%
292           \unless\ifnocritical@%
293             \Xpstart[] [false]%
294           \fi%
295         \else
296           \led@warn@BadLineation
297         \fi%
298       \fi
299     \fi
300   \fi}}

```

```

301 %
\set@continuousnumberingforR \set@continuousnumberingforR set the right line numbers at a \beginnumberingR
or a \resumenumberingR in order to have continuous numbering with single column
text.
302 \newcommand{\set@continuousnumberingforR}{%
303   \ifcontinuousnumberingwithcolumns%
304     \ifnum\line@numR<\line@num%
305       \expandafter\setlinenum\expandafter{\the\line@num}%
306     \fi%
307     \ifnum\last@page@num>\last@page@numR%
308       \global\last@page@numR=\last@page@num%
309     \fi%
310   \fi%
311 }
312 %

```

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

```

313 \WithSuffix\newcommand\lineation*[1]{%
314   \lineation{#1}%
315   \lineationR{#1}%
316 }%
317 %

```

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`.

```

318 \newcount\line@marginR
319 %

```

By default put right text numbers at the right.

```

320 \line@marginR=\@ne
321
322 %

```

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

```

323 \newcommand{\linenummarginR}[1]{%
324   \l@getline@margin{#1}%
325   \ifnum\@l@tempcntb>\m@ne%
326     \global\line@marginR=\@l@tempcntb%
327   \fi%
328 }
329 \WithSuffix\newcommand\linenummargin*[1]{%
330   \l@getline@margin{#1}%
331   \ifnum\@l@tempcntb>\m@ne%
332     \global\line@marginR=\@l@tempcntb%
333     \global\line@margin=\@l@tempcntb%
334   \fi%
335 }
336 %

```

IV.3 Setting lineation start and step

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

```

337 \newcounter{firstlinenumR}
338   \setcounter{firstlinenumR}{5}
339 \newcounter{linenumincrementR}
340   \setcounter{linenumincrementR}{5}
341 %

```

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

```

342 \newcounter{firstsublinenumR}
343   \setcounter{firstsublinenumR}{5}
344 \newcounter{sublinenumincrementR}
345   \setcounter{sublinenumincrementR}{5}
346
347 %

```

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

```

\linenumincrement
\firstsublinenum
\sublinenumincrement
\firstlinenum*
\linenumincrement*
\firstsublinenum*
\sublinenumincrement*
348 \WithSuffix\newcommand\firstlinenum*[1]{%
349   \setcounter{firstlinenumR}{#1}%
350   \setcounter{firstlinenum}{#1}%
351 }
352 \WithSuffix\newcommand\linenumincrement*[1]{%
353   \setcounter{linenumincrementR}{#1}%
354   \setcounter{linenumincrement}{#1}%

```



```

355 }
356 \WithSuffix\newcommand\firstsublinenum*[1]{%
357   \setcounter{subfirstlinenumR}{#1}%
358   \setcounter{subfirstlinenum}{#1}%
359 }
360 \WithSuffix\newcommand\sublinenumincrement*[1]{%
361   \setcounter{sublinenumincrementR}{#1}%
362   \setcounter{sublinenumincrement}{#1}%
363 }
364 %

```

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

```

\linenumincrementR
\firstsublinenumR
\sublinenumincrementR
365 \newcommand\firstlinenumR[1]{%
366   \setcounter{firstlinenumR}{#1}%
367 }
368 \newcommand\linenumincrementR[1]{%
369   \setcounter{linenumincrementR}{#1}%
370 }
371 \newcommand\firstsublinenumR[1]{%
372   \setcounter{subfirstlinenumR}{#1}%
373 }
374 \newcommand\sublinenumincrementR[1]{%
375   \setcounter{sublinenumincrementR}{#1}%
376 }
377 %

```

IV.4 Setting line flag

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

```

378 \newcommand{\setRlineflag}[1]{%
379   \gdef\Rlineflag{#1}%
380 }
381 \setRlineflag{R}
382 %

```

IV.5 Setting line number style

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

```

383 \newcommand*\linenumrepR[1]{\@arabic{#1}}
384 \newcommand*\sublinenumrepR[1]{\@arabic{#1}}
385
386 %

```

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

`\sublinenumbersstyleR`

```

387 \newcommand*\linenumberstyleR}[1]{%
388   \def\linenumrepR##1{\@nameuse{@#1}{##1}}
389 \newcommand*\sublinenumberstyleR}[1]{%
390   \def\sublinenumrepR##1{\@nameuse{@#1}{##1}}
391 %

```

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

```

392 \WithSuffix\newcommand\linenumberstyle*[1]{%
393   \linenumberstyle{#1}%
394   \linenumberstyleR{#1}%
395 }%
396
397 \WithSuffix\newcommand\sublinenumberstyle*[1]{%
398   \sublinenumberstyle{#1}%
399   \sublinenumberstyleR{#1}%
400 }%
401 %
402 %

```

IV.6 Print marginal line number

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

```

403 \newif\iflinenumberLevenifblank
404 \newif\iflinenumberRevenifblank
405 %

```

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

```

406 \newcommand*\leftlinenumR{%
407   \l@dlinenumR
408   \kern\linenumsep}
409 \newcommand*\rightlinenumR{%
410   \kern\linenumsep
411   \l@dlinenumR}
412 \newcommand*\l@dlinenumR{%
413   \numlabfont\linenumrepR{\line@numR}\@Rlineflag%
414   \ifsublines@
415     \ifnum\subline@num>\z@
416       \unskip\fullstop\sublinenumrepR{\subline@numR}%
417     \fi
418   \fi}
419
420 %

```

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.

```
421 \newcount\line@numR
422 \newcount\subline@numR
423 \newcount\absline@numR
424
425 %
```

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

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

```
426 \list@create{\line@listR}
427 \list@create{\insertlines@listR}
428 \list@create{\actionlines@listR}
429 \list@create{\actions@listR}
430
431 %
```

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

```
432 \newcount\page@numR
433
434 %
```

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.

```
435 \list@create{\linesinpar@listL}
436 \list@create{\linesinpar@listR}
437 \list@create{\maxlinesinpar@list}
438
439 %
```

IV.8 Reading the line-list file

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

```

440 \newcommand{\list@clearing@regR}{%
441   \list@clear{\line@listR}%
442   \list@clear{\insertlines@listR}%
443   \list@clear{\actionlines@listR}%
444   \list@clear{\actions@listR}%
445   \list@clear{\linesinpar@listR}%
446   \list@clear{\linesonpage@listR}
447 }
448 %

```

`\@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.

```

449 \newcommand{\@par@sync@option}[1]{%
450   \IfStrEq{#1}{\@par@this@sync@option}%
451   {}%
452   {\ifledRcol%
453     \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}%
454     %
455     \else%
456     \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}%
457     %
458     \fi%
459     \endinput%
460   }%
461 }%
462 %

```

`\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.

```

461 \newcommand{\@nl@regR}{%
462   \ifx\l@dchset@num\relax \else
463   \advance\absline@numR \@ne

```

```

464 \set@line@action
465 \let\l@dchset@num\relax
466 \advance\absline@numR \m@ne
467 \advance\line@numR \m@ne% % do we need this?
468 \fi
469 \advance\absline@numR \@ne
470 \ifx\next@page@numR\relax \else
471 \page@action
472 \let\next@page@numR\relax
473 \fi
474 \ifx\sub@change\relax \else
475 \ifnum\sub@change>\z@
476 \sublines@true
477 \else
478 \sublines@false
479 \fi
480 \sub@action
481 \let\sub@change\relax
482 \fi
483 \ifcase\@lockR
484 \or
485 \@lockR \tw@
486 \or\or
487 \@lockR \z@
488 \fi
489 \ifcase\sub@lockR
490 \or
491 \sub@lockR \tw@
492 \or\or
493 \sub@lockR \z@
494 \fi
495 \ifsublines@
496 \ifnum\sub@lockR<\tw@
497 \advance\subline@numR \@ne
498 \fi
499 \else
500 \ifnum\@lockR<\tw@
501 \advance\line@numR \@ne \subline@numR \z@
502 \fi
503 \fi}
504
505
506 %

```

`\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`.

```

507 \newcount\last@page@numR
508 \last@page@numR=-10000
509

```

510 %

- \@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`.

```

511 \newcount\@lockR
512 \newcount\sub@lockR
513
514 \newcommand*{\do@lockonR}{%
515   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
516   \ifsublines@
517     \xright@appenditem{-1005}\to\actions@listR
518     \ifnum\sub@lockR=\z@
519       \sub@lockR \@ne
520     \else
521       \ifnum\sub@lockR=\thr@@
522         \sub@lockR \@ne
523       \fi
524     \fi
525   \else
526     \xright@appenditem{-1003}\to\actions@listR
527     \ifnum\@lockR=\z@
528       \@lockR \@ne
529     \else
530       \ifnum\@lockR=\thr@@
531         \@lockR \@ne
532       \fi
533     \fi

```

```

534 \fi}
535
536 %

```

`\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`.

```

\skip@lockoff
537
538
539 \newcommand{\do@lockoffR}{%
540 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
541 \ifsublines@
542 \xright@appenditem{-1006}\to\actions@listR
543 \ifnum\sub@lockR=\tw@
544 \sub@lockR \thr@@
545 \else
546 \sub@lockR \z@
547 \fi
548 \else
549 \xright@appenditem{-1004}\to\actions@listR
550 \ifnum\@lockR=\tw@
551 \@lockR \thr@@
552 \else
553 \@lockR \z@
554 \fi
555 \fi}
556
557
558 %

```

`\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`.

```

559 \newcount\insert@countR
560 %

```

- #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.

```

561 \newcommand*{\@ref@regR}[2]{%
562   \global\advance\@edtext@level by 1%
563   \global\insert@countR=#1\relax
564   \loop\ifnum\insert@countR>\z@
565     \xright@appenditem{\the\absline@numR}\to\insertlines@listR
566     \global\advance\insert@countR \m@ne
567   \repeat
568 %

```

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.

```

569 \begingroup
570   \let\@ref=\dummy@ref
571   \let\@lopR\@gobble
572   \let\page@action=\relax
573   \let\sub@action=\relax
574   \let\set@line@action=\relax
575   \let\@lab=\relax
576   \let\@lemma=\relax
577   \let\@sw\@gobblethree%
578   #2
579   \global\endpage@num=\page@numR
580   \global\endline@num=\line@numR
581   \global\endsubline@num=\subline@numR
582 \endgroup
583 %

```

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

```

584 \xright@appenditem%
585   {\the\page@numR|\the\line@numR|}%
586   \ifsublines@ \the\subline@numR \else 0\fi|}%
587   \the\endpage@num|\the\endline@num|}%
588   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@listR
589 %

```

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

```

590   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
@edtext@level\endcsname}%
591 %

```

Declare and init boolean for lemma in this level.

```

592   \providebool{lemmacommand@\the\@edtext@level}%

```



```

593 \boolfalse{lemmacommand@\the\@edtext@level}%
594 %

```

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

```

595 #2
596 % Now, we store the list of \protect\cs{@sw} of this current \protect\cs{
edtext} as an element of
597 % the global list of list of \protect\cs{@sw} for a \protect\cs{edtext}
depth.
598 % \begin{macrocode}
599 \ifnum\@edtext@level>0%
600 \def\create@this@edtext@level{\expandafter\list@create\expandafter{
csname sw@list@edtextR@\the\@edtext@level\endcsname}}%
601 \ifcsundef{sw@list@edtextR@\the\@edtext@level}{\
create@this@edtext@level}{}%
602 \letcs{\@tmp}{sw@list@edtextR@\the\@edtext@level}%
603 \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}%
604 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
605 \global\cslet{sw@list@edtextR@\the\@edtext@level}{\@tmp}%
606 \fi%
607 %

```

Decrease edtext level counter.

```

608 \global\advance\@edtext@level by -1%
609 }
610 %

```

\@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`.

```

611 \renewcommand*{\@pend}[1]{%
612 \ifbypstart@\global\line@num=0\fi%
613 \xright@appenditem{#1}\to\linesinpar@listL}
614 \renewcommand*{\@pendR}[1]{%
615 \ifbypstartR@\global\line@numR=0\fi
616 \xright@appenditem{#1}\to\linesinpar@listR}
617
618 %

```

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

```

619 \newcommand{\@pstart}[3]{%
620 \ifcsdef{minpage@pstart@#1}%
621 {\ifnumgreater{#2}{\csuse{minpage@pstart@#1}}}%
622 {\csnumgdef{minpage@pstart@#1}{#2}}%

```

```

623     }%
624   }%
625   {\csnumgdef{minpage@pstart@#1}{#2}}
626   \csnumgdef{afterlines@pstart@#1L}{#3}%
627 }%
628
629 \newcommand{\@pstartR}[3]{%
630   \numdef{\@tmp}{#2-1}%Because we have not to know in which page the pstart
631   starts, but in which pair of facing page
632   \ifcsdef{minpage@pstart@#1}%
633     {\ifnumgreater{\@tmp}{\csuse{minpage@pstart@#1}}}%
634     {\csnumgdef{minpage@pstart@#1}{\@tmp}}%
635   }%
636   {\csnumgdef{minpage@pstart@#1}{\@tmp}}
637   \csnumgdef{afterlines@pstart@#1R}{#3}%
638 }%
639 %

```

`\@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`.

```

640 \renewcommand*{\@lopL}[1]{%
641   \xright@appenditem{#1}\to\linesonpage@listL}
642 \renewcommand*{\@lopR}[1]{%
643   \xright@appenditem{#1}\to\linesonpage@listR}
644
645 %

```

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`.

```

646 \newwrite\linenum@outR
647 %

```

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

```

\first@linenum@out@Rtrue
\first@linenum@out@Rfalse
648 \newif\iffirst@linenum@out@R
649 \first@linenum@out@Rtrue
650 %

```

\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.

```

651 \newcommand*{\line@list@stuffR}[1]{%
652   \read@linelist{#1}%
653   \iffirst@linenum@outR
654     \immediate\closeout\linenum@outR
655     \global\first@linenum@outRfalse
656     \immediate\openout\linenum@outR=#1
657     \immediate\write\linenum@outR{\string\line@list@version{\
this@line@list@version}}%
658     \ifl@dpaging%
659     \immediate\write\linenum@outR{\string\@par@sync@option{\
@par@this@sync@option}}%
660     \fi%
661   \else
662     \if@minipage%
663     \leavevmode%
664     \fi%
665     \closeout\linenum@outR%
666     \openout\linenum@outR=#1%
667   \fi}
668
669 %

```

\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.

```

670 \newcommand*{\new@lineL}{%
671   \write\linenum@out{\string\@nl[\the\c@page][\thepage]}}
672 %

```

\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.

```

673 \newcommand*{\new@lineR}{%
674   \write\linenum@outR{\string\@nl[\the\c@page][\thepage]}}
675 %

```

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

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

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

- `\setline` You can use `\setline{⟨num⟩}` in running text (i.e., within `\pstart... \pend`) to set the current visible line-number to a specified positive value. It is defined in `reledmac`.
- `\setlinenum` You can use `\setlinenum{⟨num⟩}` before a `\pstart` to set the visible line-number to a specified positive value. It writes a `\l@d@set` command to the line-list file. It is defined in `reledmac`.
- `\startlock` 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@par  \newcommand{\newseries@par}[1]{%
677 %
```

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 boolean flags, and to set them to true when a note series is not printed on one side. We check the `nofamiliar` and `nocritical` `Eledmac` options.

```
678 \unless\ifnofamiliar@%
679 \csgdef{onlysideX@#1}{}%
680 \global\newbool{keepforsideX@#1}%
681 \fi%
```

```

682 \unless\ifnocritical@%
683 \global\newbool{keepforXside@#1}%
684 \csgdef{Xonlyside@#1}{}%
685 \fi%
686 %

```

V.2 Tools specific to familiar footnotes

```

687 \unless\ifnofamiliar@%
688 %

```

V.2.1 Managing correct number

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

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

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

So here, we only defined the new counter.

```

689 \newcounter{footnote#1@typeset}%
690 %

```

V.2.2 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.

```

691 \expandafter\list@create\csname footnote#1@mk\endcsname%
692 %

```

Then, declare the `\footnoteXnomk` command.

```

693 \expandafter\newcommand\csname footnote#1nomk\endcsname[1]{%
694 %

```

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

```

695 \toggletrue{nomk}%
696 \csuse{footnote#1}{##1}%
697 \togglefalse{nomk}%
698 %

```

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`.

```

699 \letcs{\@tmp}{footnote#1@mk}%
700 \numdef\@tmpa{\csuse{c@footnote#1}}%
701 \global\xright@appenditem{\@tmpa}\to\@tmp%
702 \global\cslet{footnote#1@mk}{\@tmp}%
703 }%
704 %

```

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

```

705 \expandafter\newcommand\csname footnote#1mk\endcsname{%
706 %

```

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

```

707 \letcs{\@tmp}{footnote#1@mk}%
708 \gl@p\@tmp\to\@tmpa%
709 \global\cslet{footnote#1@mk}{\@tmp}%
710 %

```

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

```

711 \letcs{\old@footnote}{c@footnote#1}%
712 \setcounter{footnote#1}{\@tmpa}%
713 %

```

Define the footnote mark and print it

```

714 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
715 \csuse{@footnotemark#1}%
716 %

```

Restore previous footnote counter and finally add space.

```

717 \setcounter{footnote#1}{\old@footnote}%
718 \xspace%
719 }%
720 %

```

End of tools specific to familiar notes.

```

721 \fi
722 %
    End of \newseries@par.
723 }%
724 %

```

V.2.3 Get correct footnote number

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

```

725 \newcommand{\save@familiarfootnote@number}{%
726 \unless\ifnofamiliar%
727 \def\do##1{\csxdef{saved@footnote##1}{\the\csname c@footnote##1\
endcsname}}%
728 \dolistloop{\@series}%
729 \fi%
730 \xdef\saved@footnote{\the\c@footnote}%
731 }
732 \newcommand{\get@familiarfootnote@number}{%
733 \unless\ifnofamiliar%
734 \def\do##1{\setcounter{footnote##1@typeset}{\csuse{saved@footnote##1}}}
735 %
736 \dolistloop{\@series}%
737 \fi%
738 \setcounter{footnote@typeset}{\saved@footnote}%
739 }
740 %

```

V.3 Create hooks

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

```

740 \unless\ifnocritical%
741 \newhookcommand@series{Xonlyside}%
742 \fi%
743 \unless\ifnofamiliar%
744 \newhookcommand@series{onlysideX}%
745 \fi
746
747
748 %

```

V.4 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`.

```

749 \newcommand{\init@series@par}{%
750   \def\do##1{\newseries@par{##1}}%
751   \dolistloop{\@series}%
752 }%
753 \init@series@par%
754 %

```

V.5 Tools specific to \LaTeX 's classical footnotes

As users can use classical footnotes of \LaTeX (`\footnote`) in parallel texts, we must integrate the same tools to get correct number as for `reledmac`' footnotes (V.2.1 p. 45).

```

\footnote@reading55 \newcount\footnote@reading%
\footnote@typeset56 \newcounter{footnote@typeset}%
757 %

```

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@pc58 \list@create{\list@pstartL@pc}%
\list@pstartR@pc59 \list@create{\list@pstartR@pc}%
760 %

```

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@pc61 \def\dump@pstartL@pc{%
\dump@pstartR@pc62   \xright@appenditem{\the\c@pstartL}{\to\list@pstartL@pc%
763   \global\cslet{numberpstartL\the\l@dumpstartsL}{\ifnumberpstart}%
764 }%
765
766 \def\dump@pstartR@pc{%

```



```

767 \xright@appenditem{\the\c@pstartR}\to\list@pstartR@pc%
768 \global\cslet{numberpstartR\the\l@dnumpstartsR}{\ifnumberpstart}%
769 }%
770
771 %

```

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

`\restore@pstartR@pc`

```

772 \def\restore@pstartL@pc{%
773   \ifx\list@pstartL@pc\empty\else%
774     \gl@p\list@pstartL@pc\to\@temp%
775     \global\c@pstartL=\@temp%
776   \fi%
777 }%
778 \def\restore@pstartR@pc{%
779   \ifx\list@pstartR@pc\empty\else%
780     \gl@p\list@pstartR@pc\to\@temp%
781     \global\c@pstartR=\@temp%
782   \fi%
783 }%
784 %

```

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.

```

785 \newenvironment{pairs}{%}
786   \l@dpairingtrue
787   \l@dpagingfalse
788   \initnumbering@quote
789   \save@familiarfootnote@number%
790   \if@ledgroup%
791     \get@familiarfootnote@number%
792   \fi%
793   \save@section@number%
794   \at@begin@pairs%
795 }{%
796   \l@dpairingfalse
797 }
798
799 %

```

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

```

800 \newcommand{\AtBeginPairs}[1]{\xdef\at@begin@pairs{#1}}%
801 \def\at@begin@pairs{}%
802
803 %

```

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.

```

804 \newenvironment{pages}{%
805   \l@dpairingtrue
806   \l@dpagingtrue
807   \initnumbering@quote
808   \save@familiarfootnote@number%
809   \if@ledgroup%
810     \get@familiarfootnote@number%
811   \fi%
812   \save@section@number%
813   \setlength{\Lcolwidth}{\textwidth}%
814   \setlength{\Rcolwidth}{\textwidth}%
815 }{%
816   \l@dpairingfalse
817   \l@dpagingfalse
818 }
819
820 %

```

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

```

821 \newif\ifinstanzaL
822 \newif\ifinstanzaR
823 %

```

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.

```

824 \newenvironment{Leftside}{%
825   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname\else%
826     \led@err@Leftside@PreviousNotPrinted%
827   \fi%
828   \ledRcolfalse
829   \setcounter{pstartL}{1}
830   \let\pstart\pstartL
831   \let\thepstart\thepstartL
832   \let\pend\pendL
833   \let\memorydump\memorydumpL
834   \Leftsidehook

```

```

835 \let\old@startstanza\@startstanza
836 \def\@startstanza[##1]{\global\instanzaLtrue\old@startstanza[##1]}
837 }{
838 \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname%
839 \led@error@missing@numbering{Leftside}%
840 \fi%
841 \Leftsidehookend}
842 %

```

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

```

\Leftsidehookend
\Rightsidehook
\Rightsidehookend
843 \newcommand*{\Leftsidehook}{}
844 \newcommand*{\Leftsidehookend}{}
845 \newcommand*{\Rightsidehook}{}
846 \newcommand*{\Rightsidehookend}{}
847
848 %

```

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.

```

849 \newenvironment{Rightside}{%
850 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else%
851 \led@err@Rightside@PreviousNotPrinted%
852 \fi%
853 \ledRcoltrue
854 \let\beginnumbering\beginnumberingR
855 \let\endnumbering\endnumberingR
856 \let\pausenumbering\pausenumberingR
857 \let\resumenumbering\resumenumberingR
858 \let\memorydump\memorydumpR
859 \let\thepstart\thepstartR
860 \let\pstart\pstartR
861 \let\pend\pendR
862 \let\ledpb\ledpbR
863 \let\lednopb\lednopbR
864 \let\lineation\lineationR
865 \Rightsidehook
866 \let\old@startstanza\@startstanza
867 \def\@startstanza[##1]{\global\instanzaRtrue\old@startstanza[##1]}
868 }{%
869 \ledRcolfalse
870 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
871 \led@error@missing@numbering{Rightside}%
872 \fi%
873 \Rightsidehookend
874 }
875

```

876 %

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.

```
877 \newcount\num@linesR
878 \newbox\one@lineR
879 \newcount\par@lineR
880 %
```

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

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

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

```
881
882 \newcounter{pstartL}
883 \renewcommand{\thepstartL}{\bfseries\@arabic\c@pstartL}. }
884 \newcounter{pstartR}
885 \renewcommand{\thepstartR}{\bfseries\@arabic\c@pstartR}. }
886
887 \newcommandx*\pstartL}[1][1]{%
888   \if@nobreak%
889     \let\@oldnobreak\@nobreaktrue%
890   \else%
891     \let\@oldnobreak\@nobreakfalse%
```

```

892 \fi%
893 \@nbreaktrue%
894 \ifluatex%
895 \xdef\l@luatextextdir@L{\the\textdir}%
896 \xdef\l@luatexpardir@L{\the\pardir}%
897 \xdef\l@luatexbodydir@L{\the\bodydir}%
898 \fi%
899 \ifnumbering \else%
900 \led@err@PstartNotNumbered%
901 \beginnumbering%
902 \fi%
903 \ifnumberedpar@%
904 \led@err@PstartInPstart%
905 \pend%
906 \fi%
907 %

```

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

```

908 \ifpst@rtedL\else%
909 \list@clear{\inserts@list}%
910 \global\let\next@insert=\empty%
911 \global\pst@rtedLtrue%
912 \fi%
913 \begingroup\everypar{}%
914 %

```

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.

```

915 \global\advance\l@dnumpstartsL \@ne%
916 \ifnum\l@dnumpstartsL>\l@dc@maxchunks%
917 \led@err@TooManyPstarts%
918 \global\l@dnumpstartsL=\l@dc@maxchunks%
919 \fi%
920 \global\setnamebox{\l@dLcolrawbox\the\l@dnumpstartsL}=\vbox\bgroup%
921 %

```

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.

```

922 \l@dzeropenalties%
923 \ifautopar\else%
924 \ifnumberpstart%
925 \ifsidepstartnum%
926 \else%
927 \thepstartL%
928 \fi%
929 \fi%
930 \fi%

```

```

931 \hsize=\Lcolwidth%
932 \numberedpar@true%
933 \iflabelpstart\protected@edef\@currentlabel%
934     {\p@pstartL\thepstartL}\fi%
935 %

```

Dump the optional arguments

```

936 \ifstrempy{#1}%
937     {\csgdef{before@pstartL@the\l@dumpstartsL}{\at@every@pstart}}%
938     {\csgdef{before@pstartL@the\l@dumpstartsL}{\noindent#1}}%
939     \at@every@pstart@call%
940 %

```

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

```

941 \ignorespaces%
942 %

```

```

943 }
944 %

```

The same for right side.

```

945 \newcommandx*{\pstartR}[1][1]{%
946     \if@nobreak%
947         \let\@oldnobreak\@nobreaktrue%
948     \else%
949         \let\@oldnobreak\@nobreakfalse%
950     \fi%
951     \@nobreaktrue%
952     \ifluatex%
953         \xdef\l@luatextextdir@R{\the\textdir}%
954         \xdef\l@luatexpardir@R{\the\pardir}%
955         \xdef\l@luatexbodydir@R{\the\bodydir}%
956     \fi%
957     \ifnumberingR \else%
958         \led@err@PstartNotNumbered%
959         \beginnumberingR%
960     \fi%
961     \ifnumberedpar@%
962         \led@err@PstartInPstart%
963         \pendR%
964     \fi%
965     \ifpst@rtedR\else%
966         \list@clear{\inserts@listR}%
967         \global\let\next@insertR=\empty%
968         \global\pst@rtedRtrue%
969     \fi%
970     \begingroup\everypar{}%
971     \global\advance\l@dumpstartsR \@one%
972     \ifnum\l@dumpstartsR>\l@dc@maxchunks%

```

```

973 \led@err@TooManyPstarts%
974 \global\l@dnumpstartsR=\l@dc@maxchunks%
975 \fi%
976 \global\setnamebox{l@dRcolrawbox\the\l@dnumpstartsR}=\vbox\bgroup%
977 \l@dzeropenalties%
978 \ifautopar\else%
979 \ifnumberpstart%
980 \ifsidepstartnum\else%
981 \thepstartR%
982 \fi%
983 \fi%
984 \fi%
985 \hsize=\Rcolwidth%
986 \numberedpar@true%
987 \iflabelpstart\protected@edef\@currentlabel%
988 {\p@pstartR\thepstartR}\fi%
989 \ifstrempy{#1}%
990 {\csgdef{before@pstartR@the\l@dnumpstartsR}{\at@every@pstart}}%
991 {\csgdef{before@pstartR@the\l@dnumpstartsR}{\noindent#1}}%
992 \at@every@pstart@call%
993 \ignorespaces%
994 }
995 %

```

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

```

996 \newcommandx*{\pendL}[1][1]{%
997 \ifnumbering \else%
998 \led@err@PendNotNumbered%
999 \fi%
1000 \ifnumberedpar@ \else%
1001 \led@err@PendNoPstart%
1002 \fi%
1003 %

```

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.

```

1004 \endgraf\global\num@lines=\prevgraf\egroup%
1005 \global\par@line=0%
1006 %

```

End the group that was begun in the \pstart.

```

1007 \endgroup%
1008 \ignorespaces%
1009 \@oldnobreak%
1010 \dump@pstartL@pc%
1011 \ifnumberpstart%
1012 \addtocounter{pstartL}{1}%
1013 \fi

```

```

1014 \parledgroup@beforenotes@save{L}%
1015 %
Dump content of the optional argument.
1016 \ifstrempy{#1}%
1017   {\csgdef{after@pendL@the\l@dnumstartsL}{\at@every@pend}}%
1018   {\csgdef{after@pendL@the\l@dnumstartsL}{\noindent#1}}%
1019 }
1020 %

```

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

```

1021 \newcommandx*{\pendR}[1][1]{%
1022   \ifnumberingR \else%
1023     \led@err@PendNotNumbered%
1024   \fi%
1025   \ifnumberedpar@ \else%
1026     \led@err@PendNoPstart%
1027   \fi%
1028   \endgraf\global\num@linesR=\prevgraf\egroup%
1029   \global\par@lineR=0%
1030   \endgroup%
1031   \ignorespaces%
1032   \@oldnobreak%
1033   \dump@pstartR@pc%
1034   \ifnumberpstart%
1035     \addtocounter{pstartR}{1}%
1036   \fi%
1037   \parledgroup@beforenotes@save{R}%
1038   \ifstrempy{#1}%
1039     {\csgdef{after@pendR@the\l@dnumstartsR}{\at@every@pend}}%
1040     {\csgdef{after@pendR@the\l@dnumstartsR}{\noindent#1}}%
1041 }
1042 %
1043 %

```

\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.

```

1044 \newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
1045 \gdef\at@every@pstart@call{}%
1046 %

```

\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

```

1047 \newif\ifprint@last@after@pendL%
1048 \newif\ifprint@last@after@pendR%
1049 %

```


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
`\l@drightbox` text.

```
1050 \newbox\l@dleftbox
1051 \newbox\l@drightbox
1052
1053 %
```

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

```
\countRline
1054 \newcount\countLline
1055 \countLline \z@
1056 \newcount\countRline
1057 \countRline \z@
1058
1059 %
```

`\@donereallinesL` We need to know the number of ‘real’ lines output (i.e., those that have been input by
`\@donetotallinesL` the user), and the total lines output (which includes any blank lines output for synchro-
`\@donereallinesR` nisation).
`\@donetotallinesR`

```
1060 \newcount\@donereallinesL
1061 \newcount\@donetotallinesL
1062 \newcount\@donereallinesR
1063 \newcount\@donetotallinesR
1064
1065 %
```

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

```
1066 \newcommand*\do@lineL{%
1067 \letcs{\ifnumberpstart}{numberpstart@L\the\l@dpscL}%
1068 \advance\countLline \@ne%
1069 \ifvbox\namebox{\l@dLcolrawbox\the\l@dpscL}%
1070 {\vbadness=10000%
1071 \splittopskip=\z@%
1072 \do@lineLhook%
1073 \l@demptyd@ta%
1074 \global\setbox\one@line=\vsplit\namebox{\l@dLcolrawbox\the\l@dpscL}%
1075 to\baselineskip}%
1076 \IfStrEq{\splitfirstmarks\parledgroup@}{\begin}{\
parledgroup@notes@startL}{}%
1077 \unvbox\one@line \global\setbox\one@line=\lastbox%
1078 \@writepageofparL%
```

```

1079 \getline@numL%
1080 \ifnum\@lock>\@ne%
1081   \inserthangingsymboltrue%
1082 \else%
1083   \inserthangingsymbolfalse%
1084 \fi%
1085 \setbox\l@leftbox%
1086 \hb@xt@ \Lcolwidth{%
1087   \ifl@dhidenumber%
1088     \global\l@dhidenumberfalse%
1089     \f@x@l@cks%
1090   \else%
1091     \affixline@num%
1092   \fi%
1093   \xifinlist{\the\l@dpscl}{\eled@sections@@}%
1094     {\add@inserts\affixside@note}%
1095     {\print@lineL}%
1096   }%
1097 \add@penaltiesL%
1098 \global\advance\@donereallinesL\@ne%
1099 \global\advance\@donetotallinesL\@ne%
1100 \else%
1101   \iflinenumberLevenifblank
1102     \new@lineL%
1103     \l@emptyd@ta%
1104     \getline@numL%
1105     \affixline@num%
1106     \setbox\l@leftbox \hb@xt@ \Lcolwidth{%
1107       \l@dld@ta %space kept for backward compatibility
1108       \hspace*{\Lcolwidth}%
1109       \ledrlfill\l@drd@ta%
1110     }%
1111   \else%
1112     \setbox\l@leftbox \hb@xt@ \Lcolwidth{\hspace*{\Lcolwidth}}%
1113   \fi%
1114   \global\advance\@donetotallinesL\@ne%
1115 \fi%
1116 }%
1117
1118
1119 %

```

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

```

1120 \def\print@lineL{%
1121   \affixpstart@numL%
1122   \l@dld@ta %space kept for backward compatibility
1123   \add@inserts\affixside@note%
1124   \l@dlsn@te %space kept for backward compatibility

```

```

1125 \hb@xt@ \Lcolwidth{\ledllfill\hb@xt@ \wd\one@line{%
1126 \do@insidelineLhook%
1127 \ifluatex%
1128 \texdir\l@luatextextdir@L%
1129 \fi%
1130 \new@lineL%
1131 \inserthangingsymbolL%
1132 \l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta%
1133 \l@drsn@te}}
1134
1135 %

```

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

```

1136 \def\print@eledsectionL{%%
1137 \addtocounter{pstartL}{-1}%
1138 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{%
1139 \ifdefstring{\@eledsectmark}{L}{\ledsectnomark}%
1140 \numdef{\temp@}{\l@dpscl-1}%
1141 \xifinlist{\temp@}{\eled@sections@}{\@nobreaktrue}{\@nobreakfalse}%
1142 \@eled@sectioningtrue%
1143 \bgroup%
1144 \ifluatex%
1145 \texdir\l@luatextextdir@L%
1146 \pardir\l@luatexpardir@L%
1147 \bodydir\l@luatexbodydir@L%
1148 \ifdefstring{\l@luatextextdir@L}{TRT}{\@RTLtrue}{}%
1149 \fi%
1150 \csuse{eled@sectioning@the\l@dpscl}%
1151 \egroup%
1152 \@eled@sectioningfalse%
1153 \global\csundef{eled@sectioning@the\l@dpscl}%
1154 \if@RTL%
1155 \hspace{-3\paperwidth}%
1156 {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1157 \else%
1158 \hspace{3\paperwidth}%
1159 {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1160 \fi%
1161 \vskip\eledsection@correcting@skip%
1162 }
1163
1164 %

```

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

```

\doinssidelineLhook 1165 \newcommand*{\dolineLhook}[1]{\gdef\do@lineLhook{#1}}%
\doinssidelineRhook 1166 \newcommand*{\dolineRhook}[1]{\gdef\do@lineRhook{#1}}%
1167 \newcommand*{\doinssidelineLhook}[1]{\gdef\do@insidelineLhook{#1}}%

```

```

1168 \newcommand*{\doinsidelineRhook}[1]{\gdef\do@insidelineRhook{#1}}%
1169
1170 %

```

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

```

\do@lineRhook
\do@insidelineLhook 1171 \newcommand*{\do@lineLhook}{%
1172 \newcommand*{\do@lineRhook}{%
\do@insidelineRhook 1173 \newcommand*{\do@insidelineLhook}{%
1174 \newcommand*{\do@insidelineRhook}{%
1175
1176 %

```

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

```

1177 \newcommand*{\do@lineR}{%
1178   \let\linenumrepL\linenumrep%
1179   \let\sublinenumrepL\sublinenumrep%
1180   \let\linenumrep\linenumrepR%
1181   \let\sublinenumrep\sublinenumrepR%
1182   \letcs{\ifnumberpstart}{numberpstart@R\the\l@dpscR}%
1183   \ledRcol@true%
1184   \advance\countRline \@ne%
1185   \ifvbox\namebox{\l@dRcolrawbox\the\l@dpscR}%
1186     {\vbadness=10000%
1187      \splittopskip=\z@%
1188      \do@lineRhook%
1189      \l@demtyd@ta%
1190      \global\setbox\one@lineR=\vsplit\namebox{\l@dRcolrawbox\the\l@dpscR}%
1191        to\baselineskip}%
1192     \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startR}{}%
1193     \unvbox\one@lineR \global\setbox\one@lineR=\lastbox%
1194     \@writepageofparR%
1195     \getline@numR%
1196     \ifnum\@lockR>\@ne%
1197       \inserthangingsymbolRtrue%
1198     \else%
1199       \inserthangingsymbolRfalse%
1200     \fi%
1201     \setbox\l@drightrightbox%
1202     \hb@xt@ \Rcolwidth{%
1203       \ifl@dhidenumber%
1204         \global\l@dhidenumberfalse%
1205         \f@x@l@cksR%
1206       \else%
1207         \affixline@numR%
1208       \fi%
1209       \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%

```

```

1210     {\add@insertsR\affixside@noteR}%
1211     {\print@lineR}%
1212   }%
1213   \add@penaltiesR%
1214   \global\advance\@donereallinesR\@ne%
1215   \global\advance\@donetotallinesR\@ne%
1216   \else%
1217     \iflinenumberRevenifblank%
1218       \new@lineR
1219       \l@emptyd@ta%
1220       \getline@numR%
1221       \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1222         \affixline@numR%
1223         \l@dld@ta %space kept for backward compatibility
1224         \hspace*{\Rcolwidth}%
1225         \ledrlfill\l@drd@ta%
1226       }%
1227     \else%
1228       \setbox\l@drightbox \hb@xt@ \Rcolwidth{\hspace*{\Rcolwidth}}%
1229     \fi%
1230     \global\advance\@donetotallinesR\@ne%
1231   \fi%
1232   \ledRcol@false%
1233   \let\linenumrep\linenumrepL%
1234   \let\sublinenumrep\sublinenumrepL%
1235 }
1236
1237
1238 %

```

`\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.

```

1239 \newcommand*{\getline@numR}{%
1240   \global\advance\absline@numR \@ne
1241   \do@actionsR
1242   \do@ballastR
1243   \ifledgroupnotesR\else
1244     \ifnumberline
1245       \ifsublines@
1246         \ifnum\sub@lockR<\tw@
1247           \global\advance\subline@numR \@ne
1248         \fi
1249       \else
1250         \ifnum\@lockR<\tw@
1251           \global\advance\line@numR \@ne

```

```

1252         \global\subline@numR \z@
1253         \fi
1254     \fi
1255 \fi
1256 }
1257 \newcommand*{\getline@numL}{%
1258     \global\advance\absline@num \@ne
1259     \do@actions
1260     \do@ballast
1261     \ifledgroupnotesL@\else
1262     \ifnumberline
1263         \ifsublines@
1264             \ifnum\sub@lock<\tw@
1265                 \global\advance\subline@num \@ne
1266             \fi
1267         \else
1268             \ifnum\@lock<\tw@
1269                 \global\advance\line@num \@ne
1270                 \global\subline@num \z@
1271             \fi
1272         \fi
1273     \fi
1274 \fi
1275 }
1276 }
1277
1278 %
1279 %

```

`\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.

```

1280 \newcommand*{\do@ballastR}{\global\ballast@count=\z@
1281     \begingroup
1282     \advance\absline@numR \@ne
1283     \ifnum\next@actionlineR=\absline@numR
1284         \ifnum\next@actionR>-1001
1285             \global\advance\ballast@count by -\c@ballast
1286         \fi
1287     \fi
1288     \endgroup}
1289 %

```

`\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`

```

1290
1291 \newif\ifl@dskipversenumberR

```

```

1292 \newcommand*{\do@actions@fixedcodeR}{%
1293   \ifcase\@l@dtmpcnta%
1294     \or%                               % 1001
1295       \global\sublines@true
1296     \or%                               % 1002
1297       \global\sublines@false
1298     \or%                               % 1003
1299       \global\@lockR=\@ne
1300     \or%                               % 1004%
1301       \ifnum\@lockR=\tw@
1302         \global\@lockR=\thr@@
1303       \else
1304         \global\@lockR=\z@
1305       \fi
1306     \or%                               % 1005
1307       \global\sub@lockR=\@ne
1308     \or%                               % 1006
1309       \ifnum\sub@lockR=\tw@
1310         \global\sub@lockR=\thr@@
1311       \else
1312         \global\sub@lockR=\z@
1313       \fi
1314     \or%                               % 1007
1315       \l@dskipnumbertrue
1316     \or%                               % 1008
1317       \l@dskipversenumberRtrue%
1318     \or%                               % 1009
1319       \l@dhidenumbertrue%
1320     \else%
1321       \led@warn@BadAction
1322     \fi%
1323 }
1324
1325
1326 \newcommand*{\do@actionsR}{%
1327   \global\let\do@actions@nextR=\relax
1328   \@l@dtmpcntb=\absline@numR
1329   \ifnum\@l@dtmpcntb<\next@actionlineR\else
1330     \ifnum\next@actionR>-1001\relax
1331       \global\page@numR=\next@actionR
1332       \ifbypage@R
1333         \global\line@numR \z@ \global\subline@numR \z@
1334       \fi
1335     \else
1336       \ifnum\next@actionR<-4999\relax % 9/05 added relax here
1337         \@l@dtmpcnta=-\next@actionR
1338         \advance\@l@dtmpcnta by -5001\relax
1339         \ifsublines@
1340           \global\subline@numR=\@l@dtmpcnta
1341         \else

```

```

1342     \global\line@numR=\@l@tempcnta
1343     \fi
1344     \else
1345       \@l@tempcnta=-\next@actionR
1346       \advance\@l@tempcnta by -1000\relax
1347       \do@actions@fixedcodeR
1348     \fi
1349   \fi
1350   \ifx\actionlines@listR\empty
1351     \gdef\next@actionlineR{1000000}%
1352   \else
1353     \glp\actionlines@listR\to\next@actionlineR
1354     \glp\actions@listR\to\next@actionR
1355     \global\let\do@actions@nextR=\do@actionsR
1356   \fi
1357 \fi
1358 \do@actions@nextR}
1359
1360 %

```

VIII.4 Line number printing

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

```

1361 \ch@cksub@l@ckR
1362 \ch@ck@l@ckR
1363 \f@x@l@cksR
1364 \affixline@numR
1365
1366 \newcommand*{\l@dcalcnun}[3]{%
1367   \ifnum #1 > #2\relax
1368     \@l@tempcnta = #1\relax
1369     \advance\@l@tempcnta by -#2\relax
1370     \divide\@l@tempcnta by #3\relax
1371     \multiply\@l@tempcnta by #3\relax
1372     \advance\@l@tempcnta by #2\relax
1373   \else
1374     \@l@tempcnta=#2\relax
1375   \fi}
1376
1377 \newcommand*{\ch@cksub@l@ckR}{%
1378   \ifcase\sub@lockR
1379   \or
1380     \ifnum\sublock@disp=\@ne
1381       \@l@tempcntb \z@ \@l@tempcnta \@ne
1382     \fi
1383   \or
1384     \ifnum\sublock@disp=\tw@
1385     \else
1386       \@l@tempcntb \z@ \@l@tempcnta \@ne
1387     \fi
1388   \or
1389     \ifnum\sublock@disp=\z@

```



```

1386 \l@dttempcntb \z@ \l@dttempcnta \@ne
1387 \fi
1388 \fi}
1389
1390 \newcommand*{\ch@ck@l@ckR}{%
1391 \ifcase\@lockR
1392 \or
1393 \ifnum\lock@disp=\@ne
1394 \l@dttempcntb \z@ \l@dttempcnta \@ne
1395 \fi
1396 \or
1397 \ifnum\lock@disp=\tw@
1398 \else
1399 \l@dttempcntb \z@ \l@dttempcnta \@ne
1400 \fi
1401 \or
1402 \ifnum\lock@disp=\z@
1403 \l@dttempcntb \z@ \l@dttempcnta \@ne
1404 \fi
1405 \fi}
1406
1407 \newcommand*{\f@x@l@cksR}{%
1408 \ifcase\@lockR
1409 \or
1410 \global\@lockR \tw@
1411 \or \or
1412 \global\@lockR \z@
1413 \fi
1414 \ifcase\sub@lockR
1415 \or
1416 \global\sub@lockR \tw@
1417 \or \or
1418 \global\sub@lockR \z@
1419 \fi}
1420
1421
1422 \newcommand*{\affixline@numR}{%
1423 \ifledgroupnotesR\else\ifnumberline
1424 \ifl@dskipnumber
1425 \global\l@dskipnumberfalse
1426 \else
1427 \ifsublines@
1428 \l@dttempcntb=\subline@numR
1429 \l@dcalcnum{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR}
1430 }%
1431 \ch@cksub@lockR
1432 \else
1433 \l@dttempcntb=\line@numR
1434 \ifx\linenumberlist\empty
1435 \l@dcalcnum{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%

```

```

1435 \else
1436 \l@l@tempcnta=\line@numR
1437 \edef\rem@inder{\linenumberlist,\number\line@numR,}%
1438 \edef\sc@n@list{\def\noexpand\sc@n@list
1439 ###1,\number\l@l@tempcnta,###2|{\def\noexpand\rem@inder{###2}}}%
1440 \sc@n@list\expandafter\sc@n@list\rem@inder|
1441 \ifx\rem@inder\empty\advance\l@l@tempcnta\@ne\fi
1442 \fi
1443 \ch@ck@l@ckR
1444 \fi
1445 \ifnum\l@l@tempcnta=\l@l@tempcntb
1446 \ifl@dskipversenumberR\else
1447 \if@twocolumn
1448 \if@firstcolumn
1449 \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1450 \else
1451 \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1452 \fi
1453 \else
1454 \l@l@tempcntb=\line@marginR
1455 \ifnum\l@l@tempcntb>\@ne
1456 \advance\l@l@tempcntb by\page@numR
1457 \fi
1458 \ifodd\l@l@tempcntb
1459 \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1460 \else
1461 \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1462 \fi
1463 \fi
1464 \fi
1465 \fi
1466 \f@x@l@cksR
1467 \fi
1468 \fi
1469 \fi}
1470 %

```

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@numL71
\affixpstart@numR72 \newcommand*{\affixpstart@numL}{%
\leftpstartnumR
\rightpstartnumR
\leftpstartnumL
\rightpstartnumL
\ifpstartnumR

```

```

1473 \ifsidepstartnum
1474 \if@twocolumn
1475   \if@firstcolumn
1476     \gdef\l@dld@ta{\llap{\leftpstartnumL}}}%
1477   \else
1478     \gdef\l@drd@ta{\rlap{\rightpstartnumL}}}%
1479   \fi
1480 \else
1481   \l@dttempcntb=\line@margin
1482   \ifnum\l@dttempcntb>\@ne
1483     \advance\l@dttempcntb \page@num
1484   \fi
1485   \ifodd\l@dttempcntb
1486     \gdef\l@drd@ta{\rlap{\rightpstartnumL}}}%
1487   \else
1488     \gdef\l@dld@ta{\llap{\leftpstartnumL}}}%
1489   \fi
1490 \fi
1491 \fi
1492 }
1493 \newcommand*{\affixpstart@numR}{%
1494 \ifsidepstartnum
1495 \if@twocolumn
1496   \if@firstcolumn
1497     \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
1498   \else
1499     \gdef\l@drd@ta{\rlap{\rightpstartnumR}}}%
1500   \fi
1501 \else
1502   \l@dttempcntb=\line@marginR
1503   \ifnum\l@dttempcntb>\@ne
1504     \advance\l@dttempcntb \page@numR
1505   \fi
1506   \ifodd\l@dttempcntb
1507     \gdef\l@drd@ta{\rlap{\rightpstartnumR}}}%
1508   \else
1509     \gdef\l@dld@ta{\llap{\leftpstartnumR}}}%
1510   \fi
1511 \fi
1512 \fi
1513 }
1514
1515 \newcommand*{\leftpstartnumL}{
1516 \ifpstartnum
1517 \thepstartL
1518 \kern\linenumsep\global\pstartnumfalse\fi
1519 }
1520 \newcommand*{\rightpstartnumL}{
1521 \ifpstartnum\kern\linenumsep
1522 \thepstartL

```

```

1523 \global\pstartnumfalse\fi
1524 }
1525 \newif\ifpstartnumR
1526 \pstartnumRtrue
1527 \newcommand*{\leftpstartnumR}{
1528 \ifpstartnumR
1529 \thepstartR
1530 \kern\linenumsep\global\pstartnumRfalse\fi
1531 }
1532 \newcommand*{\rightpstartnumR}{
1533 \ifpstartnumR\kern\linenumsep
1534 \thepstartR
1535 \global\pstartnumRfalse\fi
1536 }
1537 %

```

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.

```

1538 \list@create{\inserts@listR}
1539 %

```

`\add@insertsR` The right text version.

`\add@inserts@nextR`

```

1540 \newcommand*{\add@insertsR}{%
1541 \global\let\add@inserts@nextR=\relax
1542 \ifx\inserts@listR\empty \else
1543 \ifx\next@insertR\empty
1544 \ifx\insertlines@listR\empty
1545 \global\noteschanged@true
1546 \gdef\next@insertR{100000}%
1547 \else
1548 \gl@p\insertlines@listR\to\next@insertR
1549 \fi
1550 \fi
1551 \ifnum\next@insertR=\absline@numR
1552 \gl@p\inserts@listR\to\@insertR
1553 \@insertR
1554 \global\let\@insertR=\undefined
1555 \global\let\next@insertR=\empty
1556 \global\let\add@inserts@nextR=\add@insertsR
1557 \fi
1558 \fi
1559 \add@inserts@nextR}
1560
1561 %

```

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 not really relevant. Peter Wilson thinks that it is likely with parallel text that there is no real need to add back any penalties; even if there was, they would have to match across the left and right lines. So, Peter Wilson ends up with the following.

```
1562 \newcommand*{\add@penaltiesL}{\}
1563 \newcommand*{\add@penaltiesR}{\}
1564
1565 %
```

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.

```

1566 \newcommand*{\flush@notesR}{%
1567   \@xloop
1568   \ifx\inserts@listR\empty \else
1569     \glp\inserts@listR\to\@insertR
1570     \@insertR
1571     \global\let\@insertR=\undefined
1572   \repeat}
1573
1574 %

```

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

`\correct@Xfootins@box`

`\print@notesX@forpages`

`\correct@footinsX@box`

- If notes are to be printed in both sides, we just proceed the usual way: print the foot starts for the series, then the foot group.
- If notes are to be printed in the left side, we do these prints only for even pages ; if notes are to be printed in the right side, we do these prints only for odd pages.
- However, that is not enough. Because the problem does not only consists in printing notes in any particular page. It is also not to put aside room for notes in the pages where we do not want to print them. To take an example: if some note in the left side is too long by 160pt to be printed in full in the left page, we do not want to put aside 160pt a space for it in the following right page.
- To solve this problem, we change the magnification factor associated with notes before going to the next page. If we start a page where no notes are supposed to be printed, the magnification counter is set to 0. 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 and skip, read *The TeXbook* p. 122-125).
- There still remains a last problem. This problem is quite complex to understand, so an example will speak for itself. Suppose we allow 10 lines of notes by page. Suppose a long note, be it 25 lines, which needs three pages to be printed. Suppose it must be printed only on left pages, namely odd pages.

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

On p. 2, the first 10 lines of the notes are printed. On p. 3, the box associated to the notes contains 10 lines. However, as we are in a right page, we do not void this box. So \TeX will keep its content for the pages to come. However, on p. 4 it will also add one line in the footnote box, because in any case, \TeX adds some content in the box when preparing the output routines, even if there is some content left in this box from the previous pages. So the lines in the note box at p. 4 will be $10 + 1 = 11$. There is one line which should not be there. Furthermore, as the box size is for 10 lines and not for 11 lines, this last line will be glued to the previous one.

To fix this double issue:

- For the pages where notes must be NOT printed, we allow to every note box one line less than it ought to be. In our example, that means that we allow \TeX to add only $10 - 1 = 9$ line in the note box on p. 3. Before shifting to the pages where notes must be printed, we allow to every notes the expected number of lines. In our example, that means that we allow \TeX to add 10 lines in the note box on p. 4. As on p. 3 only 9 lines were allowed, that means note box of p. 4 will contain $9 + 1 = 10$ lines. So the “one line too many” problem is solved.
- Still remains the “glue” problem. We solve it by recreating a clean note box. We split the one which is created by \TeX to get the next line printed. Then, we create the new box, by bringing together the first part and the last part of the split box, adding some skip between them. That is achieved by `\correct@Xfootins@box` (or `\correct@footinsX@box` for familiar notes).

The code to print critical notes, when processing `\Pages`.

```
1575 \newcommand\print@Xnotes@forpages[1]{%
1576 %
```

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

```
1577 \ifcseempty{Xonlyside@#1}{%
1578 \csuse{#1footstart}{#1}%
1579 \csuse{#1footgroup}{#1}%
1580 }%
1581 %
```

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

```
1582 {%
1583 \ifboolexpr{%
1584 ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page
}})%
1585 or%
1586 (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}}))%
1587 }%
1588 %
```

If we are in a page where notes must be printed, print the notes, after having made the corrections which are needed for boxes.

```

1589     {%
1590         \correct@Xfootins@box{#1}%
1591         \csuse{#1footstart}{#1}%
1592         \csuse{#1footgroup}{#1}%
1593     %

```

Then, say not to keep room for notes in the next page.

```

1594         \global\count\csuse{#1footins}=0%
1595         \global\skip\csuse{#1footins}=0pt%
1596     %

```

And also, allow one line less for notes in the next page.

```

1597         \csuse{Xnotefontsize@#1}%
1598         \global\advance\dimen\csuse{#1footins} by -\baselineskip%
1599     %

```

Now we have printed the notes. So we put aside this fact.

```

1600         \global\boolfalse{keepforXside@#1}%
1601     }%
1602 %

```

In case we are on a page where notes must NOT be printed. First, memorize that we have not printed the notes, despite having some to print.

```

1603     {%
1604         \global\booltrue{keepforXside@#1}%
1605     %

```

Then restore expected rooms for notes on the next page.

```

1606         \global\count\csuse{#1footins}=\csuse{default@#1footins}%
1607         \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
1608     %

```

Last but not least, restore the normal line number allowed to notes for the following page.

```

1609         \bgroup%
1610         \csuse{Xnotefontsize@#1}%
1611         \global\advance\dimen\csuse{#1footins} by \baselineskip%
1612         \egroup%
1613     %

```

```

1614 % End of \protect\cs{print@Xnotes@forpages}.
1615 }%
1616 }%
1617 }%
1618 %

```


Now, `\correct@Xfootins@box`, to fix problem of last line being glued to the previous one.

```
1619 \newcommand{\correct@Xfootins@box}[1]{%
1620 %
```

We need to make correction only in case we have not printed any note in the previous page, although there was to be “normally” printed.

```
1621 \ifbool{keepforXside@#1}{%
1622 %
```

Some setting needed to do the right splitting.

```
1623 \csuse{Xnotefontsize@#1}%
1624 \splittopskip=0pt%
1625 %
```

And now, split the last line, and push in the right place.

```
1626 \global\setbox\csuse{#1footins}=\vbox{%
1627 \vsplit\csuse{#1footins} to \dimexpr\ht\csuse{#1footins}-1pt\relax%
1628 \vskip \dimexpr-0.5\baselineskip-0.5\lineskip-0.5pt\relax%
1629 \unvbox\csuse{#1footins}%
1630 }%
1631 %
```

End of the macro.

```
1632 }{}%
1633 }%
1634 %
```

And now, the same for familiar footnotes.

```
1635 \newcommand\print@notesX@forpages[1]{%
1636 \ifcempty{onlysideX@#1}{%
1637 \csuse{footstart#1}{#1}%
1638 \csuse{footgroup#1}{#1}%
1639 }%
1640 {%
1641 \ifboolexpr{%
1642 ((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page
1643 }})%
1644 or%
1645 (test {\ifcsstring{onlysideX@#1}{R}} and test{\ifnumodd{\c@page}))%
1646 }%
1647 \correct@footinsX@box{#1}%
1648 \csuse{footstart#1}{#1}%
1649 \csuse{footgroup#1}{#1}%
1650 \global\count\csuse{footins#1}=0%
1651 \global\skip\csuse{footins#1}=0pt%
1652 \csuse{notefontsizeX@#1}%
```

```

1653     \global\advance\dimen\csuse{footins#1} by -\baselineskip%
1654     \global\boolfalse{keepforsideX@#1}%
1655     }%
1656     {%
1657         \global\booltrue{keepforsideX@#1}%
1658         \global\count\csuse{footins#1}=\csuse{default@footins#1}%
1659         \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
1660         \bgroup%
1661             \csuse{notefontsizeX@#1}%
1662             \global\advance\dimen\csuse{footins#1} by \baselineskip%
1663         \egroup%
1664     }%
1665 }%
1666 }%
1667 \newcommand{\correct@footinsX@box}[1]{%
1668     \ifbool{keepforsideX@#1}{%
1669         \csuse{notefontsizeX@#1}%
1670         \splittopskip=0pt%
1671         \global\setbox\csuse{footins#1}=\vbox{%
1672             \vsplit\csuse{footins#1} to \dimexpr\ht\csuse{footins#1}-1pt\relax%
1673             \vskip \dimexpr-0.5\baselineskip-0.5\lineskip-0.5pt\relax%
1674             \unvbox\csuse{footins#1}%
1675         }%
1676     }{}%
1677 }%
1678 %

```

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.

```

1679 \list@create{\labelref@listR}
1680
1681 %

```

\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.

```

1682 \def\l@dmake@labelsR#1|#2|#3|#4|#5{%
1683     \expandafter\ifx\csname the@label\csuse{XR@prefix}#5\endcsname \relax\
1684     else
1685         \led@warn@DuplicateLabel{\csuse{XR@prefix}#5}%
1686         \fi
1687     \expandafter\gdef\csname the@label\csuse{XR@prefix}#5\endcsname
1688     {#1|#2|#3|#4|\@Rlineflag}%
1689     \ignorespaces}

```

```

1688 \AtBeginDocument{%
1689   \def\l@dmake@labelsR#1|#2|#3|#4|#5{%
1690 }
1691
1692 %

```

\@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*
1693 \WithSuffix\newcommand\sidenotemargin*[1]{%
1694   \l@dgetsidenote@margin{#1}
1695   \global\sidenote@marginR=\@l@tempcntb
1696   \global\sidenote@margin=\@l@tempcntb
1697 }
1698 \newcount\sidenote@marginR
1699 \global\sidenote@margin=\@one
1700
1701 %

```

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

```

1702 \newcommand*\affixside@noteR{%
1703   \def\sidenotecontent@{%
1704     \numgdef{\itemcount@}{0}%
1705     \def\do##1{%
1706       \ifnumequal{\itemcount@}{0}%
1707         {%
1708           \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
1709         {\appto\sidenotecontent@{\sidenotessep ##1}}%
1710       }%
1711       \numgdef{\itemcount@}{\itemcount@+1}%
1712     }%
1713     \dolistloop{\l@dcsnotetext}%
1714     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
1715     \gdef\@templ@d{%
1716       \gdef\@templ@n{\l@dcsnotetext\l@dcsnotetext@1\l@dcsnotetext@r}%
1717       \ifx\@templ@d\@templ@n \else%

```

```

1718 \if@twocolumn%
1719 \if@firstcolumn%
1720 \setl@dlp@rbox{##1}{\sidenotecontent@}%
1721 \else%
1722 \setl@drp@rbox{\sidenotecontent@}%
1723 \fi%
1724 \else%
1725 \l@dttempcntb=\sidenote@marginR%
1726 \ifnum\l@dttempcntb>\@ne%
1727 \advance\l@dttempcntb by\page@numR%
1728 \fi%
1729 \ifodd\l@dttempcntb%
1730 \setl@drp@rbox{\sidenotecontent@}%
1731 \gdef\sidenotecontent@{}%
1732 \numdef{\itemcount@}{0}%
1733 \dolistloop{\l@dcsnotetext@l}%
1734 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
1735 \setl@dlp@rbox{\sidenotecontent@}%
1736 \else%
1737 \setl@dlp@rbox{\sidenotecontent@}%
1738 \gdef\sidenotecontent@{}%
1739 \numdef{\itemcount@}{0}%
1740 \dolistloop{\l@dcsnotetext@r}%
1741 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
1742 \setl@drp@rbox{\sidenotecontent@}%
1743 \fi%
1744 \fi%
1745 \fi%
1746 }
1747
1748 %

```

XII Familiar footnotes

`\l@dbfnote` `\l@dbfnote` adds the footnote to the insert list, and `\vl@dbfnote` calls the original `\@footnotetext`. There are both defined in `reledmac`.

`\normalbfnoteX`

XIII 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.

```

\inserthangingsymbolL% \newif\ifinserthangingsymbolR
\inserthangingsymbolR% \newcommand{\inserthangingsymbolL}{%
1751 \ifinserthangingsymbol%
1752 \ifinstanzaL%
1753 \hskip \ifundefined{sza@00}{0}{\expandafter%
1754 \noexpand\csname sza@00\endcsname}\stanzaindentbase%
1755 \@hangingsymbol%
1756 \fi%
1757 \fi%
1758 }%
1759 \newcommand{\inserthangingsymbolR}{%
1760 \ifinserthangingsymbolR%
1761 \ifinstanzaR%
1762 \hskip \ifundefined{sza@00}{0}{\expandafter%
1763 \noexpand\csname sza@00\endcsname}\stanzaindentbase%
1764 \@hangingsymbol%
1765 \fi%
1766 \fi%
1767 }%
1768 %

```

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.

```

1769 \chardef\next=\catcode`&
1770 \catcode`&=\active
1771
1772 %

```

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

```

1773 \newenvironment{astanzaa}[1][\]{%
1774 \catcode`&\active
1775 \global\stanza@count\@ne\stanza@modulo\@ne
1776 \ifnum\usernamecount{sza@00}=\z@
1777 \let\stanza@hang\relax
1778 \let\endlock\relax
1779 \else
1780 \rightskip\z@ plus 1fil\relax
1781 \fi
1782 \ifnum\usernamecount{szp@00}=\z@
1783 \let\sza@penalty\relax
1784 \fi
1785 \def&{%
1786 \endlock\mbox{}%
1787 \sza@penalty
1788 \global\advance\stanza@count\@ne
1789 \@astanzaa@line}%
1790 \def\&{\@stopastanzaa}%

```

```

1791 \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}} and test{\ifstrempy
    {#1}}}%
1792     {\pstart[\at@every@stanza]}%
1793     {\pstart[#1]}%
1794 \@astanza@line
1795 \let\par\relax\ignorespaces%No paragraph in verses
1796 {}%
1797
1798 %

```

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

```

1799 \newcommand{\@stopastanza}[1][1,usedefault]{%
1800     \endlock\mbox{}%
1801     \ifboolexpr{not test{\ifdefvoid{\at@every@stop@stanza}} and test{\
    ifstrempy{#1}}}%
1802         {\pend[\at@every@stop@stanza]}%
1803         {\pend[#1]}%
1804 }%
1805 %

```

\@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.

```

1806 \newcommand*\@astanza@line{%
1807     \ifnum\value{stanzaindentsrepetition}=0
1808         \parindent=\csname sza@\number\stanza@count
1809             @\endcsname\stanzaindentbase
1810     \else
1811         \parindent=\csname sza@\number\stanza@modulo
1812             @\endcsname\stanzaindentbase
1813         \managestanza@modulo
1814     \fi
1815     \endgraf
1816     \stanza@hang%
1817     \ignorespaces}
1818
1819 %

```

Lastly reset the modified category codes.

```

1820 \catcode`\&=\next
1821
1822 %

```

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

```

1823 \thestanzaR
1824 \newcounter{stanzaL}
1825 \newcounter{stanzaR}
1826 \renewcommand{\thestanzaL}{%

```

```

1826 \textbf{\arabic{stanzaL}}}%
1827 }
1828 \renewcommand{\thestanzaR}{%
1829 \textbf{\arabic{stanzaR}}}%
1830 }
1831 %
1832 %

```

XIV Naming macros

The \TeX kernel provides `\@namedef` and `\@namuse` for defining and using macros that may have non-letters in their names. We need something similar here as we are going to need and use some numbered boxes and counters.

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

```

\setnamebox
\unhnamebox
\unvnamebox
\namebox
1833 \providecommand*\newnamebox}[1]{%
1834 \expandafter\newbox\csname #1\endcsname}
1835 \providecommand*\setnamebox}[1]{%
1836 \expandafter\setbox\csname #1\endcsname}
1837 \providecommand*\unhnamebox}[1]{%
1838 \expandafter\unhbox\csname #1\endcsname}
1839 \providecommand*\unvnamebox}[1]{%
1840 \expandafter\unvbox\csname #1\endcsname}
1841 \providecommand*\namebox}[1]{%
1842 \csname #1\endcsname}
1843 %
1844 %

```

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

```

\usenamecount
1845 \providecommand*\newnamecount}[1]{%
1846 \expandafter\newcount\csname #1\endcsname}
1847 \providecommand*\usenamecount}[1]{%
1848 \csname #1\endcsname}
1849 %
1850 %

```

XV 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 1851 \newif\ifl@dusedbabel
1852 %
```

`\l@dchecklang`

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

```
1853 \patchcmd{\bbl@set@language}%
1854 {\select@language{\language}}%
1855 {\edef\language{#1}\select@language{\language}}%
1856 {}%
1857 {}%
1858
1859 %
```

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

```
\selectlanguage \selectlanguage is a babel command. \theledlanguageL and \theledlanguageR
\l@duselanguage are the names of the languages of the left and right texts. \l@duselanguage is similar
\theledlanguageL to \selectlanguage.
\theledlanguageR
1860 \newcommand*{\l@duselanguage}[1]{}
1861 \gdef\theledlanguageL{}
1862 \gdef\theledlanguageR{}
1863
1864 %
```

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

```
1865 \AtBeginDocument{%
1866 \ifundefined{xpg@main@language}{%
1867 \ifundefined{bbl@main@language}{%
1868 %
```

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

```
1869 \l@dusedbabelfalse
1870 }{%
1871 %
```


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.

```

1872 \l@dusedbabeltrue
1873 \let\l@doldselectlanguage\selectlanguage
1874 \let\l@doldbbl@set@language\bbl@set@language
1875 \renewcommand{\selectlanguage}[1]{%
1876   \l@doldselectlanguage{#1}%
1877   \ifledRcol \gdef\theledlanguageR{#1}%
1878   \else      \gdef\theledlanguageL{#1}%
1879   \fi}
1880 %

```

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

```

1881 \renewcommand*{\l@duselanguage}[1]{%
1882   \expandafter\l@doldselectlanguage\expandafter{#1}}
1883 %

```

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

```

1884 \gdef\theledlanguageL{\bbl@main@language}%
1885 \gdef\theledlanguageR{\bbl@main@language}%
1886 }%
1887 }
1888 %

```

If use polyglossia

```

1889 { \let\old@otherlanguage\otherlanguage%
1890   \renewcommand{\otherlanguage}[2][]{%
1891     \selectlanguage[#1]{#2}%
1892     \ifledRcol \gdef\theledlanguageR{#2}%
1893     \else      \gdef\theledlanguageL{#2}%
1894     \fi}%
1895   \renewcommand{\l@duselanguage}[1]{%
1896     \csuse{no\language@name @numbers}\select@language{#1}%
1897   }%
1898   \gdef\theledlanguageL{\xpg@main@language}%
1899   \gdef\theledlanguageR{\xpg@main@language}%
1900 %

```

That is it.

```

1901 }}
1902 %

```

XVI 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.

```
1903 \newcount\l@dc@maxchunks
1904 \newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}
1905 \maxchunks{5120}
1906
1907 %
```

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

```
1908 \newcount\l@dnumpstartsR
1909
1910 %
```

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

```
\l@pscR
1911 \newcount\l@dpscL
1912 \newcount\l@dpscR
1913
1914 %
```

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

```
1915 \newcommand*{\l@dsetuprawboxes}{%
1916 \l@l@tempcntb=\l@dc@maxchunks
1917 \loop\ifnum\l@l@tempcntb>\z@
1918 \newnamebox{\l@dLcolrawbox\the\l@l@tempcntb}
1919 \newnamebox{\l@dRcolrawbox\the\l@l@tempcntb}
1920 \advance\l@l@tempcntb \m@ne
1921 \repeat}
1922
1923 %
```

`\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.

```
1924 \newcommand*{\l@dsetupmaxlinecounts}{%
1925 \l@l@tempcntb=\l@dc@maxchunks
1926 \loop\ifnum\l@l@tempcntb>\z@
1927 \newnamecount{\l@dmaxlinesinpar\the\l@l@tempcntb}
1928 \advance\l@l@tempcntb \m@ne
1929 \repeat}
1930 \newcommand*{\l@dzeromaxlinecounts}{%
```

```

1931 \begingroup
1932 \l@dttempcntb=\l@dc@maxchunks
1933 \loop\ifnum\l@dttempcntb>\z@
1934   \global\usernamecount{l@maxlinesinpar\the\l@dttempcntb}=\z@
1935   \advance\l@dttempcntb \m@ne
1936 \repeat
1937 \endgroup}
1938
1939 %

```

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

```

1940 \AtBeginDocument{%
1941   \l@dsetuprawboxes
1942   \l@dsetupmaxlinecounts
1943   \l@dzeromaxlinecounts
1944   \l@dnumpstartsL=\z@
1945   \l@dnumpstartsR=\z@
1946   \l@dpscL=\z@
1947   \l@dpscR=\z@}
1948
1949 %

```

XVII Checking text to be processed

```

\if@pstarts \check@pstarts returns \@pstartstrue if there are any unprocessed chunks.
\@pstartstrue
\@pstartsfalse
\check@pstarts
1950 \newif\if@pstarts
1951 \newcommand*{\check@pstarts}{%
1952   \@pstartsfalse
1953   \ifnum\l@dnumpstartsL>\l@dpscL
1954     \@pstartstrue
1955   \else
1956     \ifnum\l@dnumpstartsR>\l@dpscR
1957       \@pstartstrue
1958     \fi
1959   \fi
1960 }
1961
1962 %

```

```

\ifaraw@text \checkraw@text checks whether the current Left or Right box is void or not. If
\araw@texttrue one or other is not void it sets \araw@texttrue, otherwise both are void and it sets
\araw@textfalse \araw@textfalse.
\checkraw@text
1963 \newif\ifaraw@text
1964 \newcommand*{\checkraw@text}{%
1965   \araw@textfalse

```

```

1966 \ifvbox\namebox{1@dLcolrawbox\the\l@dpscL}
1967 \araw@texttrue
1968 \else
1969 \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}
1970 \araw@texttrue
1971 \fi
1972 \fi
1973 }
1974
1975 %

```

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

```

1976 \newcommand*{\@writelinesinparL}{%
1977 \edef\next{%
1978 \write\linenum@out{\string\@pend[\the\@donereallinesL]}}%
1979 \next
1980 \global\@donereallinesL \z@}
1981 \newcommand*{\@writelinesinparR}{%
1982 \edef\next{%
1983 \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}%
1984 \next
1985 \global\@donereallinesR \z@}
1986
1987 %

```

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

```

1988 \newcommand*{\@writepageofparL}[0]{%
1989 \ifnum\@donereallinesL=\z@%
1990 \edef\next{%
1991 \write\linenum@out{\string\@pstart{\the\l@dpscL}{\the\c@page}{\the\
1992 numpagelinesL}}%
1993 }%
1994 \next%
1995 \fi%
1996 }%
1997 \newcommand*{\@writepageofparR}[0]{%
1998 \ifnum\@donereallinesR=\z@%
1999 \edef\next{%
2000 \write\linenum@outR{\string\@pstartR{\the\l@dpscR}{\the\c@page}{\the\
2001 numpagelinesR}}%
2002 }%
2003 \next%
2004 \fi%
2005 }%
2006 %

```

XVIII Parallel columns

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

```

\@eledsectionR
2005 \newsavebox{\@eledsectionL}%
2006 \newsavebox{\@eledsectionR}%
2007 %

```

`\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.

```

2008 \newcommand*{\Columns}{%
2009   \ifl@dpairing%
2010     \led@err@Columns@InsideEnv%
2011   \fi%
2012   \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
2013     \led@err@Columns@WithoutEnv%
2014   \else%
2015     \global\l@dprintingcolumnstrue%
2016     \eledsection@correcting@skip=-\baselineskip% Correction for sections'
titles
2017     \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else
2018       \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
2019     \fi
2020   %

```

Start a group and zero counters, etc.

```

2021 \begingroup
2022   \l@dzeropenalties
2023   \endgraf\global\num@lines=\prevgraf
2024     \global\num@linesR=\prevgraf
2025   \global\par@line=\z@
2026   \global\par@lineR=\z@
2027   \global\l@dpscL=\z@
2028   \global\l@dpscR=\z@
2029   \get@familiarfootnote@number%
2030 %

```

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

```

2031   \check@pstarts
2032   \loop\if@pstarts
2033     \global\pstartnumtrue
2034     \global\pstartnumRtrue
2035 %

```

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.

```

2036   \global\advance\l@dpscL \@ne
2037   \global\advance\l@dpscR \@ne

```

```

2038 \restore@pstartL@pc%
2039 \restore@pstartR@pc%
2040 %

```

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

```

2041 \Columns@print@before@pstart%
2042 %

```

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

```

2043 \checkraw@text
2044 { \loop\ifaraw@text
2045 %

```

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

```

2046 \l@duselanguage{\theledlanguageL}%
2047 \do@lineL
2048 \xifinlist{\the\l@dpscL}{\eled@sections@@}
2049 {%
2050 \ifdefstring{\@eledsectmark}{L}%
2051 {\csuse{eled@sectmark@the\l@dpscL}%
2052 }}%
2053 \global\csundef{eled@sectmark@the\l@dpscL}%
2054 \savebox{\@eledsectionL}{\parbox[t]{}[t]{\Lcolwidth}{\vbox
2055 {} \print@eledsectionL}}}%\vbox{}-> prevent alignment troubles with RTL
2056 language
2057 }%
2058 {}%
2059 \l@duselanguage{\theledlanguageR}%
2060 \do@lineR
2061 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}
2062 {%
2063 \ifdefstring{\@eledsectmark}{R}%
2064 {\csuse{eled@sectmark@the\l@dpscR}%
2065 }}%
2066 \global\csundef{eled@sectmark@the\l@dpscR}%
2067 \savebox{\@eledsectionR}{\parbox[t]{}[t]{\Rcolwidth}{\vbox
2068 {} \print@eledsectionR}}}%\vbox{}-> prevent alignment troubles with RTL
2069 language
2070 }%
2071 {}%
2072 \hb@xt@ \hsize{%
2073 \ifdefstring{\columns@position}{L}{\hfill }%
2074 \unhbox\l@dleftbox%
2075 \ifhbox{\@eledsectionL}%
2076 \usebox{\@eledsectionL}%
2077 \fi%
2078 \print@columnseparator%
2079 \unhbox\l@drightbox%

```

```

2075         \ifhbox\@eledsectionR%
2076             \usebox{\@eledsectionR}%
2077         \fi%
2078         \ifdefstring{\columns@position}{R}{\hfill}%
2079     }%
2080     \checkraw@text
2081     \checkverseL
2082     \checkverseR
2083     \checkpb@columns
2084     \repeat}
2085 %

```

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.

```

2086     \@writelinesinparL
2087     \@writelinesinparR
2088     \check@pstarts
2089     \ifbypstart@%
2090         \write\linenum@out{\string\@set[1]}
2091         \resetprevline@
2092     \fi
2093     \ifbypstart@R
2094         \write\linenum@outR{\string\@set[1]}
2095         \resetprevline@
2096     \fi
2097     \Columns@print@after@pend%
2098     \repeat
2099 %

```

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.

```

2100     \flush@notes
2101     \flush@notesR
2102     \endgroup
2103 %
2104 \global\l@dpscL=\z@
2105 \global\l@dpscR=\z@
2106 \global\l@dnpstartsL=\z@
2107 \global\l@dnpstartsR=\z@
2108 \global\l@dprintingcolumnsfalse%
2109 \ignorespaces
2110 \global\instanzaLfalse%
2111 \global\instanzaRfalse%
2112 \fi}
2113
2114 %

```

`\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).

```

2115 \def\print@columnseparator{%
2116   \ifdim\beforecolumnseparator<0pt%
2117     \hfill%
2118   \else%
2119     \hspace{\beforecolumnseparator}%
2120   \fi%
2121   \columnseparator%
2122   \ifdim\aftercolumnseparator<0pt%
2123     \hfill%
2124   \else%
2125     \hspace{\beforecolumnseparator}%
2126   \fi%
2127 }%
2128 %

```

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

```

2129
2130 \newcommand{\checkpb@columns}{%
2131   \newif\if@pb
2132   \newif\if@nopb
2133   \IfStrEq{\led@pb@setting}{before}{
2134     \numdef{\next@absline}{\the\absline@num+1}%
2135     \numdef{\next@abslineR}{\the\absline@numR+1}%
2136     \xifinlistcs{\next@absline}{l@prev@pb}{\@pbtrue}{}%
2137     \xifinlistcs{\next@abslineR}{l@prev@pbR}{\@pbtrue}{%
2138     \xifinlistcs{\next@absline}{l@prev@nopb}{\@nopbtrue}{}%
2139     \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\@nopbtrue}{%
2140     }{}
2141     \IfStrEq{\led@pb@setting}{after}{
2142       \xifinlistcs{\the\absline@num}{l@prev@pb}{\@pbtrue}{}%
2143       \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\@pbtrue}{%
2144       \xifinlistcs{\the\absline@num}{l@prev@nopb}{\@nopbtrue}{}%
2145       \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\@nopbtrue}{%
2146       }{}
2147       \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
2148       \if@pb\pagebreak[4]\fi
2149     }
2150   }%

```

`\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).


```

2151 \newcommand*{\columnseparator}{%
2152   \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}}
2153 \newdimen\columnrulewidth
2154   \columnrulewidth=\z@
2155
2156 %

```

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

```

\columns@position
2157 \newcommand*{\columnspan}[1]{%
2158   \xdef\columns@position{#1}%
2159 }%
2160 \xdef\columns@position{R}%
2161 %

```

`\beforecolumnseparator` `\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`.

```

2162 \newlength{\beforecolumnseparator}%
2163 \setlength{\beforecolumnseparator}{-2pt}%
2164
2165 \newlength{\aftercolumnseparator}%
2166 \setlength{\aftercolumnseparator}{-2pt}%
2167
2168 %

```

`\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.

```

\setwidthliketwocolumns@L
\setpositionliketwocolumns@L
\setwidthliketwocolumns@C
\setpositionliketwocolumns@C
\setwidthliketwocolumns@R
\setpositionliketwocolumns@R
\setwidthliketwocolumns@R
2169 \newcommand{\setwidthliketwocolumns@L}{%
2170   % Temporary dimension, initially equal to the standard hsize, i.e. text
width
2171   % \begin{macrocode}
2172   \newdimen\temp%
2173   \temp=\hsize%
2174   %

```

Hsize : Left + Right width

```

2175   \hsize=\Lcolwidth%
2176   \advance\hsize\Rcolwidth%
2177   %

```

Now, calculating the remaining space

```

2178   \advance\temp-\hsize%
2179   %

```

And multiply the hsize by 2/3 of this space

```

2180 \multiply\temp by 2%
2181 \divide\temp by 3%
2182 \advance\hsize\temp%
2183 }%
2184
2185 \newcommand{\setpositionliketwocolumns@L}{%
2186 \renewcommand{\ledrlfill}{\hfill}%
2187 }%
2188
2189 \newcommand{\setnotespositionliketwocolumns@L}{%
2190 }%
2191
2192
2193 %

```

```

2194 \newcommand{\setwidthliketwocolumns@C}{%
2195 % Temporary dimension, initially equal to the standard hsize, i.e. text
2196 % width
2197 %

```

```

2197 \newdimen\temp%
2198 \temp=\hsize%
2199 % Hsize : Left + Right width
2200 %

```

```

2201 \hsize=\Lcolwidth%
2202 \advance\hsize\Rcolwidth%
2203 % Now, calculating the remaining space
2204 %

```

```

2205 \advance\temp-\hsize%
2206 %

```

And multiply the hsize by 1/2 of this space

```

2207 \divide\temp by 2%
2208 \advance\hsize\temp%
2209 }%
2210
2211 \newcommand{\setpositionliketwocolumns@C}{%
2212 \doinsidelinehook{\hfill}%
2213 \renewcommand{\ledrlfill}{\hfill}%
2214 }%
2215
2216 \newcommand{\setnotespositionliketwocolumns@C}{%
2217 \newdimen\temp%
2218 \newdimen\tempa%
2219 \temp=\hsize%
2220 \tempa=\Lcolwidth%

```

```

2221 \advance\tempa\Rcolwidth%
2222 \advance\temp-\tempa%
2223 \divide\temp by 2%
2224 \leftskip=\temp%
2225 \rightskip=-\temp%
2226 }%
2227
2228 \newcommand{\setwidthliketwocolumns@R}{%
2229 %

```

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

```

2230 \newdimen\temp%
2231 \temp=\hsize%
2232 %

```

Hsize : Left + Right width

```

2233 \hsize=\Lcolwidth%
2234 \advance\hsize\Rcolwidth%
2235 %

```

Now, calculating the remaining space

```

2236 \advance\temp-\hsize%
2237 %

```

And multiply the hsize by 2/3 of this space

```

2238 \multiply\temp by 2%
2239 \divide\temp by 3%
2240 \advance\hsize\temp%
2241 }%
2242
2243 \newcommand{\setpositionliketwocolumns@R}{%
2244 \doinsidelinehook{\hfill}%
2245 }%
2246
2247 \newcommand{\setnotespositionliketwocolumns@R}{%
2248 \newdimen\temp%
2249 \newdimen\tempa%
2250 \temp=\hsize%
2251 \tempa=\Lcolwidth%
2252 \advance\tempa\Rcolwidth%
2253 \advance\temp-\tempa%
2254 \divide\temp by 2%
2255 \leftskip=\temp%
2256 \rightskip=-\temp%
2257 }%
2258
2259 %

```

`\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.

```

2260 \newcommand{\Columns@print@before@pstart}{%
2261   \ifboolexpr{%
2262     test{\ifcsstring{before@pstartL@the\l@dpscl}{\at@every@pstart}}}%
2263     and test {\ifcsstring{before@pstartR@the\l@dpscr}{\at@every@pstart}}}%
2264     and test {\ifdefempty{\at@every@pstart}}}%
2265     {%
2266     {%
2267       \hb@xt@ \hsize{%
2268         \ifdefstring{\columns@position}{L}{\hfill }%
2269         \par\parbox[t] [] [t]{\Lcolwidth}{%
2270           \csuse{before@pstartL@the\l@dpscl}%
2271         }%
2272         \print@columnseparator%
2273         \parbox[t] [] [t]{\Rcolwidth}{%
2274           \set@sectcountR%
2275           \csuse{before@pstartR@the\l@dpscr}%
2276         }%
2277         \ifdefstring{\columns@position}{R}{\hfill }%
2278       }%
2279     }%
2280     \global\csundef{before@pstartL@the\l@dpscl}%
2281     \global\csundef{before@pstartR@the\l@dpscr}%
2282   }%
2283 \newcommand{\Columns@print@after@pend}{%
2284   \ifboolexpr{%
2285     test{\ifcsstring{after@pendL@the\l@dpscl}{\at@every@pend}}}%
2286     and test {\ifcsstring{after@pendR@the\l@dpscr}{\at@every@pend}}}%
2287     and test {\ifdefempty{\at@every@pend}}}%
2288     {%
2289     {%
2290       \hb@xt@ \hsize{%
2291         \ifdefstring{\columns@position}{L}{\hfill }%
2292         \parbox[t] [] [t]{\Lcolwidth}{%
2293           \csuse{after@pendL@the\l@dpscl}%
2294         }%
2295         \print@columnseparator%
2296         \parbox[t] [] [t]{\Rcolwidth}{%
2297           \set@sectcountR%
2298           \csuse{after@pendR@the\l@dpscr}%
2299         }%
2300         \ifdefstring{\columns@position}{R}{\hfill }%
2301       }%
2302     }%
2303     \global\csundef{after@pendL@the\l@dpscl}%
2304     \global\csundef{after@pendR@the\l@dpscr}%
2305   }%

```

2306 %

XIX Parallel pages

This is considerably more complicated than parallel columns.

XIX.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`

```

2307 \newcount\numpagelinesL
2308 \newcount\numpagelinesR
2309 \newcount\l@dminpagelines
2310
2311 %
```

XIX.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.

```

2312 \newcommand*{\Pages}[1][1,usedefault]{%
2313   \ifl@dpairing%
2314     \led@err@Pages@InsideEnv%
2315   \fi%
2316   \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
2317     \led@err@Pages@WithoutEnv%
2318   \else%
2319     \ifstrequal{#1}{mainmatter}{\Pages@mainmattertrue}{\Pages@mainmatterfalse}%
2320   }%
2321   \eledsection@correcting@skip=-2\baselineskip% line correcting for section
  titles.
2322   \parledgroup@notespacing@set@correction%
2323   \typeout{}%
2324   \typeout{***** PAGES *****}%
2325   \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else%
2326     \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
2327   \fi%
  %
```

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

```

2328 \cleartol@devenpage%
2329 \global\l@dprintingpagetrue%
2330 \begingroup%
2331 %
```

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

```

2332 \setlength{\Lcolwidth}{\textwidth}%
2333 \setlength{\Rcolwidth}{\textwidth}%
2334 %
2335 \l@dzeroopenalties%
2336 \endgraf\global\num@lines=\prevgraf%
2337 \global\num@linesR=\prevgraf%
2338 \global\par@line=\z@%
2339 \global\par@lineR=\z@%
2340 \global\l@dpscL=\z@%
2341 \global\l@dpscR=\z@%
2342 \writtenlinesLfalse%
2343 \writtenlinesRfalse%
2344 \get@familiarfootnote@number%
2345 %

```

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

```

2346 \let\print@Xnotes\print@Xnotes@forpages%
2347 \let\print@notesX\print@notesX@forpages%
2348 %

```

Check if there are chunks to be processed.

```

2349 \check@pstarts%
2350 \loop\if@pstarts%
2351 %

```

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

```

2352 \global\advance\l@dpscL \@ne%
2353 \global\advance\l@dpscR \@ne%
2354 %

```

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

```

2355 \getlinesfromparlistL%
2356 \getlinesfromparlistR%
2357 \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}%
2358 {\usernamecount{l@dmaxlinesinpar\the\l@dpscL}}%
2359 \check@pstarts%
2360 \repeat%
2361 %

```

Zero the counts again, ready for the next bit.

```

2362 \global\l@dpscL=\z@%
2363 \global\l@dpscR=\z@%
2364 %

```

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

```

2365 \getlinesfrompagelistL%
2366 \getlinesfrompagelistR%
2367 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
2368     {\l@dminpagelines}%
2369 %

```

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.

```

2370 \check@pstarts%
2371 \if@pstarts%
2372 %

```

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

```

2373 \global\advance\l@dpscL \@ne%
2374 \global\advance\l@dpscR \@ne%
2375 \restore@pstartL@pc%
2376 \restore@pstartR@pc%
2377 %

```

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

```

2378 \global\@donereallinesL=\z@%
2379 \global\@donetotallinesL=\z@%
2380 \global\@donereallinesR=\z@%
2381 \global\@donetotallinesR=\z@%
2382 %

```

Start a loop over the boxes (chunks).

```

2383 \checkraw@text%
2384 %
2385 % \begin group
2386 { \loop\ifaraw@text%
2387 %

```

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

```

2388 \checkpageL%
2389 \l@duselanguage{\theledlanguageL}%
2390 { \loop\ifl@dsamepage%
2391 %

```

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

```

2392 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}%
2393 \csuse{before@pstartL@the\l@dpscL}%
2394 \global\csundef{before@pstartL@the\l@dpscL}%

```

```

2395 \do@lineL%
2396 \xifinlist{\the\l@dpscL}{\eled@sections@@}
2397 {\print@eledsectionL}%
2398 {}%
2399 \advance\numpagelinesL \@ne%
2400 %

```

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

```

2401 \ifshiftedpstarts%
2402 \ifdim\ht\l@dleftbox>0pt%
2403 \parledgroup@correction@notespacing{L}%
2404 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2405 \else%
2406 \unless\ifadvancedshiftedpstarts%
2407 \dimen0=\pagetotal%
2408 \advance\dimen0 by \baselineskip%
2409 \global\pagetotal=\dimen0%
2410 \else%
2411 \ifnomaxlines%
2412 \numdef{\@tmp}{\the\l@dpscL+1}%
2413 \ifcsdef{minpage@pstart@\@tmp}{%
2414 \ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}}%
2415 {\dimen0=\pagetotal%
2416 \advance\dimen0 by \baselineskip%
2417 \global\pagetotal=\dimen0%
2418 }%
2419 {}%
2420 }{}%
2421 \fi%
2422 \fi%
2423 \fi%
2424 \else%
2425 \parledgroup@correction@notespacing{L}%
2426 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2427 \fi%
2428 %

```

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

```

2429 \get@nextboxL%

```



```

2430      \global\l@dskipversenumberfalse%
2431      \ifprint@last@after@pendL%
2432        \csuse{after@pendL@the\l@dpscL}%
2433        \global\csundef{after@pendL@the\l@dpscL}%
2434      \fi%
2435      \checkpageL%
2436      \checkverseL%
2437      \checkpbL%
2438      \repeat%
2439 %

```

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.

```

2440      \ifl@dpagefull%
2441        \@writelinesonpageL{\the\numpagelinesL}%
2442      \else%
2443        \@writelinesonpageL{1000}%
2444      \fi%
2445 %

```

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.

```

2446      \numpagelinesL \z@%
2447      \parledgroup@correction@notespacing@init%
2448      \clearl@dleftpage }%
2449 %

```

Now do the same for the right text.

```

2450      \checkpageR%
2451      \l@duselanguage{\theledlanguageR}%
2452 {
2453   \loop\ifl@dsamepage%
2454     \set@sectcountR%
2455     \ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{}%
2456     \csuse{before@pstartR@the\l@dpscR}%
2457     \global\csundef{before@pstartR@the\l@dpscR}%
2458     \do@lineR%
2459     \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
2460     {\print@eledsectionR}%
2461     {}%
2462     \advance\numpagelinesR \@ne%
2463     \ifshiftedpstarts%
2464       \ifdim\ht\l@drightbox>Opt%
2465         \parledgroup@correction@notespacing{R}%
2466         \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2467       \else%
2468         \unless\ifadvancedshiftedpstarts%

```

```

2468         \dimen0=\pagetotal%
2469         \advance\dimen0 by \baselineskip%
2470         \global\pagetotal=\dimen0%
2471     \else%
2472         \ifnomaxlines%
2473             \numdef{\@tmp}{\the\l@dpscR+1}%
2474             \ifcsdef{minpage@pstart@\@tmp}{%
2475                 \ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}}%
2476                 {\dimen0=\pagetotal%
2477                 \advance\dimen0 by \baselineskip%
2478                 \global\pagetotal=\dimen0%
2479                 }%
2480                 {}%
2481             }%
2482         \fi%
2483     \fi%
2484 \fi%
2485 \else%
2486     \parledgroup@correction@notespacing{R}%
2487     \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2488 \fi%
2489 \get@nextboxR%
2490 \global\l@dskipversenumberRfalse%
2491 \ifprint@last@after@pendR%
2492     \csuse{after@pendR@\the\l@dpscR}%
2493     \global\csundef{after@pendR@\the\l@dpscR}%
2494 \fi%
2495 \checkpageR%
2496 \checkverseR%
2497 \checkpbR%
2498 \repeat%
2499 \ifl@dpagfull%
2500     \@writelinesonpageR{\the\numpagelinesR}%
2501 \else%
2502     \@writelinesonpageR{1000}%
2503 \fi%
2504 \numpagelinesR=\z@%
2505 \parledgroup@correction@notespacing@init%
2506 %

```

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

```

2507         \clearl@drightpage}%
2508 %

```

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.

```

2509         \checkraw@text%
2510         \ifaraw@text%
2511             \getlinesfrompagelistL%

```

```

2512 \getlinesfrompagelistR%
2513 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
2514 \l@dminpagelines}%
2515 \fi%
2516 \repeat}%
2517 %

```

We have now output the text from all the chunks.

```

2518 \fi%
2519 %

```

Make sure that there are no inserts hanging around.

```

2520 \flush@notes%
2521 \flush@notesR%
2522 \endgroup%
2523 %

```

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

```

2524 \global\l@dpscL=\z@%
2525 \global\l@dpscR=\z@%
2526 \global\l@dnpstartL=\z@%
2527 \global\l@dnpstartR=\z@%
2528 \global\instanzaLfalse%
2529 \global\instanzaRfalse%
2530 \global\l@dprintingpagesfalse%
2531 \finish@Pages@notes%Needed to prevent final notes overlap line number
2532 \ignorespaces\fi}
2533
2534
2535 %

```

XIX.3 Ensure all notes be 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 is no more long notes left for the next pages.

```

2536 \newcommand{\finish@Pages@notes}{%
2537 \def\do##1{%
2538 %

```

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

```

2539 \ifnocritical{%
2540 \global\newnamebox{##1footins}
2541 \fi
2542 \ifnofamiliar{%
2543 \global\newnamebox{footins##1}
2544 \fi
2545 %

```

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

```

2546 \ifvoid\csuse{##1footins}%
2547 \ifvoid\csuse{footins##1}\else%
2548 \newpage\null%
2549 \listbreak%
2550 \fi%
2551 \else%
2552 \newpage\null%
2553 \listbreak%
2554 \fi%
2555 }%
2556 \dolistloop{\@series}%
2557 }%
2558 %

```

XIX.4 Struts

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

```

\ledstrutR
2559 \newcommand*\ledstrutL{}
2560 \newcommand*\ledstrutR{}
2561
2562 %

```

XIX.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.

```

2563 \providecommand{\cleartoevenpage}[1][\@empty]{%
2564 \clearpage
2565 \ifodd\c@page\hbox{##1}\clearpage\fi}
2566
2567 \newcommand*\cleartol@devenpage{%
2568 \ifdim\pagetotal<\topskip% on an empty page
2569 \else
2570 \clearpage
2571 \Pages@mainmatter%
2572 \fi
2573 \ifodd\c@page%
2574 \ifprevpgnotnumbered%
2575 \addtocounter{par@page}{-1}%
2576 \ifdef\prevpgstyle{\thispagestyle{\prevpgstyle}}}%
2577 \fi%
2578 \hbox{}\clearpage%
2579 \fi%
2580 }%
2581 %

```

`\clearl@dleftpage` and `\clearl@drighthouse` 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`

```

2582 \newcommand*\clearl@dleftpage}{%
2583   \ifdim\pagetotal=0pt\hbox{}\fi%
2584   \newpage%
2585   \ifodd\c@page\else
2586     \led@err@LeftOnRightPage
2587     \hbox{}%
2588     \cleardoublepage
2589   \fi}
2590
2591 \newcommand*\clearl@drighthouse}{%
2592   \ifdim\pagetotal=0pt\hbox{}\fi%
2593   \newpage%
2594   \ifodd\c@page
2595     \led@err@RightOnLeftPage
2596     \hbox{}%
2597     \cleartoevenpage
2598   \fi}
2599
2600 %

```

XIX.6 Lines managing

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

```

2601 \newcommand*\getlinesfromparlistL}{%
2602   \ifx\linesinpar@listL\empty
2603     \gdef\cs@linesinparL{0}%
2604   \else
2605     \gl@p\linesinpar@listL\to\cs@linesinparL
2606   \fi}
2607 \newcommand*\getlinesfromparlistR}{%
2608   \ifx\linesinpar@listR\empty
2609     \gdef\cs@linesinparR{0}%
2610   \else
2611     \gl@p\linesinpar@listR\to\cs@linesinparR
2612   \fi}
2613
2614 %

```

`\getlinesfrompagelistL` and `\getlinesfrompagelistR` get the next entry from the `\linesonpage@listL` and `\linesonpage@listR` and puts it into `\cs@linesonpageL` and `\cs@linesonpageR`.

puts it into `\@cs@linesonpageL`; if the list is empty, it sets `\@cs@linesonpageL` to 1000. Similarly for `\getlinesfrompagelistR`.

```

2615 \newcommand*\getlinesfrompagelistL{%
2616   \ifx\linesonpage@listL\empty
2617     \gdef\@cs@linesonpageL{1000}%
2618   \else
2619     \gl@p\linesonpage@listL\to\@cs@linesonpageL
2620   \fi}
2621 \newcommand*\getlinesfrompagelistR{%
2622   \ifx\linesonpage@listR\empty
2623     \gdef\@cs@linesonpageR{1000}%
2624   \else
2625     \gl@p\linesonpage@listR\to\@cs@linesonpageR
2626   \fi}
2627
2628 %

```

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

```

2629 \newcommand*\@writelinesonpageL[1]{%
2630   \edef\next{\write\linenum@out{\string\@lopL{#1}}}%
2631   \next}
2632 \newcommand*\@writelinesonpageR[1]{%
2633   \edef\next{\write\linenum@outR{\string\@lopR{#1}}}%
2634   \next}
2635
2636 %

```

`\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>`.

```

2637 \newcommand*\l@dcalc@maxoftwo[3]{%
2638   \ifnum #2>#1\relax
2639     #3=#2\relax
2640   \else
2641     #3=#1\relax
2642   \fi}
2643 \newcommand*\l@dcalc@minoftwo[3]{%
2644   \ifnum #2<#1\relax
2645     #3=#2\relax
2646   \else
2647     #3=#1\relax
2648   \fi}
2649
2650 %

```

XIX.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@dpagfull is set FALSE and
\l@dsamepagefalse \ifl@dsamepage is set TRUE. If the page is spatially full then \ifl@dpagfull is set
\ifl@dpagfull TRUE and \ifl@dsamepage is set FALSE. If it is not spatially full but the maximum
\l@dpagfulltrue number of lines have been output then both \ifl@dpagfull and \ifl@dsamepage
\l@dpagfullfalse are set FALSE.
\checkpageL
\checkpageR
2651 \newif\ifl@dsamepage
2652 \l@dsamepagetrue
2653 \newif\ifl@dpagfull
2654
2655 \newcommand*{\checkpageL}{%
2656 \l@dpagfulltrue
2657 \l@dsamepagetrue
2658 \check@goal
2659 \ifdim\pagetotal<\ledthegoal
2660 \ifnum\numpagelinesL<\l@dmminpagelines
2661 \else
2662 \ifnomaxlines%
2663 \else%
2664 \l@dsamepagefalse%
2665 \l@dpagfullfalse%
2666 \fi%
2667 \fi
2668 \else
2669 \l@dsamepagefalse
2670 \l@dpagfulltrue
2671 \fi%
2672 \ifprint@last@after@pendL%
2673 \l@dpagfullfalse%
2674 \l@dsamepagefalse%
2675 \print@last@after@pendLfalse%
2676 \fi%
2677 }%
2678
2679 \newcommand*{\checkpageR}{%
2680 \l@dpagfulltrue
2681 \l@dsamepagetrue
2682 \check@goal
2683 \ifdim\pagetotal<\ledthegoal
2684 \ifnum\numpagelinesR<\l@dmminpagelines
2685 \else
2686 \ifnomaxlines%
2687 \else%
2688 \l@dsamepagefalse%
2689 \l@dpagfullfalse%
2690 \fi%
2691 \fi

```

```

2692 \else
2693   \l@dsamepagefalse
2694   \l@dpagetrue
2695 \fi%
2696 \ifprint@last@after@pendR%
2697   \l@dpagetruefalse%
2698   \l@dsamepagefalse%
2699   \print@last@after@pendRfalse%
2700 \fi%
2701 }%
2702
2703 %

```

`\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`.

```

2704 \newcommand{\checkpbL}{
2705   \IfStrEq{\ledpb@setting}{after}{
2706     \xifinlistcs{\the\absline@num}{l@prev@pb}{\l@dpagetrue\l@dsamepagefalse}{
2707       \xifinlistcs{\the\absline@num}{l@prev@nopb}{\l@dpagetruefalse\l@dsamepagetrue}{
2708         }{}
2709       \IfStrEq{\ledpb@setting}{before}{
2710         \numdef{\next@absline}{\the\absline@num+1}
2711         \xifinlistcs{\next@absline}{l@prev@pb}{\l@dpagetrue\l@dsamepagefalse}{
2712           \xifinlistcs{\next@absline}{l@prev@nopb}{\l@dpagetruefalse\l@dsamepagetrue}{
2713             }{}
2714         }
2715       }
2716 \newcommand{\checkpbR}{
2717   \IfStrEq{\ledpb@setting}{after}{
2718     \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\l@dpagetrue\l@dsamepagefalse}{
2719       \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\l@dpagetruefalse\l@dsamepagetrue}{
2720         }{}
2721       \IfStrEq{\ledpb@setting}{before}{
2722         \numdef{\next@abslineR}{\the\absline@numR+1}
2723         \xifinlistcs{\next@abslineR}{l@prev@pbR}{\l@dpagetrue\l@dsamepagefalse}{
2724           \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\l@dpagetruefalse\l@dsamepagetrue}{
2725             }{}
2726         }
2727   }

```


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

```

2728 \newcommand{\checkverseL}{
2729 \ifinstanzaL
2730 \iflednopbinverse
2731 \ifinserthangingsymbol
2732 \numgdef{\prev@abslineverse}{\the\absline@num-1}
2733 \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{}
2734 \IfStrEq{\led@pb@setting}{before}{\ifnum\umpagelinesL<3\ledpbnum{\
prev@abslineverse}\fi}{}}
2735 \fi
2736 \fi
2737 \fi
2738 }
2739 \newcommand{\checkverseR}{
2740 \ifinstanzaR
2741 \iflednopbinverse
2742 \ifinserthangingsymbolR
2743 \numgdef{\prev@abslineverse}{\the\absline@numR-1}
2744 \IfStrEq{\led@pb@setting}{after}{\lednopbnumR{\prev@abslineverse}}{}
2745 \IfStrEq{\led@pb@setting}{before}{\ifnum\umpagelinesR<3\ledpbnumR{\
prev@abslineverse}\fi}{}}
2746 \fi
2747 \fi
2748 \fi
2749 }
2750 %

```

`\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
2751 \newdimen\ledthegoal
2752 \ifshiftedpstarts
2753 \newcommand*{\@goalfraction}{0.95}
2754 \else
2755 \newcommand*{\@goalfraction}{0.9}
2756 \fi
2757
2758 \newcommand*{\check@goal}{%
2759 \ledthegoal=\@goalfraction\pagegoal}
2760 \newcommand{\setgoalfraction}[1]{%
2761 \xdef\@goalfraction{#1}%
2762 }
2763 %

```

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

```

\ifwrittenlinesL
2764 \newif\ifwrittenlinesL

```

```

2765 \newif\ifwrittenlinesR
2766
2767 %

```

XIX.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.

```

2768 \newif\if@getnextbox%
2769 %

```

\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.

```

2770 \newcommand*\get@nextboxL{%
2771   \ifvbox\namebox{1@dLcolrawbox\the\1@dpscl}% box is not empty
2772   %

```

The current box is not empty; do nothing.

```

2773   \else%                                box is empty
2774   %

```

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

```

2775   \@getnextboxtrue%Should be local, but be cautious
2776   %

```

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

```

2777   \ifnum\usenamecount{1@dmaxlinesinpar\the\1@dpscl}>\@donetotallinesL
2778   \parledgroup@notes@endL%
2779   \unless\ifnosyncpstarts%
2780   \@getnextboxfalse%
2781   %

```

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.

```

2782   \ifnomaxlines%
2783   \ifdim\pagetotal<\ledthegoal%
2784   \numdef{\@tmp}{\1@dpscl+1}%
2785   \ifcsdef{afterlines@pstart@\@tmp R}{%

```

```

2786 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp R}}
2787 %
2788 {}%
2789 {\ifcsdef{minpage@pstart@\@tmp}%
2790 {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2791 {\ifnum\numpagelinesL=\l@dminpagelines%
2792 \getnextboxtrue%
2793 \fi%
2794 }%
2795 {\@getnextboxtrue}}%
2796 {\@getnextboxtrue}}%
2797 }%
2798 {}%
2799 \fi%
2800 \fi%
2801 \fi%
2802 \else%
2803 \ifnomaxlines%
2804 \numdef{\@tmp}{\the\l@dpscl+1}%
2805 \ifcsdef{minpage@pstart@\@tmp}%
2806 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2807 {\ifdimgreater{\pagetotal}{\ledthegoal}%
2808 {\@getnextboxtrue}%
2809 {\@getnextboxfalse}%
2810 }%
2811 {\@getnextboxtrue}%
2812 }{}%
2813 \fi%
2814 \fi%
2815 %

```

Sufficient lines have been output.

```

2816 \if@getnextbox%
2817 \ifnum\usernamecount\l@dmaxlinesinpar\the\l@dpscl=\@donetotallinesL
2818 \parledgroup@notes@endL
2819 \fi
2820 \ifwrittenlinesL\else
2821 %

```

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

```

2822 \@writelinesinparL
2823 \writtenlinesLtrue
2824 \fi
2825 \ifnum\l@dnumstartsL>\l@dpscl
2826 %

```

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.

```

2827     \writtenlinesLfalse
2828     \ifbypstart@
2829         \global\line@num=0%
2830         \resetprevline@%
2831     \fi
2832 % Add the content of the optional argument of the previous \protect\cs{pend
2833 }.
2834 %     \begin{macrocode}
2835     \csuse{after@pendL@the\l@dpscL}%
2836     \global\csundef{after@pendL@the\l@dpscL}%
2837 %

```

Check the number of lines

```

2837     \l@dcalc@maxoftwo{\the\usernamecount{l@dmaxlinesinpar\the\l@dpscL}}%
2838                     {\the\@donetotallinesL}%
2839                     {\usernamecount{l@dmaxlinesinpar\the\l@dpscL}}%
2840     \global\@donetotallinesL \z@
2841 %

```

Go to the next pstart

```

2842     \global\advance\l@dpscL \@ne
2843     \global\pstartnumtrue%
2844     \restore@pstartL@pc%
2845 %

```

Add notes of parallel ledgroup.

```

2846     \parledgroup@notes@endL
2847     \parledgroup@correction@notes@spacing@final{L}
2848     \else
2849 %

```

```

2850     \print@last@after@pendLtrue%
2851     \fi
2852 \fi
2853 \fi}
2854 %

```

```

2855 \newcommand*{\get@nextboxR}{%
2856     \ifvbox\namebox{l@dRcolrawbox\the\l@dpscR}% box is not empty
2857     \else% box is empty
2858         \@getnextboxtrue%
2859         \ifnum\usernamecount{l@dmaxlinesinpar\the\l@dpscR}>\@donetotallinesR
2860             \parledgroup@notes@endR
2861             \unless\ifnosyncpstarts%
2862                 \@getnextboxfalse%
2863             \ifnomaxlines%
2864                 \ifdim\pagetotal<\ledthegoal%
2865                     \numdef{\@tmp}{\l@dpscR+1}%
2866                     \ifcsdef{afterlines@pstart@\@tmp L}{%

```

```

2867 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp L}}
2868 %
2869 {}%
2870 {\ifcsdef{minpage@pstart@\@tmp}%
2871   {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2872     {\ifnum\numpagelinesR=\l@dminpagelines%
2873       \@getnextboxtrue%
2874       \fi%
2875     }%
2876     {\@getnextboxtrue}}%
2877   }%
2878 }%
2879 {}%
2880 \fi%
2881 \fi%
2882 \fi%
2883 \else%
2884 \ifnomaxlines%
2885 \numdef{\@tmp}{\the\l@dpscR+1}%
2886 \ifcsdef{minpage@pstart@\@tmp}%
2887 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
2888 {\ifdimgreater{\pagetotal}{\ledthegoal}%
2889   {\@getnextboxtrue}%
2890   {\@getnextboxfalse}%
2891 }%
2892 {\@getnextboxtrue}%
2893 }{}
2894 \fi%
2895 \fi%
2896 \if@getnextbox%
2897 \ifnum\usenamecount{l@dmaxlinesinpar\the\l@dpscR}=\@donetotallinesR
2898 \parledgroup@notes@endR
2899 \fi
2900 \ifwrittenlinesR\else
2901 \writelinesinparR
2902 \writtenlinesRtrue
2903 \fi
2904 \ifnum\l@dnumpstartsR>\l@dpscR
2905 \writtenlinesRfalse
2906 \ifbypstart@R
2907 \global\line@numR=0%
2908 \resetprevline@%
2909 \fi
2910 \csuse{after@pendR@\the\l@dpscR}%
2911 \global\csundef{after@pendR@\the\l@dpscR}%
2912 \l@dcalcm@maxoftwo{\the\usenamecount{l@dmaxlinesinpar\the\l@dpscR}}%
2913   {\the\@donetotallinesR}%
2914   {\usenamecount{l@dmaxlinesinpar\the\l@dpscR}}%
2915 \global\@donetotallinesR \z@

```

```

2916 \global\advance\l@dpscR \@ne
2917 \global\pstartnumRtrue%
2918 \restore@pstartR@pc%
2919 \parledgroup@notes@endR
2920 \parledgroup@correction@notespacing@final{R}
2921 \else
2922 \print@last@after@pendRtrue%
2923 \fi
2924 \fi
2925 \fi}
2926
2927 %

```

XX Page numbering

XX.1 Global options

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

We cannot implement these two options by changing the value of the page counter, since its value is used by many \LaTeX 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 \LaTeX 's `\@outputpage` macro, as is the page counter in this macro. However, this increase will take account of the options.

`\par@patch@thepage` and `\par@patch@pagenumbering` 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 \LaTeX 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.

```

2928
2929 \newcommand{\par@patch@thepage}{%
2930 \ifboolexpr{%
2931   bool{sameparallelpagnumber}%
2932   or bool{prevpgnotnumbered}%
2933 }%
2934 {%
2935   \patchcmd{\thepage}%
2936     {page}{par@page}%

```

```

2937     {}%
2938     {\led@error@fail@patch@thepage}%
2939   }{}%
2940 }%
2941
2942 \newcommand{\par@patch@pagenumbering}{%
2943   \ifboolexpr{%
2944     bool{sameparallelnumber}%
2945     or bool{prevpgnotnumbered}%
2946   }%
2947   {%
2948     \setcounter{par@page}{1}%
2949   }%
2950   {}%
2951   \par@patch@thepage%
2952 }%
2953
2954 \ifl@dmemoir%
2955   \apptocmd{\@mempnum}%
2956     {\par@patch@pagenumbering}%
2957     {}%
2958     {\led@error@fail@patch@mempnum}%
2959
2960 \else%
2961   \apptocmd{\pagenumbering}%
2962     {\par@patch@pagenumbering}%
2963     {}%
2964     {\led@error@fail@patch@pagenumbering}%
2965 \fi%
2966
2967 \AtBeginDocument{\par@patch@thepage}%
2968 %

```

\@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.

```

2969 \AtBeginDocument{%
2970   \apptocmd{\@outputpage}{%
2971     \ifsameparallelnumber%
2972       \ifl@dprintingpages%
2973         \ifodd\c@page\else%
2974           \stepcounter{par@page}%
2975         \fi%
2976       \else%
2977         \stepcounter{par@page}%
2978       \fi%
2979     \else%
2980       \stepcounter{par@page}%
2981     \fi%

```

```

2982 }%
2983 {}%
2984 {\led@error@fail@patch@@outputpage}%
2985 }
2986 %

```

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

```

2987 \newcounter{par@page}%
2988 \setcounter{par@page}{1}%
2989 %

```

XX.2 mainmatter option of \Pages

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

```

\ifPages@mainmatter \newif\ifPages@mainmatter
\Pages@mainmatter \newcommand{\Pages@mainmatter}{%
2992   \ifPages@mainmatter%
2993   \pagenumbering{arabic}%
2994   \addtocounter{page}{1}%
2995   \addtocounter{par@page}{-1}%
2996   \patchcmd{\thepage}{page}{par@page}{}{}%
2997   \fi%
2998 }
2999 %

```

XXI 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 \pretocmd{\chapter}{%
3001   \ifl@dprintingpages%
3002   \vbox{}%
3003   \fi%
3004 }%
3005 {}%
3006 {}%
3007 %

```

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


```

3008 \newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}}
3009 \eledsectnotoc{R}
3010 %

```

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

```

3011 \newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}}
3012 \eledsectmark{L}
3013 %

```

\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.

```

3014 \newskip\eledsection@correcting@skip
3015 %

```

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

```

3016 \newwrite\eled@sectioningR@out
3017 %

```

XXII 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).

```

3018 \def\l@prev@pbR{}
3019 \def\l@prev@nopbR{}
3020 %

```

\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.

```

3021 \newcommand{\ledpbR}{\write\linenum@outR{\string\led@pbR}}
3022 \newcommand{\ledpbnumR}[1]{\write\linenum@outR{\string\led@pbnumR{#1}}}
3023 \newcommand{\lednopbR}{\write\linenum@outR{\string\led@nopbR}}
3024 \newcommand{\lednopbnumR}[1]{\write\linenum@outR{\string\led@nopbnumR{#1}}}
3025 %

```

```

3026 \newcommand{\led@pbR}{\listadd{\l@prev@pbR}{\the\absline@numR}}
3027 \newcommand{\led@pbnumR}[1]{\listadd{\l@prev@pbR}{\#1}}
3028 \newcommand{\led@nopbR}{\listadd{\l@prev@nopbR}{\the\absline@numR}}
3029 \newcommand{\led@nopbnumR}[1]{\listadd{\l@prev@nopbR}{\#1}}
3030 %

```

```
3031 \newmarks\parledgroup@
3032 \newmarks\parledgroup@series
3033 \newmarks\parledgroup@type
3034 %
```

```

3035 \newcommand{\parledgroup@notes@startL}{%
3036 \ifnum\usernamecount{10dmaxlinesinpar\the\10dpscl}>0%
3037 \IfStrEq{\splitfirstmarks\parledgroup@type}{\footnoteX}{\csuse{
bhooknoteX@splitfirstmarks\parledgroup@series}}{ }%
3038 \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
bhookXnote@splitfirstmarks\parledgroup@series}}{ }%
3039 \fi%
3040 \global\ledgroupnotesL@true%
3041 \insert@noterule@ledgroup{L}%
3042 }
3043 \newcommand{\parledgroup@notes@startR}{%
3044 \ifnum\usernamecount{10dmaxlinesinpar\the\10dpscl}>0%
3045 \IfStrEq{\splitfirstmarks\parledgroup@type}{\footnoteX}{\csuse{
bhooknoteX@splitfirstmarks\parledgroup@series}}{ }%
3046 \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
bhookXnote@splitfirstmarks\parledgroup@series}}{ }%
3047 \fi%
3048 \global\ledgroupnotesR@true%
3049 \insert@noterule@ledgroup{R}%
3050 }
3051 %

```

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

```

3052 \newcommand{\parledgroup@notes@endL}{%
3053   \global\ledgroupnotesL@false%
3054 }
3055 \newcommand{\parledgroup@notes@endR}{%
3056   \global\ledgroupnotesR@false%
3057 }
3058 %

```

`\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`

```

3059 \newcommand{\insert@noterule@ledgroup}[1]{
3060   \IfStrEq{\splitbotmarks\parledgroup@}{begin}{%
3061     \IfStrEq{\splitbotmarks\parledgroup@type}{Xfootnote}{
3062       \csuse{ifledgroupnotes#1@}
3063       \vskip\skip\csuse{mp\splitbotmarks\parledgroup@series footins}
3064       \csuse{\splitbotmarks\parledgroup@series footnoterule}
3065       \fi
3066     }
3067     {}
3068     \IfStrEq{\splitbotmarks\parledgroup@type}{footnoteX}{
3069       \csuse{ifledgroupnotes#1@}
3070       \vskip\skip\csuse{mpfootins\splitbotmarks\parledgroup@series}
3071       \csuse{footnoterule\splitbotmarks\parledgroup@series}
3072       \fi
3073     }{}
3074   }
3075   {}
3076 }
3077 %

```

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

```

3078 \newcommand{\setparledgroupnotespacing}[1]{\gdef\@parledgroupnotespacing
3079   {#1}}
3079 \newcommand{\@parledgroupnotespacing}{}
3080 %

```

`\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`.

```

3081 \dimdef{\parledgroup@notespacing@correction}{0pt}
3082 \newcommand{\parledgroup@notespacing@set@correction}{%
3083   {\@getfirstseries\csuse{Xnotefontsize@\@firstseries}}%We suppose all the
series has the same footnote size setup

```

```

3084 \@parledgroupnotespacing\dimgdef{\temp@spacing}{\baselineskip}}%
3085 \dimgdef{\parledgroup@notespacing@correction}{\baselineskip-\temp@spacing
}%
3086 }
3087 %

```

`\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.

```

3088 \newcommand{\parledgroup@correction@notespacing@init}{
3089 \dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}
3090 \dimdef{\parledgroup@notespacing@correction@modulo}{0pt}
3091 }
3092 \parledgroup@correction@notespacing@init
3093 %

```

`\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.

```

3094 \newcommand{\parledgroup@correction@notespacing@final}[1]{
3095 \ifparledgroup
3096 \vspace{\parledgroup@notespacing@correction@accumulated}
3097 \parledgroup@correction@notespacing@init%
3098 \ifstrequal{#1}{L}{
3099 \numdef{\@checking}{\the\l@dpscl-1}
3100 }{
3101 \numdef{\@checking}{\the\l@dpscr-1}
3102 }
3103 \dimdef{\@beforenotes@current@diff}{\csuse{\parledgroup@beforenotes@
3104 \@checking L}-\csuse{\parledgroup@beforenotes@\@checking R}}%
3105 \ifstrequal{#1}{L}%
3106 {% Left
3107 \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{-\
3108 \@beforenotes@current@diff}}%
3109 }%
3110 {% Right
3111 \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{\
3112 \@beforenotes@current@diff}}{}
3113 }%
3114 }
3115 \fi
3116 }
3117 %

```

`\parledgroup@correction@notespacing` `\parledgroup@correction@notespacing` is used before each printed line. If it is a line of notes in parallel ledgroup, the space `\parledgroup@notespacing@correction` is decreased, to make interline space correct. The decreased space is added to `\parledgroup@notespacing`

and `\parledgroup@notespacing@correction@modulo`. If `\parledgroup@notespacing@correction@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.

```

3114 \newcommand{\parledgroup@correction@notespacing}[1]{%
3115   \csuse{ifledgroupnotes#1@}%
3116   \vspace{-\parledgroup@notespacing@correction}%
3117   \dimdef{\parledgroup@notespacing@correction@accumulated}{\
parledgroup@notespacing@correction@accumulated+
parledgroup@notespacing@correction}%
3118   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo+
parledgroup@notespacing@correction}%
3119   \ifdimless{\parledgroup@notespacing@correction@modulo}{\baselineskip
}{\advance\numpagelinesL -\@ne%
3120   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo-\baselineskip}%
3121   }% mean greater than equal
3122   \fi%
3123 }
3124 %

```

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

```

3125 \dimdef\parledgroup@beforenotesL{0pt}
3126 \dimdef\parledgroup@beforenotesR{0pt}
3127 %

```

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

```

3128 \newcommand{\parledgroup@beforenotes@save}[1]{
3129   \ifparledgroup
3130   \csdimgdef{@parledgroup@beforenotes@\the\csuse{ldnumpstarts#1}#1}{\
csuse{parledgroup@beforenotes#1}}
3131   \csdimgdef{parledgroup@beforenotes#1}{0pt}
3132   \fi
3133 }
3134 %

```

XXIV Compatibility with eledmac

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

```
3135 \ifeledmaccompat@%  
3136  
3137  
3138   \unless\ifnocritical@  
3139     \let\onlyXside\Xonlyside  
3140   \fi  
3141 \fi  
3142 %
```

XXV 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. 323.

Appendix A.7 Migration to `reledpar` 2.6.0

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

Appendix A.8 Migration to `reledpar` 2.6.1

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

References

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

Index

Symbols

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

\@writelinesinparR	1
\@writelinesonpageL	1
\@writelinesonpageR	1
\@writepageofparL	1
\@writepageofparR	1
CLASSmemoir	110
COMMAND\+	47
COMMAND\@Rlineflag	74, 145
COMMAND\@adv	38, 144
COMMAND\@cs@linesinparL	101
COMMAND\@cs@linesonpageL	102
COMMAND\@eledsectionL	85
COMMAND\@eledsectionR	85
COMMAND\@eledsectmark	113
COMMAND\@eledsectnotoc	112
COMMAND\@footnotetext	76
COMMAND\@goalfraction	12, 105
COMMAND\@l@dtmncnta	69
COMMAND\@lab	36, 75, 144
COMMAND\@lopL	42, 96, 102
COMMAND\@lopR	42, 102
COMMAND\@mempnum	110
COMMAND\@namedef	79
COMMAND\@namuse	79
COMMAND\@nl	36, 43, 75, 144
COMMAND\@nl@regR	36
COMMAND\@outputpage	110, 111
COMMAND\@page	75
COMMAND\@par@sync@option	36
COMMAND\@parledgroupnotespacing	115
COMMAND\@pend	41
COMMAND\@pendR	41
COMMAND\@pstart	41
COMMAND\@pstartstrue	83
COMMAND\@ref	39–41, 43, 144
COMMAND\@ref@regR	40
COMMAND\@set	38, 144
COMMAND\@sw	40
COMMAND\AtBeginPairs	8, 49, 142
COMMAND\AtEveryPend	142, 144
COMMAND\AtEveryPstart	2, 17, 56, 86, 142, 144
COMMAND\AtEveryPstartCall	2, 17, 56, 143
COMMAND\AtEveryStanza	146
COMMAND\AtEveryStopStanza	146
COMMAND\Clear the right lines for \@read@linelist	36
COMMAND\Columns	7, 8, 17, 48, 85, 89, 139–141, 144, 145
COMMAND\Columns@print@after@pend	92
COMMAND\Columns@print@before@pstart	92
COMMAND\Lcolwidth	8, 10, 94
COMMAND\Leftsidehook	139

COMMAND\Leftsidehookend	139
COMMAND\MakePerPage	13
COMMAND\Pages	4, 7, 10, 12, 17, 47, 48, 66, 70, 71, 93, 94, 99, 110, 112, 115, 139, 141, 143–145
COMMAND\Pages@mainmatter	112
COMMAND\Pairs	47
COMMAND\Rcolwidth	8, 10, 94
COMMAND\Rightsidehook	139
COMMAND\Rightsidehookend	139
COMMAND\Rlineflag	119
COMMAND\Xendlineflag	14, 145
COMMAND\Xlineflag	14, 120, 145
COMMAND\Xmaxhnotes	13
COMMAND\Xnoteswidthliketwocolumns	9, 142
COMMAND\Xonlyside	14, 70, 120
COMMAND\&	18
COMMAND\absline@numR	35
COMMAND\add@penalties	69
COMMAND\add@penaltiesL	69
COMMAND\advanceline	38, 43, 144
COMMAND\affixline@num	64
COMMAND\affixline@numR	64, 139, 140
COMMAND\affixpstart@num	66
COMMAND\affixpstart@numR	66
COMMAND\affixside@note	75
COMMAND\aftercolumnseparator	9, 89, 142
COMMAND\araw@textfalse	83
COMMAND\araw@texttrue	83
COMMAND\at@begin@pairs	49
COMMAND\autopar	17
COMMAND\ballast@count	69
COMMAND\baselineskip	88, 117
COMMAND\bbl@set@language	80, 81, 144
COMMAND\beforecolumnseparator	9, 89, 142
COMMAND\begin	18
COMMAND\beginnumbering	15–17, 26, 36, 89, 140, 142, 144
COMMAND\beginnumberingR	31, 43
COMMAND\bf	140
COMMAND\bfseries	140
COMMAND\brokenpenalty	69
COMMAND\chapter	112, 139
COMMAND\check@goal	105
COMMAND\check@pstarts	83
COMMAND\checkpageL	96, 103
COMMAND\checkpb@columns	88
COMMAND\checkpbL	104
COMMAND\checkpbR	104
COMMAND\checkraw@text	83
COMMAND\checkverseL	105
COMMAND\checkverseR	105
COMMAND\clear(double)page	100

COMMAND\clearl@dleftpage	101
COMMAND\clearl@drighpage	101
COMMAND\clearpage	101, 143
COMMAND\cleartoevenpage	100
COMMAND\cleartol@devenpage	100, 112
COMMAND\columnrulewidth	8, 88
COMMAND\columns@position	89
COMMAND\columnseparator	8, 9
COMMAND\columnspan	9, 142
COMMAND\correct@Xfootins@box	71, 73, 143
COMMAND\correct@footinsX@box	71, 143
COMMAND\critext	143
COMMAND\csname	48
COMMAND\displaywidowpenalty	69
COMMAND\do@actions	62
COMMAND\do@actions@fixedcode	139
COMMAND\do@actions@nextR	62
COMMAND\do@actionsR	62, 139
COMMAND\do@ballast	69
COMMAND\do@ballastR	62
COMMAND\do@insidelineLhook	141
COMMAND\do@insidelineRhook	141
COMMAND\do@line	57
COMMAND\do@line(L/R)	60
COMMAND\do@lineL	57, 69, 139, 140
COMMAND\do@lineLhook	139
COMMAND\do@lineR	60, 139, 140, 142
COMMAND\do@lineRhook	139
COMMAND\do@lockoff	144
COMMAND\do@lockoffR	39
COMMAND\do@lockon	144
COMMAND\do@lockonR	38
COMMAND\doinsidelineLhook	142
COMMAND\doinsidelineRhook	142
COMMAND\dolineLhook	142
COMMAND\dolineRhook	142
COMMAND\edindex	143
COMMAND\edlabel	140, 143
COMMAND\edtext	39, 43, 44, 142, 143
COMMAND\eled@sectioningR@out	113
COMMAND\eledchapter	143
COMMAND\eledsection	142, 143, 145
COMMAND\eledsection@correcting@skip	113
COMMAND\eledsectmark	20, 113
COMMAND\eledsectnotoc	20, 112
COMMAND\eledxxx	142
COMMAND\end	18
COMMAND\endgraf	55
COMMAND\endlock	44, 144
COMMAND\endnumbering	9, 15, 17, 27, 144

COMMAND\endsub	43, 144
COMMAND\endumbering	15
COMMAND\expandafter	46
COMMAND\extensionchars	26
COMMAND\firstlinenum	16, 141, 144, 145
COMMAND\firstsublinenum	141, 144, 145
COMMAND\fix@page	37, 144
COMMAND\flag@end	43, 139, 142
COMMAND\flag@start	43, 142
COMMAND\flush@notesR	70
COMMAND\footnote	48
COMMAND\footnoteX	45, 46
COMMAND\footnoteX@reading	146
COMMAND\footnoteXmk	14
COMMAND\footnoteXnomk	14, 45
COMMAND\frontmatter	12, 20
COMMAND\get@nextboxL	140
COMMAND\get@nextboxR	140
COMMAND\getline@numL	61
COMMAND\getline@numR	61
COMMAND\getlinesfrompagelistL	101
COMMAND\getlinesfrompagelistR	102
COMMAND\getlinesfromparlistL	101
COMMAND\getlinesfromparlistR	101
COMMAND\gl@p	46
COMMAND\goalfraction	119
COMMAND\hangingsymbol	119, 140
COMMAND\hfill	88, 89
COMMAND\hidenumbering	16, 144
COMMAND\if@getnextbox	106
COMMAND\ifPages@mainmatter	112
COMMAND\ifbypage@	145
COMMAND\ifbypstart@R	145
COMMAND\ifdim	88
COMMAND\ifinserthangingsymbol	76
COMMAND\ifinserthangingsymbolR	76
COMMAND\ifl@dpagfull	103
COMMAND\ifl@dpaging	24, 142
COMMAND\ifl@dpairing	24, 139
COMMAND\ifl@dsamelang	142
COMMAND\ifl@dsamepage	103
COMMAND\ifl@pagfull	96
COMMAND\ifledRcol	24
COMMAND\iflinenumberLevenifblank	34
COMMAND\iflinenumberRevenifblank	34
COMMAND\iflledRcol	140
COMMAND\ifnumberedpar@	52
COMMAND\ifnumberingR	140
COMMAND\ifnumberpstart	48
COMMAND\ifpst@rtedL	26, 27, 53, 139

COMMAND\ifpst@rtedR	26
COMMAND\ifsublines@	38
COMMAND\insert@countR	39
COMMAND\insert@noterule@ledgroup	115
COMMAND\insertlines@list	39
COMMAND\insertlines@listR	40
COMMAND\inserts@list	52
COMMAND\inserts@listR	68
COMMAND\l@d@nums	44
COMMAND\l@d@set	38, 44, 144
COMMAND\l@dLcolrawbox	52
COMMAND\l@dLcolrawbox1	82
COMMAND\l@dLcolrawbox2	82
COMMAND\l@dRcolrawbox	52
COMMAND\l@dbfnote	76, 144
COMMAND\l@dcalc@maxoftwo	102
COMMAND\l@dcalc@minoftwo	102
COMMAND\l@dchecklang	139, 141
COMMAND\l@dcsnote	142
COMMAND\l@dleftbox	57, 96, 143
COMMAND\l@dlinenumR	34, 139
COMMAND\l@dlsnote	142
COMMAND\l@dmake@labels	74
COMMAND\l@dmaxlinesinpar	94
COMMAND\l@dmaxlinesinpar1	82
COMMAND\l@dminpagelines	95, 139
COMMAND\l@dnumpstartsL	82, 139
COMMAND\l@dprintingcolumnstrue	143
COMMAND\l@dprintingpagestrue	143
COMMAND\l@dpscL	85, 94, 95, 107
COMMAND\l@dpscR	85, 94, 95
COMMAND\l@drsnote	142
COMMAND\l@dsetupmaxlinecounts	82
COMMAND\l@duselanguage	80, 81, 139
COMMAND\l@dzeromaxlinecounts	82
COMMAND\l@prev@nopbR	113
COMMAND\l@prev@pbR	113
COMMAND\labelpstarttrue	140
COMMAND\labelref@list	75
COMMAND\labelref@listR	74
COMMAND\lang	80
COMMAND\last@page@numR	37
COMMAND\led	140
COMMAND\led@nopbR	113, 114
COMMAND\led@nopbnumR	113, 114
COMMAND\led@pbR	113, 114
COMMAND\led@pbnumR	113, 114
COMMAND\ledinnerrote	19
COMMAND\ledleftnote	19
COMMAND\lednoph	18, 88, 104

COMMAND\lednopbR	113
COMMAND\lednopbnumR	113
COMMAND\ledouterote	19
COMMAND\ledpb	88, 104
COMMAND\ledpbR	113
COMMAND\ledpbnumR	113
COMMAND\ledrightnote	19
COMMAND\ledsidenote	19
COMMAND\ledstrutL	139
COMMAND\ledstrutR	139, 145
COMMAND\ledthegoal	105
COMMAND\ledtrutL	139, 145
COMMAND\leftlinenumR	34, 139
COMMAND\let	46
COMMAND\line@list@R	40
COMMAND\line@list@stuff	36, 43
COMMAND\line@margin	31
COMMAND\line@marginR	31, 139
COMMAND\line@numR	35
COMMAND\lineation	16, 143
COMMAND\lineation*	16, 31, 142
COMMAND\lineationR	16, 30, 143
COMMAND\linenum@out	75
COMMAND\linenum@outR	42
COMMAND\linenumberLevenifblanktrue	17, 146
COMMAND\linenumberRevenifblank	146
COMMAND\linenumberRevenifblanktrue	17
COMMAND\linenumberstyle	16, 120
COMMAND\linenumberstyle*	120
COMMAND\linenumberstyleR	16, 120
COMMAND\linenumincrement	16, 141, 144, 145
COMMAND\linenummargin	16, 31, 139, 144–146
COMMAND\linenummargin*	16, 31, 145
COMMAND\linenummarginR	16, 31, 145
COMMAND\linenumrepR	33, 139
COMMAND\linesinpar@listL	41, 101
COMMAND\linesonpage@listL	42, 101
COMMAND\lock@off	39
COMMAND\lock@on	38
COMMAND\mainmatter	2, 12, 20, 145
COMMAND\makeatletter	59
COMMAND\maxchunks	7, 18, 82, 83
COMMAND\maxhnotesX	13
COMMAND\memorydump	15, 29
COMMAND\n@num	143
COMMAND\new@lineL	43
COMMAND\new@lineR	43
COMMAND\newhookcommand@series	47
COMMAND\newif	143
COMMAND\newpage	100, 101, 143

COMMAND\newseries	48
COMMAND\newseries@	44
COMMAND\newseries@par	44, 47, 48
COMMAND\noledxxx	142
COMMAND\nomark@	45
COMMAND\nomaxlines	41
COMMAND\normalbfnoteX	139, 144
COMMAND\notesXwidthliketwocolumns	9, 142
COMMAND\num@lines	69
COMMAND\num@lines(R)	52
COMMAND\numberingR	28
COMMAND\numberlinefalse	7
COMMAND\numberonlyfirstinline	140
COMMAND\numberpstartfalse	16
COMMAND\numberpstarttrue	16, 140, 145
COMMAND\one@line	52, 76
COMMAND\one@lineR	52
COMMAND\onlyXside	120
COMMAND\onlysideX	14, 70, 144
COMMAND\otherlanguage	144
COMMAND\page@action	38, 144
COMMAND\pagenumbering	110, 145
COMMAND\pages	12
COMMAND\pagetotal	96, 143
COMMAND\par@line	69
COMMAND\par@line(R)	52
COMMAND\par@patch@pagenumbering	110
COMMAND\par@patch@thepage	110
COMMAND\par@sync@option	22
COMMAND\parledgroup@	114
COMMAND\parledgroup@beforenotes@save	117
COMMAND\parledgroup@beforenotesL	117
COMMAND\parledgroup@beforenotesR	117
COMMAND\parledgroup@correction@notespacing	116
COMMAND\parledgroup@correction@notespacing@final	116
COMMAND\parledgroup@correction@notespacing@init	116
COMMAND\parledgroup@notes@endL	115
COMMAND\parledgroup@notes@endR	115
COMMAND\parledgroup@notes@startL	114
COMMAND\parledgroup@notes@startR	114
COMMAND\parledgroup@notespacing@correction	115, 116
COMMAND\parledgroup@notespacing@correction@accumulated	116
COMMAND\parledgroup@notespacing@correction@modulo	117
COMMAND\parledgroup@notespacing@set@correction	115
COMMAND\parledgroup@series	114
COMMAND\parledgroup@type	114
COMMAND\parledgroupnotespacing	119
COMMAND\parledgrouptrue	19
COMMAND\patchcmd	145
COMMAND\pausenumbering	9, 29, 146

COMMAND\pend	3, 7, 10, 17–19, 48, 52, 55, 56, 81, 92, 142, 144, 145
COMMAND\pendL	141
COMMAND\pendR	141
COMMAND\pends	17
COMMAND\perpage	13
COMMAND\prev@nopbR	114
COMMAND\prev@pbR	114
COMMAND\prevpgstyle	23
COMMAND\print@Xnotes	70
COMMAND\print@Xnotes@forpages	70, 143
COMMAND\print@columnseparator	88, 141
COMMAND\print@eledsectionL	59
COMMAND\print@line	58
COMMAND\print@lineL	58
COMMAND\print@notesX@forpages	143
COMMAND\printlinenumR	120
COMMAND\printlinesR	145
COMMAND\pstart	3, 7, 10, 16–19, 30, 44, 48, 52, 53, 55, 56, 81, 86, 92, 140, 142, 144, 145
COMMAND\pstartL	56, 141
COMMAND\pstartR	56, 140, 141
COMMAND\pstartinfootnote	143
COMMAND\raw@text	81
COMMAND\read@linelist	36, 122, 144
COMMAND\ref@reg	39
COMMAND\ref@regR	39, 144
COMMAND\relax	120
COMMAND\reledmac	144
COMMAND\renewcommand	119
COMMAND\resumenumbering	9, 29, 142, 146
COMMAND\resumenumberingR	31, 142
COMMAND\rightlinenumR	34, 139
COMMAND\section	139
COMMAND\section@num	26
COMMAND\selectlanguage	17, 80, 81
COMMAND\set@continuousnumberingforR	31
COMMAND\set@line	44, 144
COMMAND\set@line@action	38, 144
COMMAND\setRlineflag	16, 119
COMMAND\setgoalfraction	12, 119
COMMAND\sethangingsymbol	18
COMMAND\setline	38, 44, 144
COMMAND\setlinenum	38, 44, 144
COMMAND\setnoteposition...	89
COMMAND\setparledgroupnotespacing	119, 145
COMMAND\setposition...	89
COMMAND\setstanzaindents	9, 18, 119
COMMAND\setwidth...	89
COMMAND\sidenotemargin	19, 141
COMMAND\sidenotemargin*	19, 142
COMMAND\skipnumbering	16, 143

COMMAND\sloppy	8
COMMAND\stanza	7, 9, 16, 18, 50, 77, 120, 140
COMMAND\stanzanumtrue	18
COMMAND\startlock	44, 144
COMMAND\startsub	43, 144
COMMAND\sub@action	38, 144
COMMAND\sub@off	75
COMMAND\sub@on	75
COMMAND\subline@numR	35
COMMAND\sublinenumberstyle	16, 120
COMMAND\sublinenumberstyle*	120
COMMAND\sublinenumberstyleR	16, 120
COMMAND\sublinenumincrement	141, 144, 145
COMMAND\sublinenumrepR	33, 139
COMMAND\syntaxonly	146
COMMAND\sza@0@	18
COMMAND\textheight	13
COMMAND\textwidth	50
COMMAND\the@labelX	145
COMMAND\thefootnoteX	141
COMMAND\theledlanguageL	80, 81
COMMAND\theledlanguageR	80, 81
COMMAND\thepage	20, 110
COMMAND\thepstartL	16, 140
COMMAND\thepstartR	16, 140
COMMAND\thestanzaL	18
COMMAND\thestanzaR	18
COMMAND\vbox	53
COMMAND\vl@dbfnote	76
COMMAND\vskip	115
COMMAND\vsplit	69
COMMAND\widthliketwocolumns	9
COMMAND\widthliketwocolumnsfalse	9
COMMAND\widthliketwocolumnstrue	9
COMMAND\xflagref	145
COMMAND\xright@appenditem	46
COMMAND\xspace	21
COMMAND\xxxfootstart	89
COMMAND\xxxmatter	110
ENVIRONMENTLeftside	50
ENVIRONMENTRightside	51
ENVIRONMENTastanza	18, 77, 78, 120, 145
ENVIRONMENTcolumns	28, 146
ENVIRONMENTledgroup	6, 146
ENVIRONMENTleft	16
ENVIRONMENTpages	28, 49, 145, 146
ENVIRONMENTpairs	49, 145
PACKAGEEDMAC	121
PACKAGEEDSTANZA	121
PACKAGEEledmac	44, 143

PACKAGEEledpar	143
PACKAGETABMAC	121
PACKAGEbabel	17, 79–81, 146
PACKAGEedmac	121
PACKAGEeledmac	4, 82, 118, 119, 121, 141, 142, 144, 145
PACKAGEeledpar	5, 6, 13, 32, 119, 120, 141–143
PACKAGEetoolbox	88, 113
PACKAGEledmac	6
PACKAGEledpar	1, 6
PACKAGEMemoir	121
PACKAGEMusixtex	142
PACKAGEperpage	2, 13, 120, 146
PACKAGEpolyglossia	17, 79–81, 146
PACKAGEreledmac	1, 3, 5–7, 9, 13, 16–21, 23, 24, 26, 27, 31, 32, 35–39, 41–45, 47, 48, 58, 66, 74–76, 94, 99, 119, 120, 144–146
PACKAGEreledpar	1, 3, 5–7, 9–12, 18–23, 30, 35, 41, 42, 44, 47, 48, 74, 119, 120, 145
PACKAGEsetspace	2, 20
PACKAGESyntonly	146
PACKAGEXkeyval	21
PACKAGEX	146

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)	18
\AtBeginPairs	1, 8
\AtEveryPstartCall	1
\autopar	17

B

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

C

\c@firstlinenumR	1
------------------	---

<code>\c@firstsublinenumR</code>	1
<code>\c@linenumincrementR</code>	1
<code>\c@sublinenumincrementR</code>	1
<code>\ch@ck@l@ckR</code>	1
<code>\ch@cksub@l@ckR</code>	1
<code>\chapter</code>	1
<code>\chapterinpages</code>	1
<code>\check@goal</code>	1
<code>\check@pstarts</code>	1
<code>\checkpageL</code>	1
<code>\checkpageR</code>	1
<code>\checkpb@columns</code>	1
<code>\checkpbL</code>	1
<code>\checkpbR</code>	1
<code>\checkraw@text</code>	1
<code>\checkverseL</code>	1
<code>\checkverseR</code>	1
<code>\clearl@dleftpage</code>	1
<code>\clearl@drightpage</code>	1
<code>\cleartoevenpage</code>	1
<code>\cleartol@devenpage</code>	1
<code>\columnrulewidth</code>	1, 8
<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, 8
<code>\columnspan</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

\doinsidelineRhook	1
\dolineLhook	1
\dolineRhook	1
\dump@pstartL@pc	1
\dump@pstartR@pc	1

E

\edlabel	1
\edtext	1
\eled@sectioningR@out	1
\eledsection@correcting@skip	1
\eledsectmark	1, 20
\eledsectnotoc	1, 20
\endlock	1
\endnumbering	1, 15
\endnumberingR	1
\endsub	1
environments:	
astanza	18
Leftside	15
pages	9
pairs	8
Rightside	15

F

\f@x@l@cksR	1
\finish@Pages@notes	1
\first@linenum@out@Rfalse	1
\first@linenum@out@Rtrue	1
\firstlinenum	1, 16
\firstlinenum*	1, 16
\firstlinenumR	1, 16
\firstsublinenum	1, 16
\firstsublinenum*	1, 16
\firstsublinenumR	1, 16
\fix@page	1
\flag@end	1
\flag@start	1
\flush@notesR	1
\footnote@reading	1
\footnote@typeset	1
\footnoteXmk	14
\footnoteXnomk	14

G

\get@familiarfootnote@number	1
\get@nextboxL	1
\get@nextboxR	1
\getline@numR	1
\getlinesfrompagelistL	1

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

H

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

I

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

L

<code>\l@d@set</code>	1
<code>\l@dbfnote</code>	1
<code>\l@dc@maxchunks</code>	1
<code>\l@dcalc@maxoftwo</code>	1

<code>\l@dcalc@minoftwo</code>	1
<code>\l@dcalcnun</code>	1
<code>\l@dchecklang</code>	1
<code>\l@dleftbox</code>	1
<code>\l@dlinenumR</code>	1
<code>\l@dmake@labelsR</code>	1
<code>\l@dminpagelines</code>	1
<code>\l@dnumpstartsL</code>	1
<code>\l@dnumpstartsR</code>	1
<code>\l@dpagefullfalse</code>	1
<code>\l@dpagefulltrue</code>	1
<code>\l@drightbox</code>	1
<code>\l@dsamepagefalse</code>	1
<code>\l@dsamepagetrue</code>	1
<code>\l@dsetupmaxlinecounts</code>	1
<code>\l@dsetuprawboxes</code>	1
<code>\l@dskipversenumberR</code>	1
<code>\l@dusedbabelfalse</code>	1
<code>\l@dusedbabeltrue</code>	1
<code>\l@duselanguage</code>	1
<code>\l@dzeromaxlinecounts</code>	1
<code>\l@pscL</code>	1
<code>\l@pscR</code>	1
<code>\labelref@listR</code>	1
<code>\last@page@numR</code>	1
<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@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@RightOnLeftPage</code>	1
<code>\led@err@Rightside@PreviousNotPrinted</code>	1
<code>\led@err@TooManyPstarts</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@nopbnumR</code>	1
<code>\led@nopbR</code>	1
<code>\led@pbnumR</code>	1
<code>\led@pbR</code>	1
<code>\led@warn@ChangeSyncOption</code>	1
<code>\led@warn@setting@in@rightside</code>	1
<code>\lednopbnum</code>	1
<code>\lednopbnumR</code>	1
<code>\ledpbnumR</code>	1

<code>\ledpbR</code>	1
<code>\ledstrutL</code>	1
<code>\ledstrutR</code>	1
<code>\ledthegoal</code>	1
<code>\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, 16
<code>\lineationR</code>	1, 16
<code>\linenum@outR</code>	1
<code>\linenumberLevenifblanktrue</code>	16
<code>\linenumberRevenifblanktrue</code>	16
<code>\linenumberstyle*</code>	1, 16
<code>\linenumberstyleR</code>	1, 16
<code>\linenumincrement</code>	1, 16
<code>\linenumincrement*</code>	1, 16
<code>\linenumincrementR</code>	1, 16
<code>\linenummargin</code>	1
<code>\linenummargin*</code>	1, 16
<code>\linenummarginR</code>	1, 16
<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>	15
<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>\normalbfnoteX</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>	10, 11
<code>optioncontinuousnumberingwithcolumns</code>	9, 146
<code>optionnomaxlines</code>	10, 11, 22
<code>optionnosyncpstarts</code>	12, 22, 106
<code>optionshiftedpstarts</code>	6, 11, 22
<code>optionwidthliketwocolumns</code>	9

P

<code>\page@action</code>	1
<code>\page@numR</code>	1
<code>\Pages</code>	1, 10
<code>pages (environment)</code>	9
<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
<code>\pausenumberingR</code>	1
<code>\pend</code>	17
<code>\pendL</code>	1
<code>\pendR</code>	1
<code>\prev@nopbR</code>	1
<code>\prev@pbR</code>	1
<code>\prevpgstyle</code>	1
<code>\print@columnseparator</code>	1
<code>\print@eledsectionL</code>	1

\print@eledsectionR	1
\print@lineL	1
\print@lineR	1
\print@notesX@forpages	1
\print@Xnotes@forpages	1
\pstart	17
\pstartL	1
\pstartR	1

R

\Rcolwidth	1, 8, 10
\read@linelist	1
\reledpar@error	1
\reledpar@warning	1
\restore@pstartL@pc	1
\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	18
\setline	1
\setlinenum	1
\setnamebox	1
\setnotepositionliketwocolumns@C	1
\setnotepositionliketwocolumns@L	1
\setnotepositionliketwocolumns@R	1
\setpositionliketwocolumns@C	1
\setpositionliketwocolumns@L	1
\setpositionliketwocolumns@R	1
\setRlineflag	16
\setwidthliketwocolumns@C	1
\setwidthliketwocolumns@L	1
\setwidthliketwocolumns@R	1
\sidenote@marginR	1

<code>\sidenotemargin*</code>	<u>1</u>
<code>\skip@lockoff</code>	<u>1</u>
<code>\skipnumbering</code>	<u>1</u> , 16
<code>\startlock</code>	<u>1</u>
<code>\startsub</code>	<u>1</u>
<code>\sub@action</code>	<u>1</u>
<code>\subline@numR</code>	<u>1</u>
<code>\sublinenumberstyle*</code>	<u>1</u> , 16
<code>\sublinenumberstyleR</code>	<u>1</u> , 16
<code>\sublinenumincrement</code>	<u>1</u> , 16
<code>\sublinenumincrement*</code>	<u>1</u> , 16
<code>\sublinenumincrementR</code>	<u>1</u> , 16
<code>\sublinenumrepR</code>	<u>1</u>

T

<code>\theledlanguageL</code>	<u>1</u>
<code>\theledlanguageR</code>	<u>1</u>
<code>\thepar@page</code>	<u>1</u>
<code>\thepstartL</code>	16
<code>\thepstartR</code>	16
<code>\thestanzaL</code>	<u>1</u> , 18
<code>\thestanzaR</code>	<u>1</u> , 18

U

<code>\unhnamebox</code>	<u>1</u>
<code>\unvnamebox</code>	<u>1</u>
<code>\usernamecount</code>	<u>1</u>

W

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

X

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

Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
\Columns: Added \l@dchecklang and \l@duselanguage to \Columns	86
\Pages: Added \l@duselanguage to \Pages	95
General: Added section of babel related code	79
Fix babel problems	1
v0.3.0.	
\Pages: Added \ledstrutL to \Pages	95
Added \ledstrutR to \Pages	97
\Rightsidehookend: Added \Leftsidehook, \Leftsidehookend, \Rightsidehook and \Rightsidehookend	51
\affixline@numR: Changed \affixline@numR to match neweledmac	64
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR	62
\do@lineL: Added \do@lineLhook to \do@lineL	57
Simplified \do@lineL by using macros for some common code	57
\do@lineR: Changed \do@lineR similarly to \do@lineL	60
\flag@end: Removed extraneous spaces from \flag@end	43
\ifledRcol: Moved \ifl@dpairing toeledmac	24
\ifpst@rtedR: Moved \ifpst@rtedL toeledmac	26
\l@dlinenumR: Simplified \leftlinenumR and \rightlinenumR by introducing \l@dlinenumR	34
\l@dnumpstartsR: Moved \l@dnumpstartsL toeledmac	82
\ledstrutR: Added \ledtrutL and \ledstrutR	100
\normalbfnoteX: Removed extraneous spaces from \normalbfnoteX	76
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR	33
General: Added \do@lineLhook and \do@lineRhook	60
Added hooks into Leftside environment	50
Reorganize for ledarab	1
v0.3.a.	
\line@marginR: Do not just set \line@marginR in \linenummargin	31
General: Minor \linenummargin fix	1
v0.3.b.	
\Pages: Added \l@dminpagelines calculation for succeeding page pairs	98
General: Improved parallel page balancing	1
v0.3.c.	
General: Compatibilty with Polyglossia	1
v0.4.0.	
General: No more ledparpatch. All patches are now in the main file.	1
v0.5.0.	
General: Corrections about \section and other titles in numbered sections	1
v0.6.0.	
General: Be able to use \chapter in parallel pages.	1
v0.7.0.	
General: Option ‘shiftedverses’ which make there is no blank between two parallel verses with unequal length.	1

v0.8.0.	
General: Possibility to have a symbol on each hanging of verses, like in the french typography. Redefine the commande <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>	24
General: Possibility to number <code>\pstart</code>	16
Possibilty 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>	87
<code>\affixline@numR</code> : Changed <code>\affixline@numR</code> to allow to disable line numbering (like in <code>eledmac 0.15</code>).	64
<code>\get@nextboxR</code> : Change <code>\get@nextboxL</code> and <code>\get@nextboxR</code> to allow to disable line numbering (like in <code>eledmac 0.15</code>).	106
<code>Pstart</code> number can be printed in side	107
<code>\inserthangingsymbolR</code> : Prevent the column separator for hanging verse from shifting	76
General: Change <code>\do@lineL</code> and <code>\do@lineR</code> to allow line numbering by <code>pstart</code> (like in <code>eledmac 0.15</code>).	57
Lineation can be by <code>pstart</code> (like in <code>eledmac 0.15</code>).	30
New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . .	76
v0.12.0.	
General: New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	76
v1.0.0.	
General: Compatibility with <code>eledmac</code> . Change name to <code>eledpar</code>	1
Debug in lineation by <code>pstart</code>	30
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>).	52
General: <code>Shiftedverses</code> becomes <code>shiftedpstarts</code>	1
v1.1.1.	
<code>\pstartR</code> : Correct <code>\pstartR</code> bug introduced by 1.1.	52
v1.1.2.	
<code>\affixside@noteR</code> : Remove spurious space between line number and line content . .	75
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.	28

v1.3.0.	
\edtext: Manage RTL language.	44
v1.3.1.	
\l@dbfnote: Compatibility of standard footnotes with eledmac when theses footnotes contain any commands.	76
v1.3.2.	
General: Debug with some classes.	1
v1.3.3.	
\l@dbfnote: Spurious space with footnote in right column.	76
General: Debugging the left notes of the right column.	75
v1.3.4.	
General: Allow use of commands in sidenotes, as introduced by eledmac 1.0.	75
v1.3.5.	
\normalbfnoteX: Allows one to redefine \thefootnoteX with alpha when some packages are loaded.	76
v1.4.0.	
General: Added \do@insidelineLhook and \do@insidelineRhook	60
v1.4.1.	
\normalbfnoteX: Fix bug with normal familiar footnotes when mixing RTL and LTR text.	76
General: Enable the use of stanzaindent repetition within stanza environment.	77
v1.4.3.	
\inserthangingsymbolR: Hanging verse is no longer automatically flush right.	76
\pendL: Spurious spaces in \pendL.	55
\pendR: Spurious spaces in \pstartR.	56
\pstartR: Spurious spaces in \pstartL and \pstartR.	52
General: Corrects a false hanging verse when a verse is exactly the length of a line.	1
v1.5.0.	
\sublinenumincrement*: Add starred version of \firstlinenum, \linenumincrement, \firstsublinenum, \sublinenumincrement to change both Left and Rightside.	32
General: Add, as in eledmac, features to manage page breaks.	1
v1.6.0.	
General: Add tool and documentation for parallel ledgroups	19
v1.7.0.	
General: Add, as in eledmac, features to make crossrefs with pstart numbers.	1
v1.8.0.	
\Columns: Modify \Columns to enable to add section's title.	85
Suppress \l@dchecklang from \Columns.	86
\Pages: Modify \Pages to enable to add section's title.	93
\l@dchecklang: Suppress \l@dchecklang which did not work and was not logical, because both columns could have the same language but not the main language of the document.	80
\pendL: As in eledmac, \pendL can have an optional argument.	55
\pendR: As in eledmac, \pendR can have an optional argument.	56
\print@columnseparator: Move some code of \Columns to \print@columnseparator.	88
\pstartR: As in eledmac, \pendL and \pendR can have an optional argument.	52
\sidenotemargin*: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.	75

Add <code>\sidenotemargin*</code>	75
<code>\theledlanguageR</code> : Correct left/right language setting with polyglossia.	81
General: <code>\beginnumbering</code> is defined only on <code>eledmac</code> , not on <code>eledpar</code>	26
<code>\l@disnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in <code>eledmac</code>	75
Add <code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code>	9
Add <code>\columnspanposition</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.	57
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>	20
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.	80
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) (?).	57
<code>\do@lineR</code> : Fix a bug with critical notes at the beginning of a page, added by v1.8.0 (?).	60
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>	43
<code>\lineation*</code> : Add <code>\lineation*</code>	31
<code>\reledpar@error</code> : Errors specific to <code>eledpar</code> send to <code>eledpar</code> handbook	24
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>	93
<code>\doinsidelineRhook</code> : Added <code>\dolineLhook</code> , <code>\dolineRhook</code> , <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code>	59
<code>\resumenumberingR</code> : Debug <code>\resumenumberingR</code>	29
General: Add <code>\noeledxxx</code> , as in <code>eledmac</code>	1
v1.9.0.	
<code>\ifwidthliketwocolumns</code> : Added <code>widthliketwocolumns</code> option	23
<code>\theledlanguageR</code> : Debug left/right language switching with polyglossia. Do not write in .aux file when setting left/right lines.	81
General: Add <code>\AtBeginPairs</code> macro.	8
Compatibility with <code>\Xnoteswidthliketwocolumns</code> and <code>\notesXwidthliketwocolumns</code>	1
v1.9.1.	
<code>\ifledRcol</code> : Moved <code>\ifl@dpaging</code> to <code>eledmac</code>	24
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).	93
Debug wrong parallel pages synchronization when an <code>\edtext</code> falls across two pages.	93
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.	
\line@list@stuffR: Revert modification of 1.4.2, which makes bugs with numbering.	
Leave vertical mode to solve spurious space before minipage.	43
v1.11.0.	
\edtext: \critext and \edtext are now defined only in eledmac.	44
General: Compatibility of standard footnotes with some biblatex styles.	1
v1.12.0.	
\Columns: Add \l@dprintingcolumnstrue	85
\Pages: Add \l@dprintingpagestrue	93
\edlabel: \edlabel and \edindex works now with hyperref when using eledpar.	74
\edlabel is now defined only one time for both eledmac and eledpar	74
\print@eledsectionL: Compatibility with Lua \TeX RTL languages.	59
\print@eledsectionR: Compatibility with Lua \TeX RTL languages.	61
\print@lineL: Compatibility with Lua \TeX RTL languages.	58
General: Compatibility with Lua \TeX RTL languages.	1
v1.12.1.	
\print@eledsectionL: Fixes bug with Lua \TeX RTL \eledsection.	59
v1.13.0.	
\Pages: Prevent false overfull hboxes when using \Pages outside of pages environment.	94
When using shiftedpstarts option, a \l@dleftbox with a null height will advance the	
\pagetotal in any case.	93
\clearl@drightpage: Use \newpage instead of \clearpage.	101
\ifledRcol: Remove false boolean settings which are not needed.	24
General: Enable the use of optional argument of & in astanza environment.	77
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)	93
\correct@footinsX@box: Call \correct@footinsX@box and	
\correct@Xfootins@box directly in \print@notesX@forpages and	
\print@Xnotes@forpages.	70
Correct \correct@footinsX@box and \correct@Xfootins@box	70
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 action 1008 and 1009	62
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting	
when a verse is hanging	76
\lineationR: As \lineation, \lineationR automatically set the	
\pstartinfootnote.	30
\n@num: \n@num defined only one time for both Eledmac and Eledpar.	39
\skipnumbering: \skipnumbering defined only one time for both Eledmac and	
Eledpar	44
General: Add \AtEveryPstartCall.	1
Add sameparallepagenumber 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.	44
General: Error message when calling \Pages inside 'pages' environment and \Columns inside 'pairs' environment.	1
Error message when starting a Leftside/a Rightside while the previous one has not been yet typeset.	1
Error message when using \beginnumbering..\endnumbering without \pstart. . .	1
Fix bug with nofamiliar / nocritical option of eledmac.	1
New package option sameparallelpagenumber to have the same page number for both left and right side.	1
v1.16.1.	
General: Write information about line-list file version in the correct file.	1
v1.16.2.	
General: Fix bug when adding empty lines before a \pend in combination with some specific penalties setting.	1
v1.17.0.	
General: Add compatibility of optional argument of \pstart/\pend and \AtEveryPstart/\AtEveryPend with two columns mode.	1
v1.21.0.	
General: Add \hidenumbers	16
v2.0.0.	
\@adv: \@adv defined only in reledmac.	38
\@lab: \@lab defined only in eledmac.	75
\@ref@regR: \@ref defined only in reledmac, code specific to right side moved in \ref@regR.	39
\@set: \@set defined only in reledmac.	38
\advanceline: \advanceline defined only in reledmac.	43
\bbl@set@language: Patch \bbl@set@language instead of redefining it	80
\do@lockonR: \do@lockon defined only in reledmac.	38
\endlock: \startlock and \endlock defined only in reledmac.	44
\endsub: \startsub and \endsub defined only in reledmac.	43
\fix@page: \fix@page is defined only once in reledmac	37
\l@d@set: \l@d@set defined only in reledmac.	38
\l@dbfnote: \l@dbfnote defined only in reledmac.	76
\line@marginR: \linenummargin now defined only once time in reledmac.	31
\normalbfnoteX: \normalbfnoteX defined only in reledmac.	76
\page@action: \page@action defined only in reledmac.	38
\read@linelist: \read@linelist is defined only once time in \reledmac.	36
\set@line: \set@line defined only in reledmac.	44
\set@line@action: \set@line@action defined only in reledmac.	38
\setline: \setline defined only in reledmac.	44
\setlinenum: \setlinenum defined only in reledmac.	44
\skip@lockoff: \do@lockoff defined only in reledmac.	39
\sub@action: \sub@action defined only in reledmac.	38
\sublinenumincrement*: \firstlinenum, \linenumincrement, \firstsublinenum, \sublinenumincrement are now defined only in reledmac.	32
\theledlanguageR: Patch \otherlanguage instead of redefining it.	81
General: \@nl is now defined only in reledmac.	36

\ifbypage@ and \ifbypstart@R defined in eledmac.	30
Fix some bugs with ‘sameparallelpagenumber’ option.	1
Many code refactored and moved to reledmac.	1
Package’s name becomes reledpar.	1
Totally new implementation of ‘sameparallelpagenumber’ option.	1
chapterinpages: Deleting the old system of managing parallel chapter, keep only the new one with \patchcmd.	50
v2.1.0.	
General: Fix bug when using \eledsection and related on right pages when page width is short.	1
Fix bug when using \pagenumbering with memoir (bug added in v2.0.0).	1
Fix bug with \setparledgroupnotespacing with the shiftedpstarts option.	1
Fix incompatibility between optional argument of \pstart and \numberpstarttrue	1
Options to custom empty right page before \Pages.	1
v2.2.0.	
General: astanza environment can take an optional argument, which will be the optional argument of \pstart started by this environment.	1
New tools to number stanza	1
v2.2.1.	
General: Fix bug with optional argument of last left \pend	1
v2.3.0.	
\Pages: Fix bug when calling \Columns after a \Pages (bug added in v1.13.0).	94
General: Change some internal codes in order to provide compatibility with L ^A T _E X 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.	
\ledstrutR: Deleted \ledstrutL and \ledstrutR	100
Fix bug with dotted letter	100
General: New way of (not) synchronizing the parallel pages.	1
Option to switch to \mainmatter when calling \Pages	1
v2.5.0.	
General: Disable empty lines as paragraph in astanza.	1
Fix bug introduced in v1.15.0 which made hanging indentation in verse not work anymore.	1
New commands \linenummarginR and \linenummargin*	1
v2.5.1.	
General: Fix spurious space when using optional argument of astanza environment (introduced in v2.5.0).	1
v2.5.2.	
General: Fix bug introduced in v2.5.0 with \linenummargin, \firstlinenum, \linenumincrement, \firstsublinenum, \sublinenumincrement.	1
v2.6.0.	
\l@dmake@labelsR: \@Rlineflag is not stored directly after the line number, but as a fifth argument of \the@labelX. Can be retrieved by \xflagref.	74
General: \Xlineflag and \Xendlineflag added	1
\printlinesR deleted	1
Error message when calling \Pages or \Columns without previous pages or pairs environment.	1

Fix bug with footnote numbering when using the same series of familiar footnotes on both sides.	1
Fix bug with right side title number when using title commands before <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> . . .	45
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 accurate error messages.	1