

Parallel typesetting for critical editions: the reledpar package*

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

Abstract

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

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

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

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

Contents

1 Introduction	5
1.1 Aim of this package	5
1.2 Historical overview	5
2 General	5
3 Parallel columns	6
3.1 Basic use	6
3.2 Setting	7
3.2.1 Column’s width	7
3.2.2 Column’s separator	7
3.2.3 Column’s positions	7
3.2.4 Mixing two columns and one column texts	8

*This file (reledpar.dtx) has version number v2.0.0, last revised 2015/07/19.

[†]maieul at maieul dot net

[‡]herries dot press at earthlink dot net

4 Facing pages	8
4.1 Basic usage	8
4.2 Setting	9
4.2.1 Text width	9
4.2.2 Page number	9
4.2.3 Page breaking	9
4.3 Critical and familiar footnotes	9
4.3.1 Notes height setting	9
4.3.2 Notes for one side only	10
4.3.3 Familiar notes called in the right side, but to be printed in the left side	10
5 Left and right texts	11
5.1 Environments	11
5.2 Numbering text lines and paragraphs	11
5.3 Line numbering scheme	12
5.4 Lineation system	12
5.5 Chunks	12
5.6 <code>\AtEveryPstart</code> and <code>\AtEveryPstartCall</code>	13
5.7 Language setting	13
5.8 Shifting	14
6 Verse	14
7 Side notes	14
8 Parallel ledgroups	15
8.1 General	15
8.2 Parallel ledgroups and <code>setspace</code> package	16
9 Sectioning commands	16
I Implementation overview	17
II Preliminaries	17
II.1 Package's meta-data	17
II.2 Package's requirement	17
II.3 Package's options	17
II.4 Determining side and category of parallel processing	18
II.5 Text's width	18
II.6 Messages	18
III Sectioning commands	20

IV Line counting	22
IV.1 Setting lineation reset	22
IV.2 Setting line number margin	23
IV.3 Setting lineation start and step	24
IV.4 Setting line flag	25
IV.5 Setting line number style	25
IV.6 Print marginal line number	25
IV.7 Line-number counters and lists	25
IV.7.1 Correspond to those in reledmac for regular or left text	25
IV.7.2 Specific to reledpar	26
IV.8 Reading the line-list file	26
IV.9 Commands within the line-list file	27
IV.10 Writing to the line-list file	31
V Marking text for notes	33
V.1 Specific hooks and commands for notes	33
V.1.1 Notes to be printed on one side only	33
V.1.2 Familiar footnotes without marks	34
V.1.3 Create hooks	35
V.1.4 Init standards series (A,B,C,D,E,Z)	35
VI Pstart numbers dumping and restoration	35
VII Parallel environments	36
VIII Paragraph decomposition and reassembly	38
VIII.1 Boxes, counters, \pstart and \pend	38
VIII.2 Processing one line	43
VIII.3 Line and page number computation	46
VIII.4 Line number printing	49
VIII.5 Pstart number printing in side	51
VIII.6 Add insertions to the vertical list	53
VIII.7 Penalties	53
VIII.8 Printing leftover notes	54
IX Footnotes	55
IX.1 Line number printing	55
IX.2 Footnotes output specific to \Pages	55
X Cross referencing	59
XI Side notes	60
XII Familiar footnotes	61
XIII Verse	61

XIV Naming macros	63
XV Fixing babel and polyglossia	63
XVI Counts and boxes for parallel texts	65
XVII Checking text to be processed	67
XVIII Parallel columns	68
XIX Parallel pages	75
XIX.1 Specific counters	75
XIX.2 Main macro	75
XIX.3 Ensure all notes be printed at the end of parallel pages	80
XIX.4 Struts	80
XIX.5 Page clearing	80
XIX.6 Lines managing	81
XIX.7 Page break managing	82
XIX.8 Getting boxes content	85
XIX.9 Same page number in both side	87
XX Sections' titles' commands	88
XXI Page break/no page break, depending on the specific line	89
XXII Parallel ledgroup	90
XXIII Compatibility with eledmac	93
XXIV The End	93
Appendix A Some things to do when changing version	94
Appendix A.1 Migration to eledpar 1.4.3	94
Appendix A.2 Migration from eledpar to reledpar	94
Appendix A.2.1 Deprecated options	94
Appendix A.2.2 \renewcommand replaced with command	94
Appendix A.2.3 Commands the names of which have changed	95
References	95
Index	95
Change History	113

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 2; the complete source code for the package, with extensive documentation (in sections I through XXIV); 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 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 parallel¹. 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

¹You can use, anyway, `\numberlinefalse` to disable printing of line numbers.

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. The line numbering schemes need not be the same for the two texts.

`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 in this respect.

3 Parallel columns

3.1 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 5.

`\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} ... \end{Leftside}
\begin{Rightside} ... \end{Rightside}
\end{pairs}
\Columns
\begin{pairs}
\begin{Leftside} ... \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.

3.2 Setting

3.2.1 Column's width

`\Lcolwidth` The lengths `\Lcolwidth` and `\Rcolwidth` are the widths of the left and right columns, respectively. By default, these are:
`\Rcolwidth` `\setlength{\Lcolwidth}{0.45\textwidth}`
`\setlength{\Rcolwidth}{0.45\textwidth}`
 They may be adjusted if one text tends to be 'bulkier' than the other.

3.2.2 Column's separator

`\columnrulewidth` The macro `\columnseparator` is called between each left/right pair of lines. By
`\columnseparator` default it inserts a vertical rule of width `\columnrulewidth`. As this is initially defined to be 0pt the rule is invisible. For a visible rule between the columns you could try:
`\setlength{\columnrulewidth}{0.4pt}`
 You can also modify `\columnseparator` if you want more control.

3.2.3 Column's positions

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

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

`\beforecolumnseparator`
`\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.

3.2.4 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 call `\widthliketwocolumns` as a global option when loading `reledmac` or `reledpar`.

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

4 Facing pages

4.1 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} ... \end{Leftside}
\begin{Rightside} ... \end{Rightside}
\begin{Leftside} ... \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.

4.2 Setting

4.2.1 Text width

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

4.2.2 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.

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

You can also switch the two system using `\sameparallelpagenumbertrue` and `\sameparallelpagenumberfalse`.

4.2.3 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>}`.

4.3 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.

4.3.1 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.² We have not changed this default setting, because we do not want to break compatibility with older version of reledmac and we want to be as close as possible to default L^AT_EX's feature.

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

```
\maxhXnotes{0.6\textheight}
\maxhnotesX{0.6\textheight}
```

4.3.2 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.

4.3.3 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\footnotemk. And so one ...
\pend
\endnumbering
\end{Rightside}
```

²The same applies to L^AT_EX normal notes. Read <http://tex.stackexchange.com/a/228283/7712> for technical informations.

5 Left and right texts

5.1 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
Rightside in the Rightside environment. The number of Leftside and Rightside environments must be the same.

5.2 Numbering text lines and paragraphs

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

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

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

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

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

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

```

\end{Leftside}
\begin{Rightside}
  \memorydump
  ...
\end{pages}

```

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

5.3 Line numbering scheme

Within these environments you can designate the line numbering scheme(s) to be used.

5.4 Lineation system

Following `\firstlinenum{<num>}` the first line number will be `<num>`, and following `\linenumincrement{<num>}` only every `<num>`th line will have a printed number. Using these macros inside the `Leftside` and `Rightside` environments gives you independent control over the left and right numbering schemes. The `\firstsublinenum` and `\sublinenumincrement` macros correspondingly set the numbering scheme for sublines. The starred versions change both left and right numbering schemes.

Generally speaking, controls like `\firstlinenum` or `\linenummargin` apply to sequential and left texts. To effect right texts only, they have to be within a `Rightside` environment.

`\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.

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

5.5 Chunks

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 (4.2.2 p. 15).

5.6 `\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 4.2.4 p. 16) 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`.

5.7 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.

5.8 Shifting

Corresponding left and right sides must have the same number of paragraph chunks — if there are four on the left there must be four on the right, even if some are empty. The start of each pair of left and right chunks are aligned horizontally on the page. The ends may come at different positions — if one chunk is shorter than the other then blank lines are output on the shorter side until the end of the longer chunk is reached.

However, sometime if the left pstarts are much greater than right pstarts, or *vice-versa*, you can decide to shift the pstarts on the left and right side. That means the start of pstarts are not aligned horizontally on the page, the shift is offset at the end of each double pages. To enable this function, load `reledpar` with the option `shiftedpstarts`.

6 Verse

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

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

`\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.

7 Side notes

As in `reledmac`, you must use one of the following commands to add side notes: `\led-sidenote`, `\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.

8 Parallel ledgroups

8.1 General

You can also make parallel ledgroups (see the documentation of `reledmac` about ledgroups, 9 p. 42). 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
```

8.2 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.

9 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 \LaTeX marks for header are taken from left side. You can change it, using `\eledsectmark{⟨arg⟩}`, where `⟨arg⟩` could be L (for left side) or R (for right side).

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}[2015/07/19 v2.0.0 reledmac exten-
  sion for parallel texts]%
4
```

II.2 Package’s requirement

Few commands use `\xspace` command.

```
5 \RequirePackage{xspace}%
```

II.3 Package’s options

With the option ‘`shiftedpstarts`’ a long `pstart` on the left side (or in 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.

```
\ifshiftedpstarts
```

```
6 \newif\ifshiftedpstarts
7 \DeclareOption{shiftedpstarts}{\shiftedpstartstrue}
```

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

```
8 \DeclareOption{parledgroup}{\parledgrouptrue}
```

`\ifwidthliketwocolumns` The `\widthliketwocolumns` option can be called either on `reledmac` or `reledpar`.

```
9 \DeclareOption{widthliketwocolumns}{\widthliketwocolumnstrue}%
```

The `sameparallelpagelength` enable to have the same page number in left and right side.

`\ifsameparallelpagelength`

```
10 \newif\ifsameparallelpagelength%
```

```
11 \DeclareOption{sameparallelpagelength}{\sameparallelpagelengthtrue}%
```

```
12 \ProcessOptions%
```

II.4 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.

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

II.5 Text’s width

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

```
\Rcolwidth
13 \newdimen\Lcolwidth
14 \Lcolwidth=0.45\textwidth
15 \newdimen\Rcolwidth
16 \Rcolwidth=0.45\textwidth
```

II.6 Messages

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

`\eledpar@error`

```
17 \newcommand{\reledpar@error}[2]{\PackageError{reledpar}{#1}{#2}}
```

`\led@err@TooManyPstarts`

```
18 \newcommand*{\led@err@TooManyPstarts}{%
```

```
19 \reledpar@error{Too many \string\pstart\space without printing.
20
```

```
Some text will be lost}{\@ehc}}
```

@BadLeftRightPstarts

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

@err@LeftOnRightPage

@err@RightOnLeftPage

```
24 \newcommand*{\led@err@LeftOnRightPage}{%
25   \reledpar@error{The left page has ended on a right page}{\@ehc}}
26 \newcommand*{\led@err@RightOnLeftPage}{%
27   \reledpar@error{The right page has ended on a left page}{\@ehc}}
```

@PreviousNotPrinted

@PreviousNotPrinted

```
28 \newcommand*{\led@err@Leftside@PreviousNotPrinted}{%
29   \reledpar@error{You call a new Leftside environment while the pre-
30     vious one has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}
31 \newcommand*{\led@err@Rightside@PreviousNotPrinted}{%
32   \reledpar@error{You call a new Rightside environment while the pre-
33     vious one has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}
```

@err@Pages@InsideEnv

@err@Columns@InsideEnv

```
32 \newcommand*{\led@err@Pages@InsideEnv}{%
33   \reledpar@error{\string\Pages\space must be called *out-
34     side* of the `pages` environment}{\@ehc}}
34 \newcommand*{\led@err@Columns@InsideEnv}{%
35   \reledpar@error{\string\Columns\space must be called *out-
36     side* of the `pairs` environment}{\@ehc}}
```

@fail@patch@thepage

```
36 \newcommand{\led@error@fail@patch@thepage}{%
37   \reledpar@error{Fail to patch \string\@thepage\space com-
38     mand.}{\@ehc}%
39 }
```

@patch@pagenumbering

```
39 \newcommand{\led@error@fail@patch@pagenumbering}{%
40   \reledpar@error{Fail to patch \string\pagenumbering\space com-
41     mand.}{\@ehc}%
42 }
```

@fail@patch@@outputpage

```
42 \newcommand{\led@error@fail@patch@@outputpage}{%
43   \reledpar@error{Fail to patch \string\@outputpage\space com-
44     mand.}{\@ehc}%
45 }
```

III Sectioning commands

<code>\section@numR</code>	<p>This is the right side equivalent of <code>\section@num</code>.</p> <p>Each section will read and write an associated ‘line-list file’, containing information used to do the numbering. Normally the file will be called <code>\jobname\section@num</code>, where <code>nn</code> is the section number. However, for right side texts the file is called <code>\jobname\section@numR</code>. The <code>\extensionchars</code> applies to the right side files just as it does to the normal files.</p> <pre> 45 \newcount\section@numR 46 \section@numR=\z@ </pre>
<code>\ifpst@rtedL</code> <code>\ifpst@rtedR</code>	<p><code>\ifpst@rtedL</code> is set FALSE at the start of left side numbering, and similarly for <code>\ifpst@rtedR</code>. <code>\ifpst@rtedL</code> is defined in <code>reledmac</code>.</p> <pre> 47 \pst@rtedLfalse 48 \newif\ifpst@rtedR 49 </pre>
<code>\beginnumberingR</code>	<p>This is the right text equivalent of <code>\beginnumbering</code>, and begins a section of numbered text.</p> <pre> 50 \newcommand*{\beginnumberingR}{% 51 \ifnumberingR 52 \led@err@NumberingStarted 53 \endnumberingR 54 \fi 55 \global\l@dnumpstartsR \z@ 56 \global\pst@rtedRfalse 57 \global\numberingRtrue 58 \global\advance\section@numR \@ne 59 \global\absline@numR \z@ 60 \gdef\normal@page@breakR{} 61 \gdef\l@prev@pbR{} 62 \gdef\l@prev@nopbR{} 63 \global\line@numR \z@ 64 \global\@lockR \z@ 65 \global\sub@lockR \z@ 66 \global\sublines@false 67 \global\let\next@page@numR\relax 68 \global\let\sub@change\relax 69 \message{Section \the\section@numR R }% 70 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}% 71 \l@dend@stuff 72 \setcounter{pstartR}{1} 73 \begingroup 74 \initnumbering@sectcountR 75 \gdef\eled@sectionsR@{ }% 76 \if@noeled@sec\else% 77 \makeatletter\inputIfFileExists{\jobname.eledsec\the\section@numR R}{ }{ } 78 \immediate\openout\eled@sectioningRout=\jobname.eledsec\the\section@numR R 79 \fi% </pre>

80 }

`\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.

```

81 \def\endnumberingR{%
82   \ifnumberingR
83     \global\numberingRfalse
84     \normal@pars
85     \ifnum\l@dnumpstartsR=0%
86       \led@err@NumberingWithoutPstart%
87     \fi%
88     \ifl@dpairing
89       \global\pst@rtedRfalse
90     \else
91       \ifx\insertlines@listR\empty\else
92         \global\noteschanged@true
93       \fi
94       \ifx\line@listR\empty\else
95         \global\noteschanged@true
96       \fi
97     \fi
98     \ifnoteschanged@
99       \led@mess@NotesChanged
100    \fi
101  \else
102    \led@err@NumberingNotStarted
103  \fi
104  \endgroup
105  \if@noeled@sec\else%
106    \immediate\closeout\eled@sectioningR@out%
107  \fi%
108  }
109
```

`\numbering@sectcountR` We do not want the numbering of the right-side section commands to be continuous with the numbering of the left side, we switch the \TeX counter in `\numberingR`.

```

110 \newcounter{chapterR}
111 \newcounter{sectionR}
112 \newcounter{subsectionR}
113 \newcounter{subsubsectionR}
114 \newcommand{\initnumbering@sectcountR}{
115   \let\c@chapter\c@chapterR
116   \let\c@section\c@sectionR
117   \let\c@subsection\c@subsectionR
118   \let\c@subsubsection\c@subsubsectionR
119 }

```

```

\pausenumberingR These are the right text equivalents of \pausenumbering and \resumenum-
\resumenumberingR bering.

120 \newcommand*{\pausenumberingR}{%
121   \endnumberingR\global\numberingRtrue}
122 \newcommand*{\resumenumberingR}{%
123   \ifnumberingR
124     \global\pst@rtedRtrue
125     \global\advance\section@numR \@ne
126     \led@mess@SectionContinued{\the\section@numR R}%
127     \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
128     \l@dend@stuff
129     \begingroup%
130     \initnumbering@sectcountR%
131   \else
132     \led@err@numberingShouldHaveStarted
133     \endnumberingR
134     \beginnumberingR
135   \fi}
136

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

137 \newcommand*{\memorydumpL}{%
138   \endnumbering
139   \numberingtrue
140   \global\pst@rtedLtrue
141   \global\advance\section@num \@ne
142   \led@mess@SectionContinued{\the\section@num}%
143   \line@list@stuff{\jobname.\extensionchars\the\section@num}%
144   \l@dend@stuff}
145
146 \newcommand*{\memorydumpR}{%
147   \endnumberingR
148   \numberingRtrue
149   \global\pst@rtedRtrue
150   \global\advance\section@numR \@ne
151   \led@mess@SectionContinued{\the\section@numR R}%
152   \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
153   \l@dend@stuff}
154

```

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`.

```

155 \newcommand*{\lineationR}[1]{%
156   \ifnumbering
157     \led@err@LineationInNumbered
158   \else
159     \def\@tempa{#1}\def\@tempb{page}%
160     \ifx\@tempa\@tempb
161       \global\bypage@Rtrue
162       \global\bypstart@Rfalse
163       \unless\ifnocritical@%
164         \Xpstart[][false]%
165       \fi%
166     \else
167       \def\@tempb{pstart}%
168       \ifx\@tempa\@tempb
169         \global\bypage@Rfalse
170         \global\bypstart@Rtrue
171         \unless\ifnocritical@%
172         \Xpstart%
173       \fi%
174     \else
175       \def\@tempb{section}%
176       \ifx\@tempa\@tempb
177         \global\bypage@Rfalse%
178         \global\bypstart@Rfalse%
179         \unless\ifnocritical@%
180         \Xpstart[][false]%
181       \fi%
182     \else
183       \led@warn@BadLineation
184     \fi%
185   \fi
186 \fi
187 \fi}}
```

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

```

188 \WithSuffix\newcommand\lineation*[1]{%
189   \lineation{#1}%
190   \lineationR{#1}%
191 }%
```

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

`\line@marginR`

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`.

```
192 \newcount\line@marginR
```

By default put right text numbers at the right.

```
193 \line@marginR=\@ne
```

```
194
```

IV.3 Setting lineation start and step

```
\c@firstlinenumR
\c@linenumincrementR
```

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

```
195 \newcounter{firstlinenumR}
```

```
196 \setcounter{firstlinenumR}{5}
```

```
197 \newcounter{linenumincrementR}
```

```
198 \setcounter{linenumincrementR}{5}
```

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

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

```
199 \newcounter{firstsublinenumR}
```

```
200 \setcounter{firstsublinenumR}{5}
```

```
201 \newcounter{sublinenumincrementR}
```

```
202 \setcounter{sublinenumincrementR}{5}
```

```
203
```

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

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

```
204 \WithSuffix\newcommand\firstlinenum*[1]{%
```

```
205 \setcounter{firstlinenumR}{#1}%
```

```
206 \setcounter{firstlinenum}{#1}%
```

```
207 }
```

```
208 \WithSuffix\newcommand\linenumincrement*[1]{%
```

```
209 \setcounter{linenumincrementR}{#1}%
```

```
210 \setcounter{linenumincrement}{#1}%
```

```
211 }
```

```
212 \WithSuffix\newcommand\firstsublinenum*[1]{%
```

```
213 \setcounter{subfirstlinenumR}{#1}%
```

```
214 \setcounter{subfirstlinenum}{#1}%
```

```
215 }
```



```

216 \WithSuffix\newcommand\sublinenumincrement*[1]{%
217   \setcounter{sublinenumincrementR}{#1}%
218   \setcounter{sublinenumincrement}{#1}%
219 }

```

IV.4 Setting line flag

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

```

220 \newcommand{\setRlineflag}[1]{%
221   \gdef\Rlineflag{#1}%
222 }
223 \setRlineflag{R}

```

IV.5 Setting line number style

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

```

224 \newcommand*{\linenumrepR}[1]{\@arabic{#1}}
225 \newcommand*{\sublinenumrepR}[1]{\@arabic{#1}}
226

```

IV.6 Print marginal line number

`\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`.

```

227 \newcommand*{\leftlinenumR}{%
228   \l@dlinenumR
229   \kern\linenumsep}
230 \newcommand*{\rightlinenumR}{%
231   \kern\linenumsep
232   \l@dlinenumR}
233 \newcommand*{\l@dlinenumR}{%
234   \numlabfont\linenumrepR{\line@numR}\@Rlineflag%
235   \ifsublines@
236     \ifnum\subline@num>\z@
237       \unskip\fullstop\sublinenumrepR{\subline@numR}%
238     \fi
239   \fi}
240

```

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.

```
241 \newcount\line@numR
242 \newcount\subline@numR
243 \newcount\absline@numR
244
```

`\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.

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

```
245 \list@create{\line@listR}
246 \list@create{\insertlines@listR}
247 \list@create{\actionlines@listR}
248 \list@create{\actions@listR}
249
```

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

```
250 \newcount\page@numR
251
```

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.

```
252 \list@create{\linesinpar@listL}
253 \list@create{\linesinpar@listR}
254 \list@create{\maxlinesinpar@list}
255
```

IV.8 Reading the line-list file

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

```
256 \newcommand{\list@clearing@regR}{%
257   \list@clear{\line@listR}%
258   \list@clear{\insertlines@listR}%
259   \list@clear{\actionlines@listR}%
260   \list@clear{\actions@listR}%
261   \list@clear{\linesinpar@listR}%
262   \list@clear{\linesonpage@listR}
263 }
```

`\read@linelist` `\read@linelist{\<file>}` is the control sequence that is called by `\beginnum-`
`bering` (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
`\@nl` the start of a new line of numbered text on a right side.

```

264 \newcommand{\@nl@regR}{%
265   \ifx\l@dchset@num\relax \else
266     \advance\absline@numR \@ne
267     \set@line@action
268     \let\l@dchset@num\relax
269     \advance\absline@numR \m@ne
270     \advance\line@numR \m@ne%    % do we need this?
271   \fi
272   \advance\absline@numR \@ne
273   \ifx\next@page@numR\relax \else
274     \page@action
275     \let\next@page@numR\relax
276   \fi
277   \ifx\sub@change\relax \else
278     \ifnum\sub@change>\z@
279       \sublines@true
280     \else
281       \sublines@false
282     \fi
283     \sub@action
284     \let\sub@change\relax
285   \fi
286   \ifcase\@lockR
287   \or
288     \@lockR \tw@
289   \or\or
290     \@lockR \z@
291   \fi
292   \ifcase\sub@lockR
293   \or
294     \sub@lockR \tw@
295   \or\or
296     \sub@lockR \z@
297   \fi
298   \ifsublines@

```

```

299     \ifnum\sub@lockR<\tw@
300         \advance\subline@numR \@ne
301     \fi
302 \else
303     \ifnum\@lockR<\tw@
304         \advance\line@numR \@ne \subline@numR \z@
305     \fi
306 \fi}
307
308

```

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

```

309 \newcount\last@page@numR
310 \last@page@numR=-10000
311

```

`\@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`.

```

312 \newcount\@lockR
313 \newcount\sub@lockR
314
315 \newcommand*{\do@lockonR}{%
316     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
317     \ifsublines@
318         \xright@appenditem{-1005}\to\actions@listR
319     \ifnum\sub@lockR=\z@

```

```

320     \sub@lockR \@ne
321   \else
322     \ifnum\sub@lockR=\thr@@
323       \sub@lockR \@ne
324     \fi
325   \fi
326 \else
327   \xright@appenditem{-1003}\to\actions@listR
328   \ifnum\@lockR=\z@
329     \@lockR \@ne
330   \else
331     \ifnum\@lockR=\thr@@
332       \@lockR \@ne
333     \fi
334   \fi
335 \fi}
336

```

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

```

337
338
339 \newcommand{\do@lockoffR}{%
340   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
341   \ifsublines@
342     \xright@appenditem{-1006}\to\actions@listR
343     \ifnum\sub@lockR=\tw@
344       \sub@lockR \thr@@
345     \else
346       \sub@lockR \z@
347     \fi
348   \else
349     \xright@appenditem{-1004}\to\actions@listR
350     \ifnum\@lockR=\tw@
351       \@lockR \thr@@
352     \else
353       \@lockR \z@
354     \fi
355   \fi}
356
357

```

`\n@num`

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

```
358 \newcount\insert@countR
```

- #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.

```
359 \newcommand*{\@ref@regR}[2]{%
360   \global\advance\@edtext@level by 1%
361   \global\insert@countR=#1\relax
362   \loop\ifnum\insert@countR>\z@
363     \xright@appenditem{\the\absline@numR}\to\insertlines@listR
364     \global\advance\insert@countR \m@ne
365   \repeat
```

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.

```
366 \begingroup
367   \let\@ref=\dummy@ref
368   \let\@lopR\@gobble
369   \let\page@action=\relax
370   \let\sub@action=\relax
371   \let\set@line@action=\relax
372   \let\@lab=\relax
373   \let\@lemma=\relax
374   \let\@sw\@gobblethree%
375   #2
376   \global\endpage@num=\page@numR
377   \global\endline@num=\line@numR
378   \global\endsubline@num=\subline@numR
379 \endgroup
```

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

```
380 \xright@appenditem%
381   {\the\page@numR|\the\line@numR}%
382   \ifsublines@ \the\subline@numR \else 0\fi}%
383   \the\endpage@num|\the\endline@num}%
384   \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@listR
```

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

```
385 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\@
```

Declare and init boolean for lemma in this level.

```
386 \providebool{lemmacommand@the\@edtext@level}%
387 \boolfalse{lemmacommand@the\@edtext@level}%
```

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

```
388 #2
389 % Now, we store the list of \cs{@sw} of this current \cs{edtext} as an element of
390 % the global list of list of \cs{@sw} for a \cs{edtext} depth.
391 % \begin{macrocode}
392 \ifnum\@edtext@level>0%
393 \def\create@this@edtext@level{\expandafter\list@create\expandafter{\csname sw@
394 \ifcsundef{sw@list@edtextR@the\@edtext@level}{\create@this@edtext@level}{}%
395 \letcs{\@tmpp}{sw@list@edtextR@the\@edtext@level}%
396 \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}%
397 \xright@appenditem{\expandonce\@tmpp}\to\@tmpp%
398 \global\cslet{sw@list@edtextR@the\@edtext@level}{\@tmpp}%
399 \fi%
```

Decrease edtext level counter.

```
400 \global\advance\@edtext@level by -1%
401 }
```

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

```
402 \renewcommand*{\@pend}[1]{%
403 \ifbypstart@\global\line@num=0\fi%
404 \xright@appenditem{#1}\to\linesinpar@listL}
405 \renewcommand*{\@pendR}[1]{%
406 \ifbypstartR@\global\line@numR=0\fi
407 \xright@appenditem{#1}\to\linesinpar@listR}
408
```

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

```
409 \renewcommand*{\@lopL}[1]{%
410 \xright@appenditem{#1}\to\linesonpage@listL}
411 \renewcommand*{\@lopR}[1]{%
412 \xright@appenditem{#1}\to\linesonpage@listR}
413
```

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.

<code>\linenum@outR</code>	The file for right texts will be opened on output stream <code>\linenum@outR</code> . 414 <code>\newwrite\linenum@outR</code>
<code>\iffirst@linenum@out@R</code> <code>\first@linenum@out@Rtrue</code> <code>\first@linenum@out@Rfalse</code>	Once any file is opened on this stream, we keep it open forever, or else switch to another file that we keep open. 415 <code>\newif\iffirst@linenum@out@R</code> 416 <code>\first@linenum@out@Rtrue</code>
<code>\line@list@stuffR</code>	This is the right text version of the <code>\line@list@stuff{<file>}</code> macro. It is called by <code>\beginnumberingR</code> and performs all the line-list operations needed at the start of a section. Its argument is the name of the line-list file. 417 <code>\newcommand*{\line@list@stuffR}[1]{%</code> 418 <code>\read@linelist{#1}%</code> 419 <code>\iffirst@linenum@out@R</code> 420 <code>\immediate\closeout\linenum@outR</code> 421 <code>\global\first@linenum@out@Rfalse</code> 422 <code>\immediate\openout\linenum@outR=#1</code> 423 <code>\immediate\write\linenum@outR{\string\line@list@version{\this@line@lis</code> 424 <code>\else</code> 425 <code>\if@minipage%</code> 426 <code>\leavevmode%</code> 427 <code>\fi%</code> 428 <code>\closeout\linenum@outR%</code> 429 <code>\openout\linenum@outR=#1%</code> 430 <code>\fi}</code> 431
<code>\new@lineL</code>	The <code>\new@lineL</code> macro sends the <code>\@n1</code> command to the left text line-list file, to mark the start of a new text line. 432 <code>\newcommand*{\new@lineL}{%</code> 433 <code>\write\linenum@out{\string\@n1[\the\c@page][\thepage]}</code>
<code>\new@lineR</code>	The <code>\new@lineR</code> macro sends the <code>\@n1</code> command to the right text line-list file, to mark the start of a new text line. 434 <code>\newcommand*{\new@lineR}{%</code> 435 <code>\write\linenum@outR{\string\@n1[\the\c@page][\thepage]}</code>
<code>\flag@start</code> <code>\flag@end</code>	We enclose a lemma marked by <code>\edtext</code> in <code>\flag@start</code> and <code>\flag@end</code> : these send the <code>\@ref</code> command to the line-list file. They are both defined in <code>reledmac</code> .
<code>\startsub</code> <code>\endsub</code>	<code>\startsub</code> and <code>\endsub</code> turn sub-lineation on and off, by writing appropriate instructions to the line-list file. There are both defined in <code>reledmac</code> .
<code>\advanceline</code>	You can use <code>\advanceline{<num>}</code> in running text to advance the current visible line-number by a specified value, positive or negative. It is defined in <code>reledmac</code> .
<code>\setline</code>	You can use <code>\setline{<num>}</code> in running text (i.e., within <code>\pstart... \pend</code>) to set the current visible line-number to a specified positive value. It is defined in <code>reledmac</code> .

`\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`

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

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.

```
437 \unless\ifnofamiliar@%
438 \csgdef{onlysideX@#1}{}%
439 \global\newbool{keepforsideX@#1}%
440 \fi%
441 \unless\ifnocritical@%
442 \global\newbool{keepforXside@#1}%
443 \csgdef{Xonlyside@#1}{}%
444 \fi%
```

V.1.2 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`.

```
445 \unless\ifnofamiliar@%
446 % So declare the list.
447 % \begin{macrocode}
448 \expandafter\list@create\csname footnote#1@mk\endcsname%
```

Then, declare the `\footnoteXnomk` command.

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

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

```
450 \toggletrue{nomk@}%
451 \csuse{footnote#1}{##1}%
452 \togglefalse{nomk@}%
```

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`.

```
453 \letcs{\@tmp}{footnote#1@mk}%
454 \numdef\@tmpa{\csuse{c@footnote#1}}%
455 \global\xright@appenditem{\@tmpa}\to\@tmp%
456 \global\cslet{footnote#1@mk}{\@tmp}%
457 }%
```

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

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

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

```
459 \letcs{\@tmp}{footnote#1@mk}%
460 \gl@p\@tmp\to\@tmpa%
461 \global\cslet{footnote#1@mk}{\@tmp}%
```

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

```
462 \letcs{\old@footnote}{c@footnote#1}%
463 \setcounter{footnote#1}{\@tmpa}%
```

Define the footnote mark and print it

```
464 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
465 \csuse{@footnotemark#1}%
```

Restore previous footnote counter and finally add space.

```
466 \setcounter{footnote#1}{\old@footnote}%
467 \xspace%
468 }%
```

End of tools for familiar notes without marks

```
469 \fi
End of \newseries@par.
470 }%
```

V.1.3 Create hooks

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

```
471 \unless\ifnocritical@%
472 \newhookcommand@series{Xonlyside}%
473 \fi%
474 \unless\ifnofamiliar@%
475 \newhookcommand@series{onlysideX}%
476 \fi
477
478
```

V.1.4 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.

```
479 \newcommand{\init@series@par}{%
480 \def\do##1{\newseries@par{##1}}%
481 \dolistloop{\@series}%
482 }%
483 \init@series@par%
```

VI Pstart numbers dumping and restoration

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

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

```
\list@pstartL@pc
\list@pstartR@pc
484 \list@create{\list@pstartL@pc}%
485 \list@create{\list@pstartR@pc}%
```

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

```

\dump@pstartL@pc
\dump@pstartR@pc
486 \def\dump@pstartL@pc{%
487   \xright@appenditem{\the\c@pstartL}\to\list@pstartL@pc%
488   \global\cslet{numberpstart@L\the\l@dnumpstartsL}{\ifnumberpstart}%
489 }%
490
491 \def\dump@pstartR@pc{%
492   \xright@appenditem{\the\c@pstartR}\to\list@pstartR@pc%
493   \global\cslet{numberpstart@R\the\l@dnumpstartsR}{\ifnumberpstart}%
494 }%
495

\restore@pstartL@pc And so, the commands to restore them.
\restore@pstartR@pc
496 \def\restore@pstartL@pc{%
497   \ifx\list@pstartL@pc\empty\else%
498     \gl@p\list@pstartL@pc\to\@temp%
499     \global\c@pstartL=\@temp%
500   \fi%
501 }%
502 \def\restore@pstartR@pc{%
503   \ifx\list@pstartR@pc\empty\else%
504     \gl@p\list@pstartR@pc\to\@temp%
505     \global\c@pstartR=\@temp%
506   \fi%
507 }%

```

VII Parallel environments

The initial set up for parallel processing is deceptively simple.

```

pairs The pairs environment is for parallel columns and the pages environment for par-
pages allel pages.
chapterinpages
508 \newenvironment{pairs}{%}
509   \l@dpairingtrue
510   \l@dpagingfalse
511   \initnumbering@quote
512   \at@begin@pairs%
513 }{%
514   \l@dpairingfalse
515 }
516

\AtBeginPairs The \AtBeginPairs macro just define a \at@begin@pairs macro, called at the
beginning of each pairs environments.
517 \newcommand{\AtBeginPairs}[1]{\xdef\at@begin@pairs{#1}}%
518 \def\at@begin@pairs{%
519

```

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.

```

520 \newenvironment{pages}{%
521   \l@dpairingtrue
522   \l@dpagingtrue
523   \initnumbering@quote
524   \setlength{\Lcolwidth}{\textwidth}%
525   \setlength{\Rcolwidth}{\textwidth}%
526 }{%
527   \l@dpairingfalse
528   \l@dpagingfalse
529 }
530

```

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

```

531 \newif\ifinstanzaL
532 \newif\ifinstanzaR

```

`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.

```

533 \newenvironment{Leftside}{%
534   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname\else%
535   \led@err@Leftside@PreviousNotPrinted%
536   \fi%
537   \ledRcolfalse
538   \setcounter{pstartL}{1}
539   \let\pstart\pstartL
540   \let\thepstart\thepstartL
541   \let\pend\pendL
542   \let\memorydump\memorydumpL
543   \Leftsidehook
544   \let\old@startstanza\@startstanza
545   \def\@startstanza[##1]{\global\instanzaLtrue\old@startstanza[##1]}
546 }{
547   \Leftsidehookend}

```

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

```

\Rightsidehook 548 \newcommand*{\Leftsidehook}{}
\Rightsidehookend 549 \newcommand*{\Leftsidehookend}{}
550 \newcommand*{\Rightsidehook}{}
551 \newcommand*{\Rightsidehookend}{}
552

```

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.

```

553 \newenvironment{Rightside}{%
554   \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else%
555   \led@err@Rightside@PreviousNotPrinted%
556   \fi%
557   \ledRcoltrue
558   \let\beginnumbering\beginnumberingR
559   \let\endnumbering\endnumberingR
560   \let\pausenumbering\pausenumberingR
561   \let\resumenumbering\resumenumberingR
562   \let\memorydump\memorydumpR
563   \let\thepstart\thepstartR
564   \let\pstart\pstartR
565   \let\pend\pendR
566   \let\ledpb\ledpbR
567   \let\lednopb\lednopbR
568   \let\lineation\lineationR
569   \Rightsidehook
570   \let\old@startstanza\@startstanza
571   \def\@startstanza[##1]{\global\instanzaRtrue\old@startstanza[##1]}
572 }{%
573   \ledRcolfalse
574   \Rightsidehookend
575 }
576

```

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.

```
577 \newcount\num@linesR
578 \newbox\one@lineR
579 \newcount\par@lineR
```

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

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

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

```
580
581 \newcounter{pstartL}
582 \renewcommand{\thepstartL}{\bfseries\@arabic\c@pstartL}. }
583 \newcounter{pstartR}
584 \renewcommand{\thepstartR}{\bfseries\@arabic\c@pstartR}. }
585
586 \newcommandx*{\pstartL}[1][1]{%
587   \if@nobreak%
588     \let\@oldnobreak\@nobreaktrue%
589   \else%
590     \let\@oldnobreak\@nobreakfalse%
591   \fi%
592   \@nobreaktrue%
593   \ifluatex%
594     \xdef\l@luatextextdir@L{\the\luatextextdir}%
595     \xdef\l@luatexpardir@L{\the\luatexpardir}%
596     \xdef\l@luatexbodydir@L{\the\luatexbodydir}%
597   \fi%
598   \ifnumbering \else%
599     \led@err@PstartNotNumbered%
600     \beginnumbering%
601   \fi%
602   \ifnumberedpar@%
603     \led@err@PstartInPstart%
604     \pend%
605   \fi%
```

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

```
606 \ifpstart@rtedL\else%
607   \list@clear{\inserts@list}%
608   \global\let\next@insert=\empty%
609   \global\pstart@rtedLtrue%
610 \fi%
611 \begingroup\normal@pars%
```

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.

```

612 \global\advance\l@dnumpstartsL \@ne%
613 \ifnum\l@dnumpstartsL>\l@dc@maxchunks%
614 \led@err@TooManyPstarts%
615 \global\l@dnumpstartsL=\l@dc@maxchunks%
616 \fi%
617 \global\setnamebox{\l@dLcolrawbox\the\l@dnumpstartsL}=\vbox\bgroup%

```

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.

```

618 \l@dzeropenalties%
619 \ifautopar\else%
620 \ifnumberpstart%
621 \ifsidepstartnum%
622 \else%
623 \thepstartL%
624 \fi%
625 \fi%
626 \fi%
627 \hsize=\Lcolwidth%
628 \numberedpar@true%
629 \iflabelpstart\protected@edef\@currentlabel%
630 {\p@pstartL\thepstartL}\fi%

```

Dump the optional arguments

```

631 \ifstrempy{#1}%
632 {\csgdef{before@pstartL@the\l@dnumpstartsL}{\at@every@pstart}}%
633 {\csgdef{before@pstartL@the\l@dnumpstartsL}{\noindent#1}}%
634 \at@every@pstart@call%
635 }

```

The same for right side.

```

636 \newcommandx*{\pstartR}[1][1]{%
637 \if@nobreak%
638 \let\@oldnobreak\@nobreaktrue%
639 \else%
640 \let\@oldnobreak\@nobreakfalse%
641 \fi%
642 \@nobreaktrue%
643 \ifluatex%
644 \xdef\l@luatextextdir@R{\the\luatextextdir}%
645 \xdef\l@luatexpardir@R{\the\luatexpardir}%
646 \xdef\l@luatexbodydir@R{\the\luatexbodydir}%
647 \fi%
648 \ifnumberingR \else%
649 \led@err@PstartNotNumbered%
650 \beginnumberingR%
651 \fi%

```



```

652 \ifnumberedpar@%
653   \led@err@PstartInPstart%
654   \pendR%
655 \fi%
656 \ifpstart@rtedR\else%
657   \list@clear{\inserts@listR}%
658   \global\let\next@insertR=\empty%
659   \global\pstart@rtedRtrue%
660 \fi%
661 \beginingroup\normal@pars%
662   \global\advance\l@dnumpstartsR \@ne%
663   \ifnum\l@dnumpstartsR>\l@dc@maxchunks%
664     \led@err@TooManyPstarts%
665     \global\l@dnumpstartsR=\l@dc@maxchunks%
666   \fi%
667   \global\setnamebox{\l@dc@rowbox\the\l@dnumpstartsR}=\vbox\bgroup%
668     \l@dc@zeropenalties%
669     \ifautopar\else%
670       \ifnumberpstart%
671         \ifsidepstartnum\else%
672           \thepstartR%
673         \fi%
674       \fi%
675     \fi%
676     \hsize=\Rcolwidth%
677     \numberedpar@true%
678     \iflabelpstart\protected@edef\@currentlabel%
679       {\p@pstartR\thepstartR}\fi%
680     \ifstrempy{#1}%
681       {\csgdef{before@pstartR@the\l@dnumpstartsR}{\at@every@pstart}}%
682       {\csgdef{before@pstartR@the\l@dnumpstartsR}{\noindent#1}}%
683     \at@every@pstart@call%
684   }

```

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

```

685 \newcommand*{\pendL}[1][1]{%
686   \ifnumbering \else%
687     \led@err@PendNotNumbered%
688   \fi%
689   \ifnumberedpar@ \else%
690     \led@err@PendNoPstart%
691   \fi%

```

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.

```

692 \endgraf\global\num@lines=\prevgraf\egroup%
693 \global\par@line=0%

```

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

```

694 \endgroup%
695 \ignorespaces%
696 \@oldnobreak%
697 \dump@pstartL@pc%
698 \ifnumberpstart%
699   \addtocounter{pstartL}{1}%
700 \fi
701 \parledgroup@beforenotes@save{L}%
Dump content of the optional argument.
702 \ifstrempy{#1}%
703   {\csgdef{after@pendL@the\l@dnumpstartsL}{\at@every@pend}}%
704   {\csgdef{after@pendL@the\l@dnumpstartsL}{\noindent#1}}%
705 }

```

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

```

706 \newcommand*{\pendR}[1][1]{%
707   \ifnumberingR \else%
708     \led@err@PendNotNumbered%
709   \fi%
710   \ifnumberedpar@ \else%
711     \led@err@PendNoPstart%
712   \fi%
713   \endgraf\global\num@linesR=\prevgraf\egroup%
714   \global\par@lineR=0%
715   \endgroup%
716   \ignorespaces%
717   \@oldnobreak%
718   \dump@pstartR@pc%
719   \ifnumberpstart%
720     \addtocounter{pstartR}{1}%
721   \fi%
722   \parledgroup@beforenotes@save{R}%
723   \ifstrempy{#1}%
724     {\csgdef{after@pendR@the\l@dnumpstartsR}{\at@every@pend}}%
725     {\csgdef{after@pendR@the\l@dnumpstartsR}{\noindent#1}}%
726 }
727

```

`\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.

```

728 \newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
729 \gdef\at@every@pstart@call{}%

```

`\ifprint@last@after@pendL` `\ifprint@last@after@pendR` Two booleans set to true, when the time is to print the last optional argument of a `\pend`.

```

730 \newif\ifprint@last@after@pendL%
731 \newif\ifprint@last@after@pendR%

```

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
`\l@drightbox` right text.

```
732 \newbox\l@dleftbox
733 \newbox\l@drightbox
734
```

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

```
\countRline
735 \newcount\countLline
736 \countLline \z@
737 \newcount\countRline
738 \countRline \z@
739
```

`\@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`

```
740 \newcount\@donereallinesL
741 \newcount\@donetotallinesL
742 \newcount\@donereallinesR
743 \newcount\@donetotallinesR
744
```

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

```
745 \newcommand*{\do@lineL}{%
746 \letcs{\ifnumberpstart}{numberpstart@L\the\l@dpscl}%
747 \advance\countLline \@ne%
748 \ifvbox\namebox{l@dLcolrawbox\the\l@dpscl}%
749 {\vbadness=10000%
750 \splittopskip=\z@%
751 \do@lineLhook%
752 \l@emptyd@ta%
753 \global\setbox\one@line=\vsplit\namebox{l@dLcolrawbox\the\l@dpscl}%
754 to\baselineskip}%
755 \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\parledgroup@notes@startL}{}%
756 \unvbox\one@line \global\setbox\one@line=\lastbox%
757 \getline@numL%
758 \ifnum\@lock>\@ne%
759 \inserthangingsymboltrue%
760 \else%
761 \inserthangingsymbolfalse%
762 \fi%
763 \setbox\l@dleftbox%
```

```

764 \hb@xt@ \Lcolwidth{%
765   \ifl@dhidenumber%
766     \global\l@dhidenumberfalse%
767     \f@x@l@cks%
768   \else%
769     \affixline@num%
770   \fi%
771   \xifinlist{\the\l@dpscL}{\eled@sections@@}%
772     {\add@inserts\affixside@note}%
773     {\print@lineL}}%
774 \add@penaltiesL%
775 \global\advance\@donereallinesL\@ne%
776 \global\advance\@donetotallinesL\@ne%
777 \else%
778 \setbox\l@dleftbox \hb@xt@ \Lcolwidth{\hspace*{\Lcolwidth}}%
779 \global\advance\@donetotallinesL\@ne%
780 \fi}
781
782

```

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

```

783 \def\print@lineL{%
784   \affixpstart@numL%
785   \l@dld@ta %space kept for backward compatibility
786   \add@inserts\affixside@note%
787   \l@dlsn@te %space kept for backward compatibility
788   {\ledllfill\hb@xt@ \Lcolwidth{%
789     \do@insidelineLhook%
790     \ifluatex%
791       \luatextextdir\l@luatextextdir@L%
792     \fi%
793     \new@lineL%
794     \inserthangingsymbolL%
795     \l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta%
796   \l@drsn@te}}
797

```

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

```

798 \def\print@eledsectionL{%%
799   \addtocounter{pstartL}{-1}%
800   \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{ }
801   \ifdefstring{\@eledsectmark}{L}{ }{\ledsectnomark}
802   \numdef{\temp@}{\l@dpscL-1}%
803   \xifinlist{\temp@}{\eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
804   \@eled@sectioningtrue%
805   \bgroup%
806     \ifluatex%
807       \luatextextdir\l@luatextextdir@L%
808       \luatexpardir\l@luatexpardir@L%

```

```

809      \luatexbodydir\l@luatexbodydir@L%
810      \ifdefstring{\l@luatextextdir@L}{TRT}{\@RTLtrue}{}%
811      \fi%
812      \csuse{eled@sectioning@the\l@dpscl}%
813      \egroup%
814      \@eled@sectioningfalse%
815      \global\csundef{eled@sectioning@the\l@dpscl}%
816      \if@RTL%
817        \hspace{-3\paperwidth}%
818        {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
819      \else%
820        \hspace{3\paperwidth}%
821        {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
822      \fi%
823      \vskip\eledsection@correcting@skip%
824    }
825

```

`\dolineLhook` These high-level commands just redefine the low-level commands. They have to be used
`\dolineRhook` be user, without `\makeatletter`.
`\doinsidelineLhook` 826 `\newcommand*{\dolineLhook}[1]{\gdef\do@lineLhook{#1}}%`
`\doinsidelineRhook` 827 `\newcommand*{\dolineRhook}[1]{\gdef\do@lineRhook{#1}}%`
828 `\newcommand*{\doinsidelineLhook}[1]{\gdef\do@insidelineLhook{#1}}%`
829 `\newcommand*{\doinsidelineRhook}[1]{\gdef\do@insidelineRhook{#1}}%`
830

`\do@lineLhook` Hooks, initially empty, into the respective `\do@line(L/R)` macros.
`\do@lineRhook` 831 `\newcommand*{\do@lineLhook}{}%`
`\do@insidelineLhook` 832 `\newcommand*{\do@lineRhook}{}%`
`\do@insidelineRhook` 833 `\newcommand*{\do@insidelineLhook}{}%`
834 `\newcommand*{\do@insidelineRhook}{}%`
835

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

```

836 \newcommand*{\do@lineR}{%
837   \letcs{\ifnumberpstart}{numberpstart@R\the\l@dpscr}%
838   \ledRcol@true%
839   \advance\countRline \@ne%
840   \ifvbox\namebox{\l@dRcolrawbox\the\l@dpscr}%
841   {\vbadness=10000%
842    \splittopskip=\z@%
843    \do@lineRhook%
844    \l@emptyd@ta%
845    \global\setbox\one@lineR=\vsplit\namebox{\l@dRcolrawbox\the\l@dpscr}%
846    to\baselineskip}%
847   \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\parledgroup@notes@startR}{}%
848   \unvbox\one@lineR \global\setbox\one@lineR=\lastbox%

```

```

849 \getline@numR%
850 \ifnum\@lockR>\@ne%
851   \inserthangingsymbolRtrue%
852 \else%
853   \inserthangingsymbolRfalse%
854 \fi%
855 \setbox\l@drightbox%
856 \hb@xt@ \Rcolwidth{%
857   \ifl@dhiddenumber%
858     \global\l@dhiddenumberfalse%
859     \f@x@l@cksR%
860   \else%
861     \affixline@numR%
862   \fi%
863   \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
864   {\add@insertsR\affixside@noteR}%
865   {\print@lineR}%
866 }%
867 \add@penaltiesR%
868 \global\advance\@donereallinesR\@ne%
869 \global\advance\@donetotallinesR\@ne%
870 \else%
871   \setbox\l@drightbox \hb@xt@ \Rcolwidth{\hspace*{\Rcolwidth}}%
872   \global\advance\@donetotallinesR\@ne%
873 \fi%
874 \ledRcol@false%
875 }
876
877

```

```

\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.

```

878 \newcommand*{\getline@numR}{%
879   \global\advance\absline@numR \@ne
880   \do@actionsR
881   \do@ballastR
882   \ifledgroupnotesR\else
883     \ifnumberline
884       \ifsublines@
885         \ifnum\sub@lockR<\tw@
886           \global\advance\subline@numR \@ne
887         \fi
888       \else
889         \ifnum\@lockR<\tw@

```

```

890             \global\advance\line@numR \@ne
891             \global\subline@numR \z@
892         \fi
893     \fi
894 \fi
895 \fi
896 }
897 \newcommand*{\getline@numL}{%
898     \global\advance\absline@num \@ne
899     \do@actions
900     \do@ballast
901     \ifledgroupnotesL@\else
902     \ifnumberline
903     \ifsublines@
904     \ifnum\sub@lock<\tw@
905     \global\advance\subline@num \@ne
906     \fi
907     \else
908     \ifnum\@lock<\tw@
909     \global\advance\line@num \@ne
910     \global\subline@num \z@
911     \fi
912     \fi
913 \fi
914 \fi
915 }
916
917

```

`\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.

```

918 \newcommand*{\do@ballastR}{\global\ballast@count=\z@
919     \beginngroup
920     \advance\absline@numR \@ne
921     \ifnum\next@actionlineR=\absline@numR
922     \ifnum\next@actionR>-1001
923     \global\advance\ballast@count by -\c@ballast
924     \fi
925     \fi
926     \endgroup}

```

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

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

```

927
928 \newif\ifl@dskipversenumberR
929 \newcommand*{\do@actions@fixedcodeR}{%
930     \ifcase\l@dttempcnta%

```

```

931 \or% % 1001
932 \global\sublines@true
933 \or% % 1002
934 \global\sublines@false
935 \or% % 1003
936 \global\@lockR=\@ne
937 \or% % 1004%
938 \ifnum\@lockR=\tw@
939 \global\@lockR=\thr@@
940 \else
941 \global\@lockR=\z@
942 \fi
943 \or% % 1005
944 \global\sub@lockR=\@ne
945 \or% % 1006
946 \ifnum\sub@lockR=\tw@
947 \global\sub@lockR=\thr@@
948 \else
949 \global\sub@lockR=\z@
950 \fi
951 \or% % 1007
952 \l@dskipnumbertrue
953 \or% % 1008
954 \l@dskipversenumberRtrue%
955 \or% % 1009
956 \l@dhidenumbertrue%
957 \else%
958 \led@warn@BadAction
959 \fi%
960 }
961
962
963 \newcommand*{\do@actionsR}{%
964 \global\let\do@actions@nextR=\relax
965 \@l@tempcntb=\absline@numR
966 \ifnum\@l@tempcntb<\next@actionlineR\else
967 \ifnum\next@actionR>-1001\relax
968 \global\page@numR=\next@actionR
969 \ifbypage@R
970 \global\line@numR \z@ \global\subline@numR \z@
971 \fi
972 \else
973 \ifnum\next@actionR<-4999\relax % 9/05 added relax here
974 \@l@tempcnta=-\next@actionR
975 \advance\@l@tempcnta by -5001\relax
976 \ifsublines@
977 \global\subline@numR=\@l@tempcnta
978 \else
979 \global\line@numR=\@l@tempcnta
980 \fi

```



```

981     \else
982         \@l@tempcnta=-\next@actionR
983         \advance\@l@tempcnta by -1000\relax
984         \do@actions@fixedcodeR
985     \fi
986 \fi
987 \ifx\actionlines@listR\empty
988     \gdef\next@actionlineR{1000000}%
989 \else
990     \gl@p\actionlines@listR\to\next@actionlineR
991     \gl@p\actions@listR\to\next@actionR
992     \global\let\do@actions@nextR=\do@actionsR
993 \fi
994 \fi
995 \do@actions@nextR}
996

```

VIII.4 Line number printing

```

\l@dcalcnun \affixline@numR is the right text version of the \affixline@num macro.
\ch@cksub@l@ckR 997
\ch@ck@l@ckR 998 \newcommand*{\l@dcalcnun}[3]{%
\fx@l@cksR 999 \ifnum #1 > #2\relax
\affixline@numR 1000 \l@tempcnta = #1\relax
1001 \advance\l@tempcnta by -#2\relax
1002 \divide\l@tempcnta by #3\relax
1003 \multiply\l@tempcnta by #3\relax
1004 \advance\l@tempcnta by #2\relax
1005 \else
1006 \l@tempcnta=#2\relax
1007 \fi}
1008
1009 \newcommand*{\ch@cksub@l@ckR}{%
1010 \ifcase\sub@lockR
1011 \or
1012 \ifnum\sublock@disp=\@ne
1013 \l@tempcntb \z@ \l@tempcnta \@ne
1014 \fi
1015 \or
1016 \ifnum\sublock@disp=\tw@
1017 \else
1018 \l@tempcntb \z@ \l@tempcnta \@ne
1019 \fi
1020 \or
1021 \ifnum\sublock@disp=\z@
1022 \l@tempcntb \z@ \l@tempcnta \@ne
1023 \fi
1024 \fi}
1025

```

```

1026 \newcommand*{\check@l@ckR}{%
1027   \ifcase\@lockR
1028   \or
1029     \ifnum\lock@disp=\@ne
1030       \@l@tempcntb \z@ \l@tempcnta \@ne
1031     \fi
1032   \or
1033     \ifnum\lock@disp=\tw@
1034     \else
1035       \@l@tempcntb \z@ \l@tempcnta \@ne
1036     \fi
1037   \or
1038     \ifnum\lock@disp=\z@
1039       \@l@tempcntb \z@ \l@tempcnta \@ne
1040     \fi
1041   \fi}
1042
1043 \newcommand*{\f@x@l@cksR}{%
1044   \ifcase\@lockR
1045   \or
1046     \global\@lockR \tw@
1047   \or \or
1048     \global\@lockR \z@
1049   \fi
1050   \ifcase\sub@lockR
1051   \or
1052     \global\sub@lockR \tw@
1053   \or \or
1054     \global\sub@lockR \z@
1055   \fi}
1056
1057
1058 \newcommand*{\affixline@numR}{%
1059 \ifledgroupnotesR\else\ifnumberline
1060 \ifl@dskipnumber
1061   \global\l@dskipnumberfalse
1062 \else
1063   \ifsublines@
1064     \@l@tempcntb=\subline@numR
1065     \l@dcalcnum{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR}
1066     \ch@cksub@lockR
1067   \else
1068     \@l@tempcntb=\line@numR
1069     \ifx\linenumberlist\empty
1070       \l@dcalcnum{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%
1071     \else
1072       \@l@tempcnta=\line@numR
1073       \edef\rem@inder{\,\linenumberlist,\number\line@numR,}%
1074       \edef\sc@n@list{\def\noexpand\sc@n@list
1075         ###1,\number\@l@tempcnta,###2|{\def\noexpand\rem@inder{####2}}}%

```

```

1076 \sc@n@list\expandafter\sc@n@list\rem@inder|{%
1077 \ifx\rem@inder\empty\advance\@l@tempcnta\@ne\fi
1078 \fi
1079 \ch@ck@l@ckR
1080 \fi
1081 \ifnum\@l@tempcnta=\@l@tempcntb
1082 \ifl@dskipversenumberR\else
1083 \if@twocolumn
1084 \if@firstcolumn
1085 \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1086 \else
1087 \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1088 \fi
1089 \else
1090 \l@tempcntb=\line@marginR
1091 \ifnum\@l@tempcntb>\@ne
1092 \advance\@l@tempcntb by\page@numR
1093 \fi
1094 \ifodd\@l@tempcntb
1095 \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1096 \else
1097 \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1098 \fi
1099 \fi
1100 \fi
1101 \fi
1102 \f@x@l@cksR
1103 \fi
1104 \fi
1105 \fi}

```

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@numL
\affixpstart@numR 1106
\leftpstartnumR 1107 \newcommand*{\affixpstart@numL}{%
\rightpstartnumR 1108 \ifsidepstartnum
\leftpstartnumL 1109 \if@twocolumn
\rightpstartnumL 1110 \if@firstcolumn
\ifpstartnumR 1111 \gdef\l@dld@ta{\llap{\leftpstartnumL}}}%
1112 \else
1113 \gdef\l@drd@ta{\rlap{\rightpstartnumL}}}%

```

```

1114     \fi
1115   \else
1116     \@l@tempcntb=\line@margin
1117     \ifnum\@l@tempcntb>\@ne
1118       \advance\@l@tempcntb \page@num
1119     \fi
1120     \ifodd\@l@tempcntb
1121       \gdef\l@drd@ta{\rlap{{\rightpstartnumL}}}%
1122     \else
1123       \gdef\l@dld@ta{\llap{{\leftpstartnumL}}}%
1124     \fi
1125   \fi
1126 \fi
1127 }
1128 \newcommand*{\affixpstart@numR}{%
1129 \ifsidepstartnum
1130 \if@twocolumn
1131   \if@firstcolumn
1132     \gdef\l@dld@ta{\llap{{\leftpstartnumR}}}%
1133   \else
1134     \gdef\l@drd@ta{\rlap{{\rightpstartnumR}}}%
1135   \fi
1136 \else
1137   \@l@tempcntb=\line@marginR
1138   \ifnum\@l@tempcntb>\@ne
1139     \advance\@l@tempcntb \page@numR
1140   \fi
1141   \ifodd\@l@tempcntb
1142     \gdef\l@drd@ta{\rlap{{\rightpstartnumR}}}%
1143   \else
1144     \gdef\l@dld@ta{\llap{{\leftpstartnumR}}}%
1145   \fi
1146 \fi
1147 \fi
1148 }
1149
1150 \newcommand*{\leftpstartnumL}{%
1151 \ifpstartnum
1152 \thepstartL
1153 \kern\linenumsep\global\pstartnumfalse\fi
1154 }
1155 \newcommand*{\rightpstartnumL}{%
1156 \ifpstartnum\kern\linenumsep
1157 \thepstartL
1158 \global\pstartnumfalse\fi
1159 }
1160 \newif\ifpstartnumR
1161 \pstartnumRtrue
1162 \newcommand*{\leftpstartnumR}{%
1163 \ifpstartnumR

```

```

1164 \thepstartR
1165 \kern\linenumsep\global\pstartnumRfalse\fi
1166 }
1167 \newcommand*{\rightpstartnumR}{
1168 \ifpstartnumR\kern\linenumsep
1169 \thepstartR
1170 \global\pstartnumRfalse\fi
1171 }

```

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.

```

1172 \list@create{\inserts@listR}

```

`\add@insertsR` The right text version.

```

\add@inserts@nextR 1173 \newcommand*{\add@insertsR}{%
1174 \global\let\add@inserts@nextR=\relax
1175 \ifx\inserts@listR\empty \else
1176 \ifx\next@insertR\empty
1177 \ifx\insertlines@listR\empty
1178 \global\noteschanged@true
1179 \gdef\next@insertR{100000}%
1180 \else
1181 \gl@p\insertlines@listR\to\next@insertR
1182 \fi
1183 \fi
1184 \ifnum\next@insertR=\absline@numR
1185 \gl@p\inserts@listR\to\@insertR
1186 \@insertR
1187 \global\let\@insertR=\undefined
1188 \global\let\next@insertR=\empty
1189 \global\let\add@inserts@nextR=\add@insertsR
1190 \fi
1191 \fi
1192 \add@inserts@nextR}
1193

```

VIII.7 Penalties

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

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

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

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

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

```
1194 \newcommand*{\add@penaltiesL}{}
1195 \newcommand*{\add@penaltiesR}{}
1196
```

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.

```
1197 \newcommand*{\flush@notesR}{%
1198 \@xloop
1199 \ifx\inserts@listR\empty\else
1200 \gl@p\inserts@listR\to\@insertR
1201 \@insertR
1202 \global\let\@insertR=\undefined
```

```

1203 \repeat}
1204

```

IX Footnotes

IX.1 Line number printing

The `\printlinesR` macro prints the line numbers for a note—which, in the general case, is a rather complicated task. The seven parameters of the argument are the line numbers as stored in `\l@d@nums`, in the form described on V.9 p. 77 of Eledmac’ handbook: the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma.

```

\printlinesR This is the right text version of \printlines and takes account of \@Rlineflag.
Just a reminder of the arguments:
\printlinesR #1 | #2 | #3 | #4 | #5 | #6 | #7
\printlinesR start-page | line | subline | end-page | line | sub-
line | font
1205 \def\printlinesR#1|#2|#3|#4|#5|#6|#7|{\begingroup
1206 \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
1207 \ifl@d@pnum #1\fullstop\fi
1208 \ifledplinenum \linenumr@p{#2}\@Rlineflag\else \symlinenum\fi
1209 \ifl@d@ssub \fullstop \sublinenumr@p{#3}\fi
1210 \ifl@d@dash \endashchar\fi
1211 \ifl@d@pnum #4\fullstop\fi
1212 \ifl@d@elin \linenumr@p{#5}\@Rlineflag\fi
1213 \ifl@d@esl \ifl@d@elin \fullstop\fi \sublinenumr@p{#6}\fi
1214 \endgroup}
1215
1216

```

IX.2 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 proceed³:

- 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

³See <http://tex.stackexchange.com/a/230332/7712>.

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.

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`.

```
1217 \newcommand\print@Xnotes@forpages[1]{%
```

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

```
1218   \ifcsemt{Xonlyside@#1}{%
1219       \csuse{#1footstart}{#1}%
```



```

1220     \csuse{#1footgroup}{#1}%
1221 }%

```

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

```

1222 {%
1223   \ifboolexpr{%
1224     ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page}}))%
1225     or%
1226     (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}}))%
1227   }%

```

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

```

1228 {%
1229   \correct@Xfootins@box{#1}%
1230   \csuse{#1footstart}{#1}%
1231   \csuse{#1footgroup}{#1}%

```

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

```

1232   \global\count\csuse{#1footins}=0%
1233   \global\skip\csuse{#1footins}=0pt%

```

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

```

1234   \csuse{Xnotefontsize@#1}%
1235   \global\advance\dimen\csuse{#1footins} by -
    \baselineskip%

```

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

```

1236   \global\boolfalse{keepforXside@#1}%
1237 }%

```

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.

```

1238 {%
1239   \global\booltrue{keepforXside@#1}%

```

Then restore expected rooms for notes on the next page.

```

1240   \global\count\csuse{#1footins}=\csuse{default@#1footins}%
1241   \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%

```

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

```

1242   \bgroup%
1243     \csuse{Xnotefontsize@#1}%
1244     \global\advance\dimen\csuse{#1footins} by \base-
lineskip%
1245   \egroup%
1246 % End of \cs{print@Xnotes@forpages}.
1247 }%
1248 }%
1249 }%

```

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

```
1250 \newcommand{\correct@Xfootins@box}[1]{%
```

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.

```
1251 \ifbool{keepforXside@#1}{%
```

Some setting needed to do the right splitting.

```
1252 \csuse{Xnotefontsize@#1}%
```

```
1253 \splittopskip=0pt%
```

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

```
1254 \global\setbox\csuse{#1footins}=\vbox{%
```

```
1255 \vsplit\csuse{#1footins} to \dimexpr\ht\csuse{#1footins}-
1pt\relax%
```

```
1256 \vskip \dimexpr-0.5\baselineskip-0.5\lineskip-
0.5pt\relax%
```

```
1257 \unvbox\csuse{#1footins}%
```

```
1258 }%
```

End of the macro.

```
1259 }{}%
```

```
1260 }%
```

And now, the same for familiar footnotes.

```
1261 \newcommand\print@notesX@forpages[1]{%
```

```
1262 \ifcsempy{onlysideX@#1}{%
```

```
1263 \csuse{footstart#1}{#1}%
```

```
1264 \csuse{footgroup#1}{#1}%
```

```
1265 }%
```

```
1266 {%
```

```
1267 \ifboolexpr{%
```

```
1268 ((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page}}
1269 or%
```

```
1270 (test {\ifcsstring{onlysideX@#1}{R}} and test{\ifnumodd{\c@page}}))%
```

```
1271 }%
```

```
1272 {%
```

```
1273 \correct@footinsX@box{#1}%
```

```
1274 \csuse{footstart#1}{#1}%
```

```
1275 \csuse{footgroup#1}{#1}%
```

```
1276 \global\count\csuse{footins#1}=0%
```

```
1277 \global\skip\csuse{footins#1}=0pt%
```

```
1278 \csuse{notefontsizeX@#1}%
```

```
1279 \global\advance\dimen\csuse{footins#1} by -
\baselineskip%
```

```
1280 \global\boolfalse{keepforsideX@#1}%
```

```
1281 }%
```

```
1282 {%
```

```
1283 \global\booltrue{keepforsideX@#1}%
```

```
1284 \global\count\csuse{footins#1}=\csuse{default@footins#1}%
```

```

1285     \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
1286     \bgroup%
1287         \csuse{notefontsizeX@#1}%
1288     \global\advance\dimen\csuse{footins#1} by \base-
lineskip%
1289     \egroup%
1290 }%
1291 }%
1292 }%
1293 \newcommand{\correct@footinsX@box}[1]{%
1294     \ifbool{keepforsideX@#1}{%
1295         \csuse{notefontsizeX@#1}%
1296         \splittopskip=0pt%
1297         \global\setbox\csuse{footins#1}=\vbox{%
1298             \vsplit\csuse{footins#1} to \dimexpr\ht\csuse{footins#1}-
1pt\relax%
1299             \vskip \dimexpr-0.5\baselineskip-0.5\lineskip-
0.5pt\relax%
1300             \unvbox\csuse{footins#1}%
1301         }%
1302     }{}%
1303 }%

```

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.

```

1304 \list@create{\labelref@listR}
1305

```

`\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`.

```

1306 \def\l@dmake@labelsR#1|#2|#3|#4|#5{%
1307     \expandafter\ifx\csname the@label#5\endcsname \relax\else
1308         \led@warn@DuplicateLabel{#4}%
1309     \fi
1310     \expandafter\gdef\csname the@label#5\endcsname{#1|#2\@Rlineflag|#3|#4}%
1311     \ignorespaces}
1312 \AtBeginDocument{%
1313     \def\l@dmake@labelsR#1|#2|#3|#4|#5{%
1314 }
1315

```

`\@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`, `\@n1`, 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* 1316 \WithSuffix\newcommand\sidenotemargin*[1]{%
1317   \l@dgetsidenote@margin{#1}
1318   \global\sidenote@marginR=\@l@tempcntb
1319   \global\sidenote@margin=\@l@tempcntb
1320 }
1321 \newcount\sidenote@marginR
1322 \global\sidenote@margin=\@ne
1323

\affixside@noteR The right text version of \affixside@note.
1324 \newcommand*{\affixside@noteR}{%
1325   \def\sidenotecontent@{}%
1326   \numgdef{\itemcount@}{0}%
1327   \def\do##1{%
1328     \ifnumequal{\itemcount@}{0}%
1329     {%
1330       \appto\sidenotecontent@{##1}}% Not print not sep-
arator before the 1st note
1331       {\appto\sidenotecontent@{\sidenotesep ##1}%
1332       }%
1333       \numgdef{\itemcount@}{\itemcount@+1}%
1334     }%
1335     \dolistloop{\l@dcnotetext}%
1336     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
1337     \gdef\@templ@d{}%
1338     \gdef\@templ@n{\l@dcnotetext\l@dcnotetext@l\l@dcnotetext@r}%
1339     \ifx\@templ@d\@templ@n \else%
1340       \if@twocolumn%
1341         \if@firstcolumn%
1342           \setl@dlp@rbox{##1}{\sidenotecontent@}%
1343         \else%
1344           \setl@drp@rbox{\sidenotecontent@}%
1345         \fi%
1346       \else%
1347         \@l@tempcntb=\sidenote@marginR%
1348         \ifnum\@l@tempcntb>\@ne%
1349           \advance\@l@tempcntb by\page@numR%
1350         \fi%
```

```

1351     \ifodd\l@dttempcntb%
1352         \setl@drp@rbox{\sidenotecontent@}%
1353         \gdef\sidenotecontent@{}%
1354         \numdef{\itemcount@}{0}%
1355         \dolistloop{\l@dcsnotetext@l}%
1356     \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
1357         \setl@dlp@rbox{\sidenotecontent@}%
1358     \else%
1359         \setl@dlp@rbox{\sidenotecontent@}%
1360         \gdef\sidenotecontent@{}%
1361         \numdef{\itemcount@}{0}%
1362         \dolistloop{\l@dcsnotetext@r}%
1363     \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
1364         \setl@drp@rbox{\sidenotecontent@}%
1365     \fi%
1366 \fi%
1367 \fi%
1368 }
1369

```

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
\inserthangingsymbolR
1370 \newif\ifinserthangingsymbolR
1371 \newcommand{\inserthangingsymbolL}{%
1372     \ifinserthangingsymbol%
1373         \ifinstanzaL%
1374             \hspace \ifundefined{sza@0@}{0}{\expandafter%
1375                 \noexpand\csname sza@0@\endcsname}\stanzaindentbase%
1376             \@hangingsymbol%
1377         \fi%
1378     \fi%
1379 }%
1380 \newcommand{\inserthangingsymbolR}{%

```

```

1381 \ifinserthangingsymbolR%
1382 \ifinstanzaR%
1383     \hskip \@ifundefined{sza@0@}{0}{\expandafter%
1384         \noexpand\csname sza@0@\endcsname}\stanzaindentbase%
1385     \@hangingsymbol%
1386 \fi%
1387 \fi%
1388 }%

```

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.

```

1389 \chardef\next=\catcode`&
1390 \catcode`&=\active
1391

```

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

```

1392 \newenvironment{astanza}{%
1393     \catcode`&\active
1394     \global\stanza@count\@ne\stanza@modulo\@ne
1395     \ifnum\usernamecount{sza@0@}=\z@
1396         \let\stanza@hang\relax
1397         \let\endlock\relax
1398     \else
1399         \rightskip\z@ plus 1fil\relax
1400     \fi
1401     \ifnum\usernamecount{szp@0@}=\z@
1402         \let\sza@penalty\relax
1403     \fi
1404     \def&{%
1405         \endlock\mbox{}%
1406         \sza@penalty
1407         \global\advance\stanza@count\@ne
1408         \@astanza@line}%
1409     \def\&{\@stopastanza}%
1410     \pstart
1411     \@astanza@line
1412 }{}
1413

```

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

```

1414 \newcommandx{\@stopastanza}[1][1,usedefault]{%
1415     \endlock\mbox{}%
1416     \pend[#1]%
1417 }%

```

`\@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.

```

1418 \newcommand*{\@astanza@line}{%

```

```

1419 \ifnum\value{stanzaindentrepetition}=0
1420     \parindent=\csname sza@\number\stanza@count
1421             @\endcsname\stanzaindentbase
1422 \else
1423     \parindent=\csname sza@\number\stanza@modulo
1424             @\endcsname\stanzaindentbase
1425     \managestanza@modulo
1426 \fi
1427 \par
1428 \stanza@hang%\mbox{}%
1429 \ignorespaces}
1430

```

Lastly reset the modified category codes.

```

1431 \catcode`\&=\next
1432

```

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.

<code>\newnamebox</code>	A set of macros for creating and using ‘named’ boxes; the macros are called after the
<code>\setnamebox</code>	regular box macros, but including the string ‘name’.
<code>\unhnamebox</code>	
<code>\unvnamebox</code>	
<code>\namebox</code>	

```

1433 \providecommand*{\newnamebox}[1]{%
1434     \expandafter\newbox\csname #1\endcsname}
1435 \providecommand*{\setnamebox}[1]{%
1436     \expandafter\setbox\csname #1\endcsname}
1437 \providecommand*{\unhnamebox}[1]{%
1438     \expandafter\unhbox\csname #1\endcsname}
1439 \providecommand*{\unvnamebox}[1]{%
1440     \expandafter\unvbox\csname #1\endcsname}
1441 \providecommand*{\namebox}[1]{%
1442     \csname #1\endcsname}
1443

```

<code>\newnamecount</code>	Macros for creating and using ‘named’ counts.
<code>\usenamecount</code>	

```

1444 \providecommand*{\newnamecount}[1]{%
1445     \expandafter\newcount\csname #1\endcsname}
1446 \providecommand*{\usenamecount}[1]{%
1447     \csname #1\endcsname}
1448

```

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 1449 \newif\ifl@dusedbabel
\l@dusedbabeltrue
\l@dchecklang

\babel@set@language In babel the macro \babel@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 \.
1450 \patchcmd{\babel@set@language}%
1451   {\select@language{\language}}%
1452   {\edef\language{#1}\select@language{\language}}%
1453   {}%
1454   {}%
1455

```

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 \theled-
\l@duselanguage languageR are the names of the languages of the left and right texts. \l@duselanguage
\theledlanguageL is similar to \selectlanguage.
\theledlanguageR 1456 \newcommand*{\l@duselanguage}[1]{}
1457 \gdef\theledlanguageL{}
1458 \gdef\theledlanguageR{}
1459

```

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

```

1460 \AtBeginDocument{%
1461   \@ifundefined{xpg@main@language}{%
1462     \@ifundefined{babel@main@language}{%

```

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

```

1463   \l@dusedbabelfalse
1464   }{%

```

Here we deal with the case where babel has been used. \selectlanguage has to be redefined to use our version of \babel@set@language and to store the left or right language.


```

1465 \l@dusedbabeltrue
1466 \let\l@doldselectlanguage\selectlanguage
1467 \let\l@doldbbl@set@language\bbl@set@language
1468 \renewcommand{\selectlanguage}[1]{%
1469   \l@doldselectlanguage{#1}%
1470   \ifledRcol \gdef\theledlanguageR{#1}%
1471   \else      \gdef\theledlanguageL{#1}%
1472   \fi}

```

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

```

1473 \renewcommand*{\l@duselanguage}[1]{%
1474   \l@doldselectlanguage{#1}}

```

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

```

1475 \gdef\theledlanguageL{\bbl@main@language}%
1476 \gdef\theledlanguageR{\bbl@main@language}%
1477 }%
1478 }

```

If use polyglossia

```

1479 { \let\old@otherlanguage\otherlanguage%
1480   \renewcommand{\otherlanguage}[2][1]{%
1481     \selectlanguage[#1]{#2}%
1482     \ifledRcol \gdef\theledlanguageR{#2}%
1483     \else      \gdef\theledlanguageL{#2}%
1484     \fi}%
1485   \let\l@duselanguage\select@language%
1486   \gdef\theledlanguageL{\xpg@main@language}%
1487   \gdef\theledlanguageR{\xpg@main@language}%

```

That is it.

```

1488 } }

```

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 5120 chunk pairs.

```

1489 \newcount\l@dc@maxchunks
1490 \newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}
1491 \maxchunks{5120}
1492

```

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

\l@dnumpstartsR 1493 \newcount\l@dnumpstartsR

1494

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

\l@pscR 1495 \newcount\l@dpScL

1496 \newcount\l@dpScR

1497

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

1498 \newcommand*{\l@dsetuprawboxes}{%

1499 \l@tempcntb=\l@dc@maxchunks

1500 \loop\ifnum\l@tempcntb>\z@

1501 \newnamebox{\l@dLcolrawbox\the\l@tempcntb}

1502 \newnamebox{\l@dRcolrawbox\the\l@tempcntb}

1503 \advance\l@tempcntb \m@ne

1504 \repeat}

1505

\l@dsetupmaxlinecounts To be able to synchronise left and right texts we need to know the maximum number
 \l@dzeromaxlinecounts 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.

1506 \newcommand*{\l@dsetupmaxlinecounts}{%

1507 \l@tempcntb=\l@dc@maxchunks

1508 \loop\ifnum\l@tempcntb>\z@

1509 \newnamecount{\l@dmaxlinesinpar\the\l@tempcntb}

1510 \advance\l@tempcntb \m@ne

1511 \repeat}

1512 \newcommand*{\l@dzeromaxlinecounts}{%

1513 \begingroup

1514 \l@tempcntb=\l@dc@maxchunks

1515 \loop\ifnum\l@tempcntb>\z@

1516 \global\usenamecount{\l@dmaxlinesinpar\the\l@tempcntb}=\z@

1517 \advance\l@tempcntb \m@ne

1518 \repeat

1519 \endgroup}

1520

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

1521 \AtBeginDocument{%

1522 \l@dsetuprawboxes

1523 \l@dsetupmaxlinecounts

1524 \l@dzeromaxlinecounts

1525 \l@dnumpstartsL=\z@

1526 \l@dnumpstartsR=\z@

1527 \l@dpScL=\z@

1528 \l@dpScR=\z@}

1529

XVII Checking text to be processed

```

\if@pstarts \check@pstarts returns \@pstartstrue if there are any unprocessed chunks.
\@pstartstrue 1530 \newif\if@pstarts
\@pstartsfalse 1531 \newcommand*{\check@pstarts}{%
\check@pstarts 1532 \@pstartsfalse
1533 \ifnum\l@dnumpstartsL>\l@dpscL
1534 \@pstartstrue
1535 \else
1536 \ifnum\l@dnumpstartsR>\l@dpscR
1537 \@pstartstrue
1538 \fi
1539 \fi
1540 }
1541

```

```

\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 1542 \newif\ifaraw@text
1543 \newcommand*{\checkraw@text}{%
1544 \araw@textfalse
1545 \ifvbox\namebox{l@dLcolrawbox\the\l@dpscL}
1546 \araw@texttrue
1547 \else
1548 \ifvbox\namebox{l@dRcolrawbox\the\l@dpscR}
1549 \araw@texttrue
1550 \fi
1551 \fi
1552 }
1553

```

```

\@writelinesinparL These write the number of text lines in a chunk to the section files, and then afterwards
\@writelinesinparR zero the counter.
1554 \newcommand*{\@writelinesinparL}{%
1555 \edef\next{%
1556 \write\linenum@out{\string\@pend[\the\@donereallinesL]}}%
1557 \next
1558 \global\@donereallinesL \z@}
1559 \newcommand*{\@writelinesinparR}{%
1560 \edef\next{%
1561 \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}%
1562 \next
1563 \global\@donereallinesR \z@}
1564

```

XVIII Parallel columns

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

`\@eledsectionR` 1565 `\newsavebox{\@eledsectionL}%`

1566 `\newsavebox{\@eledsectionR}%`

`\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.

1567 `\newcommand*{\Columns}{%`

1568 `\ifl@dpairing%`

1569 `\led@err@Columns@InsideEnv%`

1570 `\fi%`

1571 `\l@dprintingcolumnstrue%`

1572 `\eledsection@correcting@skip=-\baselineskip%` Correction for sections' titles

1573 `\ifnum\l@dnumpstartsL=\l@dnumpstartsR\else`

1574 `\led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%`

1575 `\fi`

Start a group and zero counters, etc.

1576 `\begingroup`

1577 `\l@dzero penalties`

1578 `\endgraf\global\num@lines=\prevgraf`

1579 `\global\num@linesR=\prevgraf`

1580 `\global\par@line=\z@`

1581 `\global\par@lineR=\z@`

1582 `\global\l@dpscL=\z@`

1583 `\global\l@dpscR=\z@`

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

1584 `\check@pstarts`

1585 `\loop\if@pstarts`

1586 `\global\pstartnumtrue`

1587 `\global\pstartnumRtrue`

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.

1588 `\global\advance\l@dpscL \@ne`

1589 `\global\advance\l@dpscR \@ne`

1590 `\restore@pstartL@pc%`

1591 `\restore@pstartR@pc%`

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

1592 `\Columns@print@before@pstart%`

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

1593 `\checkraw@text`

1594 `{ \loop\ifaraw@text`

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

```

1595         \l@duselanguage{\theledlanguageL}%
1596         \do@lineL
1597         \xifinlist{\the\l@dpscL}{\eled@sections@@}
1598         {%
1599         \ifdefstring{\@eledsectmark}{L}%
1600         {\csuse{eled@sectmark@\the\l@dpscL}%
1601         }{}%
1602         \global\csundef{eled@sectmark@\the\l@dpscL}%
1603         \savebox{\@eledsectionL}{\parbox[t][t]{\Lcolwidth}{\vbox{\print@eled
> prevent alignment troubles with RTL language
1604         }%
1605         }{}%
1606         \l@duselanguage{\theledlanguageR}%
1607         \do@lineR
1608         \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}
1609         {%
1610         \ifdefstring{\@eledsectmark}{R}%
1611         {\csuse{eled@sectmark@\the\l@dpscR R}%
1612         }{}%
1613         \global\csundef{eled@sectmark@\the\l@dpscR R}%
1614         \savebox{\@eledsectionR}{\parbox[t][t]{\Rcolwidth}{\vbox{\print@eled
> prevent alignment troubles with RTL language
1615         }{}%
1616         \hb@xt@ \hsize{%
1617         \ifdefstring{\columns@position}{L}{\hfill}%
1618         \unhbox\l@dleftbox%
1619         \ifhbox\@eledsectionL%
1620         \usebox{\@eledsectionL}%
1621         \fi%
1622         \print@columnseparator%
1623         \unhbox\l@drightbox%
1624         \ifhbox\@eledsectionR%
1625         \usebox{\@eledsectionR}%
1626         \fi%
1627         \ifdefstring{\columns@position}{R}{\hfill}%
1628         }%
1629         \checkraw@text
1630         \checkverseL
1631         \checkverseR
1632         \checkpb@columns
1633         \repeat}

```

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.

```

1634         \@writelinesinparL
1635         \@writelinesinparR
1636         \check@pstarts

```

```

1637         \ifbypstart@%
1638             \write\linenum@out{\string\@set[1]}
1639             \resetprevline@
1640         \fi
1641     \ifbypstart@R
1642         \write\linenum@outR{\string\@set[1]}
1643         \resetprevline@
1644     \fi
1645     \Columns@print@after@pend%
1646 \repeat

```

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.

```

1647     \flush@notes
1648     \flush@notesR
1649 \endgroup
1650 \global\l@dpscL=\z@
1651 \global\l@dpscR=\z@
1652 \global\l@dnumstartsL=\z@
1653 \global\l@dnumstartsR=\z@
1654 \l@dprintingcolumnsfalse%
1655 \ignorespaces
1656 \global\instanzaLfalse
1657 \global\instanzaRfalse}
1658

```

`\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).

```

1659 \def\print@columnseparator{%
1660     \ifdim\beforecolumnseparator<0pt%
1661         \hfill%
1662     \else%
1663         \hspace{\beforecolumnseparator}%
1664     \fi%
1665     \columnseparator%
1666     \ifdim\aftercolumnseparator<0pt%
1667         \hfill%
1668     \else%
1669         \hspace{\aftercolumnseparator}%
1670     \fi%
1671 }%
1672 %\end{macrocode}
1673 % \end{macro}
1674 % \begin{macro}{\checkpb@columns}
1675 % \cs{checkpb@columns} prevent or make pagebreaking in columns, de-
    pending of the use of \cs{ledpb} or \cs{lednopb}.
1676 % \begin{macrocode}
1677

```

```

1678 \newcommand{\checkpb@columns}{%
1679     \newif\if@pb
1680     \newif\if@nopb
1681     \IfStrEq{\led@pb@setting}{before}{
1682         \numdef{\next@absline}{\the\absline@num+1}%
1683         \numdef{\next@abslineR}{\the\absline@numR+1}%
1684         \xifinlistcs{\next@absline}{l@prev@pb}{\@pbtrue}{}%
1685         \xifinlistcs{\next@abslineR}{l@prev@pbR}{\@pbtrue}{}
1686         \xifinlistcs{\next@absline}{l@prev@nopb}{\@nopbtrue}{}%
1687         \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\@nopbtrue}{}
1688     }{}
1689     \IfStrEq{\led@pb@setting}{after}{
1690         \xifinlistcs{\the\absline@num}{l@prev@pb}{\@pbtrue}{}%
1691         \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\@pbtrue}{}
1692         \xifinlistcs{\the\absline@num}{l@prev@nopb}{\@nopbtrue}{}%
1693         \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\@nopbtrue}{}
1694     }{}
1695 \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
1696 \if@pb\pagebreak[4]\fi
1697 }

```

`\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).

```

1698 \newcommand*{\columnseparator}{%
1699     \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}}
1700 \newdimen\columnrulewidth
1701 \columnrulewidth=\z@
1702

```

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

```

\columns@position 1703 \newcommand*{\columnspan}{1}{%
1704     \xdef\columns@position{#1}%
1705 }%
1706 \xdef\columns@position{R}%

```

`\beforecolumnseparator` and `\aftercolumnseparator` lengths are defined to -1pt. If user changes them to a positive length, the lengths are used to define blank spaces before / after the column separator, instead of `\hfill`.

```

1707 \newlength{\beforecolumnseparator}%
1708 \setlength{\beforecolumnseparator}{-2pt}%
1709
1710 \newlength{\aftercolumnseparator}%
1711 \setlength{\aftercolumnseparator}{-2pt}%
1712

```

`\widthliketwocolumns@L`, `\widthliketwocolumns@C`, `\widthliketwocolumns@R` The `\setwidth...` macros are called in `\beginnumbering` in a **non-parallel** typesetting context, to fix the width of the lines to be vertically aligned with parallel

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

```

1713 \newcommand{\setwidthliketwocolumns@L}{%
1714 % Temporary dimension, initially equal to the standard hsize, i.e. text width
1715 %     \begin{macrocode}
1716     \newdimen\temp%
1717     \temp=\hsize%

    Hsize : Left + Right width
1718     \hsize=\Lcolwidth%
1719     \advance\hsize\Rcolwidth%

    Now, calculating the remaining space
1720     \advance\temp-\hsize%

    And multiply the hsize by 2/3 of this space
1721     \multiply\temp by 2%
1722     \divide\temp by 3%
1723     \advance\hsize\temp%
1724 }%
1725
1726 \newcommand{\setpositionliketwocolumns@L}{%
1727     \renewcommand{\ledrlfill}{\hfill}%
1728 }%
1729
1730 \newcommand{\setnotespositionliketwocolumns@L}{%
1731 }%
1732
1733
1734 \newcommand{\setwidthliketwocolumns@C}{%
1735 % Temporary dimension, initially equal to the standard hsize, i.e. text width
1736     \newdimen\temp%
1737     \temp=\hsize%
1738 % Hsize : Left + Right width
1739     \hsize=\Lcolwidth%
1740     \advance\hsize\Rcolwidth%
1741 % Now, calculating the remaining space
1742     \advance\temp-\hsize%

    And multiply the hsize by 1/2 of this space
1743     \divide\temp by 2%
1744     \advance\hsize\temp%
1745 }%
1746
1747 \newcommand{\setpositionliketwocolumns@C}{%
1748     \doinsidelinehook{\hfill}%

```



```

1749 \renewcommand{\ledrlfill}{\hfill}%
1750 }%
1751
1752 \newcommand{\setnotespositionliketwocolumns@C}{%
1753 \newdimen\temp%
1754 \newdimen\tempa%
1755 \temp=\hsize%
1756 \tempa=\Lcolwidth%
1757 \advance\tempa\Rcolwidth%
1758 \advance\temp-\tempa%
1759 \divide\temp by 2%
1760 \leftskip=\temp%
1761 \rightskip=-\temp%
1762 }%
1763
1764 \newcommand{\setwidthliketwocolumns@R}{%
    Temporary dimension, initially equal to the standard hsize, i.e. text width
1765 \newdimen\temp%
1766 \temp=\hsize%
    Hsize : Left + Right width
1767 \hsize=\Lcolwidth%
1768 \advance\hsize\Rcolwidth%
    Now, calculating the remaining space
1769 \advance\temp-\hsize%
    And multiply the hsize by 2/3 of this space
1770 \multiply\temp by 2%
1771 \divide\temp by 3%
1772 \advance\hsize\temp%
1773 }%
1774
1775 \newcommand{\setpositionliketwocolumns@R}{%
1776 \doinsidelinehook{\hfill}%
1777 }%
1778
1779 \newcommand{\setnotespositionliketwocolumns@R}{%
1780 \newdimen\temp%
1781 \newdimen\tempa%
1782 \temp=\hsize%
1783 \tempa=\Lcolwidth%
1784 \advance\tempa\Rcolwidth%
1785 \advance\temp-\tempa%
1786 \divide\temp by 2%
1787 \leftskip=\temp%
1788 \rightskip=-\temp%
1789 }%
1790

```

`\Columns@print@before@pstart` The `\Columns@print@before@pstart` and `\Columns@print@after@pend`
`\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.

```

1791 \newcommand{\Columns@print@before@pstart}{%
1792   \ifboolexpr{%
1793     test{\ifcsstring{before@pstartL@the\l@dpscL}{\at@every@pstart}}%
1794     and test {\ifcsstring{before@pstartR@the\l@dpscR}{\at@every@pstart}}}%
1795     and test {\ifdefempty{\at@every@pstart}}}%
1796     {}%
1797     {%
1798       \hb@xt@ \hsize{%
1799         \ifdefstring{\columns@position}{L}{}{\hfill }%
1800         \par\parbox[t][t]{\Lcolwidth}{%
1801           \csuse{before@pstartL@the\l@dpscL}%
1802         }%
1803         \print@columnseparator%
1804         \parbox[t][t]{\Rcolwidth}{%
1805           \csuse{before@pstartR@the\l@dpscR}%
1806         }%
1807         \ifdefstring{\columns@position}{R}{}{\hfill}%
1808       }%
1809     }%
1810     \global\csundef{before@pstartL@the\l@dpscL}%
1811     \global\csundef{before@pstartR@the\l@dpscR}%
1812 }%
1813 \newcommand{\Columns@print@after@pend}{%
1814   \ifboolexpr{%
1815     test{\ifcsstring{after@pendL@the\l@dpscL}{\at@every@pend}}%
1816     and test {\ifcsstring{after@pendR@the\l@dpscR}{\at@every@pend}}}%
1817     and test {\ifdefempty{\at@every@pend}}}%
1818     {}%
1819     {%
1820       \hb@xt@ \hsize{%
1821         \ifdefstring{\columns@position}{L}{}{\hfill }%
1822         \parbox[t][t]{\Lcolwidth}{%
1823           \csuse{after@pendL@the\l@dpscL}%
1824         }%
1825         \print@columnseparator%
1826         \parbox[t][t]{\Rcolwidth}{%
1827           \csuse{after@pendR@the\l@dpscR}%
1828         }%
1829         \ifdefstring{\columns@position}{R}{}{\hfill}%
1830       }%
1831     }%
1832     \global\csundef{after@pendL@the\l@dpscL}%
1833     \global\csundef{after@pendR@the\l@dpscR}%
1834 }%

```

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` 1835 `\newcount\numpagelinesL`
 1836 `\newcount\numpagelinesR`
 1837 `\newcount\l@dminpagelines`
 1838

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.

```
1839 \newcommand* {\Pages} {%
1840   \ifl@dpairing%
1841     \led@err@Pages@InsideEnv%
1842   \fi%
1843   \eledsection@correcting@skip=-2\baselineskip% line correct-
      ing for section titles.
1844   \parledgroup@notespacing@set@correction%
1845   \typeout{}%
1846   \typeout{***** PAGES *****}%
1847   \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else%
1848     \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
1849   \fi%
```

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

```
1850 \setlength{\Lcolwidth}{\textwidth}%
1851 \setlength{\Rcolwidth}{\textwidth}%
```

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

```
1852 \cleartol@devenpage%
1853 \l@dprintingpagetrue%
1854 \begingroup%
1855   \l@dzeropenalties%
1856   \endgraf\global\num@lines=\prevgraf%
1857   \global\num@linesR=\prevgraf%
1858   \global\par@line=\z@%
1859   \global\par@lineR=\z@%
1860   \global\l@dpscL=\z@%
1861   \global\l@dpscR=\z@%
1862   \writtenlinesLfalse%
1863   \writtenlinesRfalse%
```

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

```
1864 \let\print@Xnotes\print@Xnotes@forpages%
1865 \let\print@notesX\print@notesX@forpages%
```

Check if there are chunks to be processed.

```
1866 \check@pstarts%
1867 \loop\if@pstarts%
```

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

```
1868 \global\advance\l@dpscL \@ne%
1869 \global\advance\l@dpscR \@ne%
```

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

```
1870 \getlinesfromparlistL%
1871 \getlinesfromparlistR%
1872 \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}%
1873 {\usernamecount{l@dmaxlinesinpar\the\l@dpscL}}%
1874 \check@pstarts%
1875 \repeat%
```

Zero the counts again, ready for the next bit.

```
1876 \global\l@dpscL=\z@%
1877 \global\l@dpscR=\z@%
```

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

```
1878 \getlinesfrompagelistL%
1879 \getlinesfrompagelistR%
1880 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
1881 {\l@dminpagelines}%

```

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.

```
1882 \check@pstarts%
1883 \if@pstarts%
```

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

```
1884 \global\advance\l@dpscL \@ne%
1885 \global\advance\l@dpscR \@ne%
1886 \restore@pstartL@pc%
1887 \restore@pstartR@pc%
```

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

```
1888 \global\@donereallinesL=\z@%
1889 \global\@donetotallinesL=\z@%
1890 \global\@donereallinesR=\z@%
1891 \global\@donetotallinesR=\z@%
```

Start a loop over the boxes (chunks).

```
1892      \checkraw@text%
1893 %      \begingroup
1894 {      \loop\ifaraw@text%
```

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

```
1895      \checkpageL%
1896      \l@duselanguage{\theledlanguageL}%
1897 {      \loop\ifl@dsamepage%
```

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

```
1898      \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}%
1899      \csuse{before@pstartL@\the\l@dpscl}%
1900      \global\csundef{before@pstartL@\the\l@dpscl}%
1901      \do@lineL%
1902      \xifinlist{\the\l@dpscl}{\eled@sections@@}
1903      {\print@eledsectionL}%
1904      {}%
1905      \advance\numpagelinesL \@ne%
```

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

```
1906      \ifshiftedpstarts%
1907          \ifdim\ht\l@dleftbox>0pt\hb@xt@%
1908          \hsize{\ledstrutL\unhbox\l@dleftbox}%
1909          \else%
1910              \dimen0=\pagetotal%
1911              \advance\dimen0 by \baselineskip%
1912              \global\pagetotal=\dimen0%
1913          \fi%
1914      \else%
1915          \parledgroup@correction@notespacing{L}
1916          \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
1917      \fi%
```

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

```
1918      \get@nextboxL%
1919      \global\l@dskipversenumberfalse%
1920      \ifprint@last@after@pendL%
1921      \csuse{after@pendL@\the\l@dpscl}%
1922      \global\csundef{after@pendL@\the\l@dpscl}%

```

```

1923          \fi%
1924          \checkpageL%
1925          \checkverseL%
1926          \checkpbL%
1927          \repeat%

```

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.

```

1928          \ifl@dpagfull%
1929          \@writelinesonpageL{\the\numpagelinesL}%
1930          \else%
1931          \@writelinesonpageL{1000}%
1932          \fi%

```

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.

```

1933          \numpagelinesL \z@%
1934          \parledgroup@correction@notespacing@init%
1935          \clearl@dleftpage }%

```

Now do the same for the right text.

```

1936          \checkpageR%
1937          \l@duselanguage{\theledlanguageR}%
1938 {
1939          \loop\ifl@dsamepage%
1940          \initnumbering@sectcountR%
1941          \ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{}%
1942          \csuse{before@pstartR@\the\l@dpscR}%
1943          \global\csundef{before@pstartR@\the\l@dpscR}%
1944          \do@lineR%
1945          \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
1946          {\print@eledsectionR}%
1947          {}%
1948          \advance\numpagelinesR \@ne%
1949          \ifshiftedpstarts%
1950          \ifdim\ht\l@drightrightbox>0pt\hb@xt@%
1951          \hsize{\ledstrutR\unhbox\l@drightrightbox}%
1952          \else%
1953          \dimen0=\pagetotal%
1954          \advance\dimen0 by \baselineskip%
1955          \global\pagetotal=\dimen0%
1956          \fi%
1957          \else%
1958          \parledgroup@correction@notespacing{R}%
1959          \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightrightbox}%
1960          \fi%
1961          \get@nextboxR%
1962          \global\l@dskipversenumberRfalse%
1963          \ifprint@last@after@pendR%

```

```

1963             \csuse{after@pendR@the\l@dpscR}%
1964             \global\csundef{after@pendR@the\l@dpscR}%
1965             \fi%
1966             \checkpageR%
1967             \checkverseR%
1968             \checkpbR%
1969             \repeat%
1970             \ifl@dpagfull%
1971             \@writelinesonpageR{\the\numpagelinesR}%
1972             \else%
1973             \@writelinesonpageR{1000}%
1974             \fi%
1975             \numpagelinesR=\z@%
1976             \parledgroup@correction@notespacing@init%

The page is full, so move onto the next (left, odd) page and repeat left text processing.
1977             \clearl@drightpage}%

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.
1978             \checkraw@text%
1979             \ifaraw@text%
1980             \getlinesfrompagelistL%
1981             \getlinesfrompagelistR%
1982             \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
1983             {\l@dminpagelines}%
1984             \fi%
1985             \repeat}%

We have now output the text from all the chunks.
1986             \fi%

Make sure that there are no inserts hanging around.
1987             \flush@notes%
1988             \flush@notesR%
1989             \endgroup%

Zero counts ready for the next set of left/right text chunks. The boolean tests for stanza
are switched to false.
1990             \global\l@dpscL=\z@%
1991             \global\l@dpscR=\z@%
1992             \global\l@dnumpstartsL=\z@%
1993             \global\l@dnumpstartsR=\z@%
1994             \global\instanzaLfalse%
1995             \global\instanzaRfalse%
1996             \l@dprintingpagesfalse%
1997             \finish@Pages@notes%Needed to prevent final notes over-
lap line number
1998             \ignorespaces}
1999
2000

```

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.

```
2001 \newcommand{\finish@Pages@notes}{%
2002   \def\do##1{%
```

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

```
2003   \ifnocritical@%
2004     \global\newnamebox{##1footins}
2005   \fi
2006   \ifnofamiliar@%
2007     \global\newnamebox{footins##1}
2008   \fi
```

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

```
2009   \ifvoid\csuse{##1footins}%
2010     \ifvoid\csuse{footins##1}\else%
2011       \newpage\null%
2012       \listbreak%
2013     \fi%
2014   \else%
2015     \newpage\null%
2016     \listbreak%
2017   \fi%
2018 }%
2019 \dolistloop{\@series}%
2020 }%
```

XIX.4 Struts

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

```
\ledstrutR 2021 \newcommand*{\ledstrutL}{\strut}
2022 \newcommand*{\ledstrutR}{\strut}
2023
```

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.

```
2024 \providecommand{\cleartoevenpage}[1][\@empty]{%
2025   \clearpage
2026   \ifodd\c@page\hbox{##1}\clearpage\fi}
2027 \newcommand*{\cleartol@devenpage}{%
2028   \ifdim\pagetotal<\topskip% on an empty page
2029   \else
2030     \clearpage
```



```

2031 \fi
2032 \ifodd\c@page\hbox{}\clearpage\fi%
2033 }%

```

`\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`.

```

2034 \newcommand*{\clearl@dleftpage}{%
2035 \ifdim\pagetotal=0pt\hbox{}\fi%
2036 \newpage%
2037 \ifodd\c@page\else
2038 \led@err@LeftOnRightPage
2039 \hbox{}%
2040 \cleardoublepage
2041 \fi}
2042
2043 \newcommand*{\clearl@drighthouse}{%
2044 \ifdim\pagetotal=0pt\hbox{}\fi%
2045 \newpage%
2046 \ifodd\c@page
2047 \led@err@RightOnLeftPage
2048 \hbox{}%
2049 \cleartoevenpage
2050 \fi}
2051

```

XIX.6 Lines managing

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

```

\@cs@linesinparR
2052 \newcommand*{\getlinesfromparlistL}{%
2053 \ifx\linesinpar@listL\empty
2054 \gdef\@cs@linesinparL{0}%
2055 \else
2056 \gl@p\linesinpar@listL\to\@cs@linesinparL
2057 \fi}
2058 \newcommand*{\getlinesfromparlistR}{%
2059 \ifx\linesinpar@listR\empty
2060 \gdef\@cs@linesinparR{0}%
2061 \else
2062 \gl@p\linesinpar@listR\to\@cs@linesinparR
2063 \fi}
2064

```

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

`\l@dcalc@maxoftwo` $\{\langle num \rangle\} \{\langle num \rangle\} \{\langle count \rangle\}$ sets $\langle count \rangle$ to the maximum
`\l@dcalc@minoftwo` of the two $\langle num \rangle$.

```

2085 \newcommand*{\l@dcalc@maxoftwo}[3]{%
2086   \ifnum #2>#1\relax
2087     #3=#2\relax
2088   \else
2089     #3=#1\relax
2090   \fi}
2091 \newcommand*{\l@dcalc@minoftwo}[3]{%
2092   \ifnum #2<#1\relax
2093     #3=#2\relax
2094   \else
2095     #3=#1\relax
2096   \fi}
2097

```

<code>\ifl@dsamepage</code>	<code>\checkpageL</code> tests if the space and lines already taken on the page by text and foot-
<code>\l@dsamepagetrue</code>	notes is less than the constraints. If so, then <code>\ifl@dpagefull</code> is set FALSE and
<code>\l@dsamepagefalse</code>	
<code>\ifl@dpagefull</code>	
<code>\l@dpagefulltrue</code>	
<code>\l@dpagefullfalse</code>	
<code>\checkpageL</code>	
<code>\checkpageR</code>	

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

```

2098 \newif\ifl@dsamepage
2099 \l@dsamepagetrue
2100 \newif\ifl@dpagefull
2101
2102 \newcommand*{\checkpageL}{%
2103   \l@dpagefulltrue
2104   \l@dsamepagetrue
2105   \check@goal
2106   \ifdim\pagetotal<\ledthegoal
2107     \ifnum\numpagelinesL<\l@dmnpagelines
2108       \else
2109         \l@dsamepagefalse
2110         \l@dpagefullfalse
2111       \fi
2112   \else
2113     \l@dsamepagefalse
2114     \l@dpagefulltrue
2115   \fi%
2116   \ifprint@last@after@pendL%
2117     \l@dpagefullfalse%
2118     \l@dsamepagefalse%
2119     \print@last@after@pendLfalse%
2120   \fi%
2121 }%
2122
2123 \newcommand*{\checkpageR}{%
2124   \l@dpagefulltrue
2125   \l@dsamepagetrue
2126   \check@goal
2127   \ifdim\pagetotal<\ledthegoal
2128     \ifnum\numpagelinesR<\l@dmnpagelines
2129       \else
2130         \l@dsamepagefalse
2131         \l@dpagefullfalse
2132       \fi
2133   \else
2134     \l@dsamepagefalse
2135     \l@dpagefulltrue
2136   \fi%
2137   \ifprint@last@after@pendR%
2138     \l@dpagefullfalse%
2139     \l@dsamepagefalse%
2140     \print@last@after@pendRfalse%
2141   \fi%
2142 }%

```

2143

`\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`.

```

2144 \newcommand{\checkpbL}{
2145   \IfStrEq{\led@pb@setting}{after}{
2146     \xifinlistcs{\the\absline@num}{l@prev@pb}{\l@dpagfulltrue\l@dsamepage
2147     \xifinlistcs{\the\absline@num}{l@prev@nopb}{\l@dpagfullfalse\l@dsamepage
2148   }{}
2149   \IfStrEq{\led@pb@setting}{before}{
2150     \numdef{\next@absline}{\the\absline@num+1}
2151     \xifinlistcs{\next@absline}{l@prev@pb}{\l@dpagfulltrue\l@dsamepagefal
2152     \xifinlistcs{\next@absline}{l@prev@nopb}{\l@dpagfullfalse\l@dsamepage
2153   }{}
2154 }
2155
2156 \newcommand{\checkpbR}{
2157   \IfStrEq{\led@pb@setting}{after}{
2158     \xifinlistcs{\the\absline@numR}{l@prev@pbR}{\l@dpagfulltrue\l@dsamepa
2159     \xifinlistcs{\the\absline@numR}{l@prev@nopbR}{\l@dpagfullfalse\l@dsam
2160   }{}
2161   \IfStrEq{\led@pb@setting}{before}{
2162     \numdef{\next@abslineR}{\the\absline@numR+1}
2163     \xifinlistcs{\next@abslineR}{l@prev@pbR}{\l@dpagfulltrue\l@dsamepagef
2164     \xifinlistcs{\next@abslineR}{l@prev@nopbR}{\l@dpagfullfalse\l@dsamepa
2165   }{}
2166 }

```

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

```

2167 \newcommand{\checkverseL}{
2168   \ifinstanzaL
2169     \iflednopbinverse
2170       \ifinserthangingsymbol
2171         \numgdef{\prev@abslineverse}{\the\absline@num-1}
2172         \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{}
2173         \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesL<3\ledpbnum{\pr
2174       \fi
2175     \fi
2176   \fi
2177 }
2178 \newcommand{\checkverseR}{
2179   \ifinstanzaR
2180     \iflednopbinverse
2181       \ifinserthangingsymbolR
2182         \numgdef{\prev@abslineverse}{\the\absline@numR-1}
2183         \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{}
2184         \IfStrEq{\led@pb@setting}{before}{\ifnum\numpagelinesR<3\ledpbnum{\pr

```

```

2185     \fi
2186   \fi
2187 \fi
2188 }

\setgoalfraction \ledthegoal is the amount of space allowed to taken by text and footnotes on
\ledthegoal a page before a forced pagebreak. This can be controlled via \@goalfraction.
\goalfraction \ledthegoal is calculated via \check@goal.
\check@goal
2189 \newdimen\ledthegoal
2190 \ifshiftedpstarts
2191     \newcommand*{\@goalfraction}{0.95}
2192 \else
2193     \newcommand*{\@goalfraction}{0.9}
2194 \fi
2195
2196 \newcommand*{\check@goal}{%
2197     \ledthegoal=\@goalfraction\pagegoal}
2198 \newcommand{\setgoalfraction}[1]{%
2199     \xdef\@goalfraction{#1}%
2200 }
```

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

```

\ifwrittenlinesL
2201 \newif\ifwrittenlinesL
2202 \newif\ifwrittenlinesR
2203
```

XIX.8 Getting boxes content

\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.

```

2204 \newcommand*{\get@nextboxL}{%
2205     \ifvbox\namebox{l@dLcolrawbox\the\l@dpscL}% box is not empty

    The current box is not empty; do nothing.

2206     \else%                                box is empty

    The box is empty. Check if enough lines (real and blank) have been output.

2207     \ifnum\usernamecount{l@dmaxlinesinpar\the\l@dpscL}>\@donetotallinesL
2208         \parledgroup@notes@endL
2209     \else

    Sufficient lines have been output.

2210     \ifnum\usernamecount{l@dmaxlinesinpar\the\l@dpscL}=\@donetotallinesL
2211         \parledgroup@notes@endL
2212     \fi
2213     \ifwrittenlinesL\else

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

2214         \@writelinesinparL
2215         \writtenlinesLtrue
```

```

2216      \fi
2217      \ifnum\l@dnumstartsL>\l@dpscL
    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.
2218      \writtenlinesLfalse
2219      \ifbypstart@
2220      \global\line@num=0%
2221      \resetprevline@%
2222      \fi
2223 % Add the content of the optional argument of the previ-
    ous \cs{pend}.
2224 % \begin{macrocode}
2225      \csuse{after@pendL@the\l@dpscL}%
2226      \global\csundef{after@pendL@the\l@dpscL}%
    Check the number of lines
2227      \l@dcalc@maxoftwo{\the\usernamecount{l@maxlinesinpar\the\l@dpscL}}%
2228                        {\the\@donetotallinesL}%
2229                        {\username-
    count{l@maxlinesinpar\the\l@dpscL}}%
2230      \global\@donetotallinesL \z@
    Go to the next pstart
2231      \global\advance\l@dpscL \@ne
2232      \global\pstartnumtrue%
2233      \restore@pstartL@pc%
    Add notes of parallel ledgroup.
2234      \parledgroup@notes@endL
2235      \parledgroup@correction@notespadding@final{L}
2236      \else
2237      \fi
2238      \fi
2239      \fi}
2240 \newcommand*{\get@nextboxR}{%
2241 \ifvbox\namebox{l@dRcolrawbox\the\l@dpscR}% box is not empty
2242 \else% box is empty
2243 \ifnum\usernamecount{l@maxlinesinpar\the\l@dpscR}>\@donetotallinesR
2244 \parledgroup@notes@endR
2245 \else
2246 \ifnum\usernamecount{l@maxlinesinpar\the\l@dpscR}=\@donetotallinesR
2247 \parledgroup@notes@endR
2248 \fi
2249 \ifwrittenlinesR\else
2250 \@writelinesinparR
2251 \writtenlinesRtrue
2252 \fi
2253 \ifnum\l@dnumstartsR>\l@dpscR

```

```

2254     \writtenlinesRfalse
2255     \ifbypstart@R
2256         \global\line@numR=0%
2257         \resetprevline@%
2258     \fi
2259     \csuse{after@pendR@the\l@dpscR}%
2260     \global\csundef{after@pendR@the\l@dpscR}%
2261     \l@dcalc@maxoftwo{\the\usernamecount{l@dmaxlinesinpar\the\l@dpscR}}%
2262         {\the\@donetotallinesR}%
2263                                     {\username-
count{l@dmaxlinesinpar\the\l@dpscR}}%
2264     \global\@donetotallinesR \z@
2265     \global\advance\l@dpscR \@ne
2266     \global\pstartnumRtrue%
2267     \restore@pstartR@pc%
2268     \parledgroup@notes@endR
2269     \parledgroup@correction@notes@spacing@final{R}
2270 \else
2271     \print@last@after@pendRtrue%
2272 \fi
2273 \fi
2274 \fi}
2275

```

XIX.9 Same page number in both side

The `sameparallelpagenumber` allow to have the same page number for the left and the right side. We can not do it by changing the value of the page counter, since its value is used by many \TeX 's features to determine whether a page is left or right: left page are even, right page are odd. Consequently, we have to do it by patching `\thepage`, in order to use the value of the `sameparallelpage@number` counter instead of value of page counter.

This counter will be increased in a patched version of the \TeX 's `\@outputpage` macro, as is the page counter in this macro. However, `\sameparallelpage@number` will be increased only one page on two when processing `\Pages`.

`\par@patch@thepage` `\par@patch@thepage` patches `\thepage` in order to use the value of `sameparallelpage@number` counter and not the value of `sameparallelpage@number`. It must be called after any redefinition of `\thepage`. That why we insert it at the end of the \TeX macro `\pagenumbering`, which is called by some `\xxxmatter` commands. We also call it at the begin of the document.

```

2276 \newcommand{\par@patch@thepage}{%
2277   \ifsameparallelpagenumber%
2278     \patchcmd{\thepage}%
2279       {page}{sameparallelpage@number}%
2280       {}%
2281     {\led@error@fail@patch@thepage}%
2282   \fi%

```

```

2283 }%
2284 \apptocmd{\pagenumbering}{%
2285   \par@patch@thepage%
2286 }%
2287 {}%
2288 {\led@error@fail@patch@pagenumbering}%
2289
2290 \AtBeginDocument{\par@patch@thepage}%
2291

```

`\@outputpage` As its names said, `\@outputpage` is a \TeX 's macro called in the output routine. That is this macro which increase the page counter. We patch it in order to increase, conditionally, the `sameparallelpage@number` counter.

```

2292 \AtBeginDocument{%
2293   \apptocmd{\@outputpage}{%
2294     \ifsameparallelpagenumber%
2295       \ifl@dprintingpages%
2296         \ifodd\c@page\else%
2297           \stepcounter{sameparallelpage@number}%
2298         \fi%
2299       \else%
2300         \stepcounter{sameparallelpage@number}%
2301       \fi%
2302     \else%
2303       \stepcounter{sameparallelpage@number}%
2304     \fi%
2305   }%
2306   {}%
2307   {\led@error@fail@patch@@@outputpage}%
2308 }

```

`\thesameparallelpage@number` And now, initialize `sameparallelpage@number` counter.

```

2309 \newcounter{sameparallelpage@number}%
2310 \setcounter{sameparallelpage@number}{1}%

```

XX 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 mis-aligned.

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

`\chapter`

```

2311 \pretocmd{\chapter}{%
2312   \ifl@dprintingpages%
2313     \vbox{}%
2314   \fi%
2315 }%

```


2316 {}%
2317 {}%

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

2318 `\newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}}`
2319 `\eledsectnotoc{R}`

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

2320 `\newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}}`
2321 `\eledsectmark{L}`

`\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`.

2322 `\newskip\eledsection@correcting@skip`

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

2323 `\newwrite\eled@sectioningR@out`

XXI 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).

2324 `\def\l@prev@pbR{}`
2325 `\def\l@prev@nopbR{}`

`\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.

2326 `\newcommand{\ledpbR}{\write\linenum@outR{\string\led@pbR}}`
2327 `\newcommand{\ledpbnumR}[1]{\write\linenum@outR{\string\led@pbnumR{#1}}}`
2328 `\newcommand{\lednopbR}{\write\linenum@outR{\string\led@nopbR}}`
2329 `\newcommand{\lednopbnumR}[1]{\write\linenum@outR{\string\led@nopbnumR{#1}}}`

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

```

2330 \newcommand{\led@pbR}{\listxadd{\l@prev@pbR}{\the\absline@numR}}
2331 \newcommand{\led@pbnumR}[1]{\listxadd{\l@prev@pbR}{#1}}
2332 \newcommand{\led@nopbR}{\listxadd{\l@prev@nopbR}{\the\absline@numR}}
2333 \newcommand{\led@nopbnumR}[1]{\listxadd{\l@prev@nopbR}{#1}}

```

XXII Parallel ledgroup

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

```

2334 \newmarks\parledgroup@
2335 \newmarks\parledgroup@series
2336 \newmarks\parledgroup@type

```

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

```

2337 \newcommand{\parledgroup@notes@startL}{%
2338   \ifnum\usernamecount{\l@maxlinesinpar\the\l@dpscL}>0%
2339   \IfStrEq{\splitfirstmarks\parledgroup@type}{footnotex}{\csuse{bhooknotex}}
2340   \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{bhookXnotex}}
2341   \fi%
2342   \global\ledgroupnotesL@true%
2343   \insert@noterule@ledgroup{L}%
2344 }
2345 \newcommand{\parledgroup@notes@startR}{%
2346   \ifnum\usernamecount{\l@maxlinesinpar\the\l@dpscR}>0%
2347   \IfStrEq{\splitfirstmarks\parledgroup@type}{footnotex}{\csuse{bhooknotex}}
2348   \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{bhookXnotex}}
2349   \fi%
2350   \global\ledgroupnotesR@true%
2351   \insert@noterule@ledgroup{R}%
2352 }

```

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

```

2353 \newcommand{\parledgroup@notes@endL}{%
2354   \global\ledgroupnotesL@false%
2355 }
2356 \newcommand{\parledgroup@notes@endR}{%
2357   \global\ledgroupnotesR@false%
2358 }

```

`\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`

```

2359 \newcommand{\insert@noterule@ledgroup}[1]{
2360   \IfStrEq{\splitbotmarks\parledgroup@}{begin}{%

```

```

2361 \IfStrEq{\splitbotmarks\parledgroup@type}{Xfootnote}{
2362 \csuse{ifledgroupnotes#1@}
2363 \vskip\skip\csuse{mp\splitbotmarks\parledgroup@series footins}
2364 \csuse{\splitbotmarks\parledgroup@series foot-
noterule}
2365 \fi
2366 }
2367 {}
2368 \IfStrEq{\splitbotmarks\parledgroup@type}{footnotex}{
2369 \csuse{ifledgroupnotes#1@}
2370 \vskip\skip\csuse{mpfootins\splitbotmarks\parledgroup@series}
2371 \csuse{footnoterule\splitbotmarks\parledgroup@series}
2372 \fi
2373 }{}
2374 }
2375 {}
2376 }

```

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

```

2377 \newcommand{\setparledgroupnotespacing}[1]{\gdef\@parledgroupnotespacing{#1}}
2378 \newcommand{\@parledgroupnotespacing}{}

```

`\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`.

```

2379 \dimdef{\parledgroup@notespacing@correction}{0pt}
2380 \newcommand{\parledgroup@notespacing@set@correction}{%
2381 {\@getfirstseries\csuse{Xnotefontsize@\@firstseries}%We suppose
all the series has the same footnote size setup
2382 \@parledgroupnotespacing\dimgdef{\temp@spacing}{\baselineskip}}%
2383 \dimgdef{\parledgroup@notespacing@correction}{\baselineskip-
\temp@spacing}%
2384 }

```

`\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.

```

2385 \newcommand{\parledgroup@correction@notespacing@init}{
2386 \dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}
2387 \dimdef{\parledgroup@notespacing@correction@modulo}{0pt}
2388 }
2389 \parledgroup@correction@notespacing@init

```

`\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`.

```

2390 \newcommand{\parledgroup@correction@notespacing@final}[1]{
2391     \ifparledgroup
2392     \vspace{\parledgroup@notespacing@correction@accumulated}
2393     \parledgroup@correction@notespacing@init%
2394     \ifstrequal{#1}{L}{
2395         \numdef{\@checking}{\the\l@dpscL-1}
2396     }{
2397         \numdef{\@checking}{\the\l@dpscR-1}
2398     }
2399     \dimdef{\@beforenotes@current@diff}{\csuse{@parledgroup@beforenotes@\@c
\csuse{@parledgroup@beforenotes@\@checking R}}%
2400     \ifstrequal{#1}{L}%
2401     {% Left
2402     \ifdimgreater{\@beforenotes@current@diff}{0pt}{}{\vspace{-
\@beforenotes@current@diff}}%
2403     }%
2404     {% Right
2405     \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{\@beforenotes@
2406     }%
2407     \fi
2408 }

```

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

```

2409 {}
2410 \newcommand{\parledgroup@correction@notespacing}[1]{%
2411     \csuse{ifledgroupnotes#1@}%
2412     \vspace{-\parledgroup@notespacing@correction}%
2413     \dimdef{\parledgroup@notespacing@correction@accumulated}{\parledgroup@
2414     \dimdef{\parledgroup@notespacing@correction@modulo}{\parledgroup@notes
2415     \ifdimless{\parledgroup@notespacing@correction@modulo}{\baselineskip}
\@ne%
2416     \dimdef{\parledgroup@notespacing@correction@modulo}{\parledgroup@notes
\baselineskip}%
2417     }% mean greater than equal
2418     \fi%
2419 }

```

```

\parledgroup@beforenotesL and \parledgroup@beforenotesR store
\parledgroup@beforenotesR the total of space before notes in the current parallel ledgroup.
2420 \dimdef\parledgroup@beforenotesL{0pt}
2421 \dimdef\parledgroup@beforenotesR{0pt}

\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.
2422 \newcommand{\parledgroup@beforenotes@save}[1]{
2423   \ifparledgroup
2424     \csdimgdef{@parledgroup@beforenotes@\the\csuse{l@dnumpstarts#1}#1}{\csuse{parled
2425       \csdimgdef{parledgroup@beforenotes#1}{0pt}
2426     \fi
2427 }

```

XXIII Compatibility with eledmac

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

```

2428 \ifeledmaccompat@%
2429
2430
2431   \unless\ifnocritical@
2432     \let\onlyXside\Xonlyside
2433   \fi
2434 \fi

```

XXIV 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 `\setstanzaindents` command.

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

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

See the following two examples:

With standard `\hangingsymbol`:

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

With the modification of the `hangingsymbol`:

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

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>

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

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

<code>\&</code>	1389, 1390, 1393, 1409, 1431
<code>\@RTLtrue</code>	810
<code>\@Rlineflag</code>	221, 234, 1208, 1212, 1310
<code>\@adv</code>	312
<code>\@arabic</code>	224, 225, 582, 584
<code>\@astanza@line</code>	1408, 1411, <u>1418</u>
<code>\@beforenotes@current@diff</code>	2399, 2402, 2405
<code>\@checking</code>	2395, 2397, 2399
<code>\@cs@linesinparL</code>	1872, <u>2052</u>
<code>\@cs@linesinparR</code>	1872, <u>2052</u>
<code>\@cs@linesonpageL</code>	1880, 1982, <u>2065</u>
<code>\@cs@linesonpageR</code>	1880, 1982, <u>2065</u>
<code>\@currentlabel</code>	629, 678
<code>\@donereallinesL</code>	<u>740</u> , 775, 1556, 1558, 1888
<code>\@donereallinesR</code>	<u>740</u> , 868, 1561, 1563, 1890
<code>\@donetotallinesL</code>	<u>740</u> , 776, 779, 1889, 2207, 2210, 2228, 2230
<code>\@donetotallinesR</code>	<u>740</u> , 869, 872, 1891, 2243, 2246, 2262, 2264
<code>\@edtext@level</code>	360, 385–387, 392–396, 398, 400
<code>\@eled@sectioningfalse</code>	814
<code>\@eled@sectioningtrue</code>	804

\@eledsectionL	1565, 1603, 1619, 1620
\@eledsectionR	1565, 1614, 1624, 1625
\@eledsectmark	801, 1599, 1610, 2320
\@eledsectnotoc	800, 1898, 1940, 2318
\@firstseries	2381
\@getfirstseries	2381
\@goalfraction	2191, 2193, 2197, 2199
\@gobble	368
\@gobblethree	374
\@hangingsymbol	1376, 1385
\@insertR	1185–1187, 1200–1202
\@l@dttempcnta	930, 974, 975, 977, 979, 982, 983, 1000–1004, 1006, 1013, 1018, 1022, 1030, 1035, 1039, 1072, 1075, 1077, 1081
\@l@dttempcntb	965, 966, 1013, 1018, 1022, 1030, 1035, 1039, 1064, 1068, 1081, 1090–1092, 1094, 1116–1118, 1120, 1137–1139, 1141, 1318, 1319, 1347–1349, 1351, 1499–1503, 1507–1510, 1514–1517
\@lab	372, 1316
\@lemma	373
\@lock	758, 908
\@lockR	64, 286, 288, 290, 303, 312, 328, 329, 331, 332, 350, 351, 353, 850, 889, 936, 938, 939, 941, 1027, 1044, 1046, 1048
\@lopL	409, 2079
\@lopR	368, 409, 2082
\@nl	264, 433, 435
\@nl@regR	264
\@nobreakfalse	590, 640, 803
\@nobreaktrue	588, 592, 638, 642, 803
\@nopbtrue	1686, 1687, 1692, 1693
\@oldnobreak	588, 590, 638, 640, 696, 717
\@outputpage	43, 2292
\@parledgroupnotespacing	2377, 2382
\@pbtrue	1684, 1685, 1690, 1691
\@pend	402, 1556
\@pendR	402, 1561
\@pstartfalse	1530
\@pstarttrue	1530
\@ref	358
\@ref@regR	358
\@series	481, 2019
\@set	312, 1638, 1642
\@startstanza	544, 545, 570, 571
\@stopastanza	1409, 1414
\@sw	374
\@temp	498, 499, 504, 505
\@templ@d	1337, 1339
\@templ@n	1338, 1339
\@thepage	37
\@tmp	395, 397, 398, 453, 455, 456, 459–461
\@tmpa	454, 455, 460, 463
\@tmpp	396, 397

\@writelinesinparL	1554, 1634, 2214
\@writelinesinparR	1554, 1635, 2250
\@writelinesonpageL	1929, 1931, 2078
\@writelinesonpageR	1971, 1973, 2078
\@xloop	1198

A

\absline@num	898, 1682, 1690, 1692, 2146, 2147, 2150, 2171
\absline@numR	59, 241, 266, 269, 272, 316, 340, 363, 879, 920, 921, 965, 1184, 1683, 1691, 1693, 2158, 2159, 2162, 2182, 2330, 2332
\actionlines@listR	245, 259, 316, 340, 987, 990
\actions@listR	245, 260, 318, 327, 342, 349, 991
\add@inserts	772, 786
\add@inserts@nextR	1173
\add@insertsR	864, 1173
\add@penaltiesL	774, 1194
\add@penaltiesR	867, 1194
\addtocounter	699, 720, 799
\advanceline	436
\affixline@num	769
\affixline@numR	861, 997
\affixpstart@numL	784, 1106
\affixpstart@numR	1106
\affixside@note	772, 786
\affixside@noteR	864, 1324
\aftercolumnseparator	8, 1666, 1707
\appto	1330, 1331
\apptocmd	2284, 2293
\araw@textfalse	1542
\araw@texttrue	1542
astanza (environment)	14, 1392
\at@begin@pairs	512, 517, 518
\at@every@pend	703, 724, 1815–1817
\at@every@pstart	632, 681, 1793–1795
\at@every@pstart@call	634, 683, 728, 729
\AtBeginDocument	1312, 1460, 1521, 2290, 2292
\AtBeginPairs	7, 517
\AtEveryPstartCall	728
\autopar	13

B

\ballast@count	918, 923
\bbl@main@language	1475, 1476
\bbl@set@language	1450, 1467
\beforecolumnseparator	8, 1660, 1663, 1669, 1707
\beginnumbering	11, 558, 600
\beginnumberingR	50, 134, 558, 650
\bfseries	582, 584
\boolfalse	387, 1236, 1280
\booltrue	1239, 1283

\bypage@Rfalse	169, 177
\bypage@Rtrue	161
\bypstart@Rfalse	162, 178
\bypstart@Rtrue	170

C

\c@ballast	923
\c@chapter	115
\c@chapterR	115
\c@firstlinenumR	<u>195</u> , 1070
\c@firstsublinenumR	<u>199</u> , 1065
\c@linenumincrementR	<u>195</u> , 1070
\c@page	433, 435, 1224, 1226, 1268, 1270, 2026, 2032, 2037, 2046, 2296
\c@pstartL	487, 499, 582
\c@pstartR	492, 505, 584
\c@section	116
\c@sectionR	116
\c@sublinenumincrementR	<u>199</u> , 1065
\c@subsection	117
\c@subsectionR	117
\c@subsubsection	118
\c@subsubsectionR	118
\ch@ck@l@ckR	<u>997</u>
\ch@cksub@l@ckR	<u>997</u>
\ch@cksub@lockR	1066
\chapter	<u>2311</u>
\chapterinpages	<u>508</u>
\chardef	1389
\check@goal	2105, 2126, <u>2189</u>
\check@pstarts	<u>1530</u> , 1584, 1636, 1866, 1874, 1882
\checkpageL	1895, 1924, <u>2098</u>
\checkpageR	1936, 1966, <u>2098</u>
\checkpb@columns	1632, 1674, 1678
\checkpbL	1926, <u>2144</u>
\checkpbR	1968, <u>2144</u>
\checkraw@text	<u>1542</u> , 1593, 1629, 1892, 1978
\checkverseL	1630, 1925, <u>2167</u>
\checkverseR	1631, 1967, <u>2167</u>
\cleardoublepage	2040
\clearl@dleftpage	1935, <u>2034</u>
\clearl@drighthpage	1977, <u>2034</u>
\cleartoevenpage	<u>2024</u> , 2049
\cleartol@devenpage	1852, <u>2024</u>
\closeout	106, 420, 428
\columnrulewidth	7, <u>1698</u>
\Columns	6, 29, 31, 35, <u>1567</u>
\columns@position	1617, 1627, <u>1703</u> , 1799, 1807, 1821, 1829
\Columns@print@after@pend	1645, <u>1791</u>
\Columns@print@before@pstart	1592, <u>1791</u>
\columnseparator	7, 1665, <u>1698</u>

<code>\columnspan</code>	7, 1703
<code>\correct@footinsX@box</code>	1217
<code>\correct@Xfootins@box</code>	1217
<code>\count</code>	1232, 1240, 1276, 1284
<code>\countLline</code>	735 , 747
<code>\countRline</code>	735 , 839
<code>\create@this@edtext@level</code>	393, 394
<code>\critext</code>	436
<code>\cs</code>	389, 390, 1246, 1675, 2223
<code>\csdimgdef</code>	2424, 2425
<code>\csgdef</code>	438, 443, 632, 633, 681, 682, 703, 704, 724, 725
<code>\cslet</code>	398, 456, 461, 488, 493
<code>\csundef</code>	815, 1602, 1613, 1810, 1811, 1832, 1833, 1900, 1922, 1942, 1964, 2226, 2260
<code>\csuse</code>	451, 454, 464, 465, 812, 1219, 1220, 1230–1235, 1240, 1241, 1243, 1244, 1252, 1254, 1255, 1257, 1263, 1264, 1274–1279, 1284, 1285, 1287, 1288, 1295, 1297, 1298, 1300, 1600, 1611, 1801, 1805, 1823, 1827, 1899, 1921, 1941, 1963, 2009, 2010, 2225, 2259, 2339, 2340, 2347, 2348, 2362–2364, 2369–2371, 2381, 2399, 2411, 2424

D

<code>\DeclareOption</code>	7–9, 11
<code>\def@tempb</code>	175
<code>\dimdef</code>	2379, 2386, 2387, 2399, 2413, 2414, 2416, 2420, 2421
<code>\dimen</code>	1235, 1244, 1279, 1288, 1910–1912, 1952–1954
<code>\dimexpr</code>	1255, 1256, 1298, 1299
<code>\dingdef</code>	2382, 2383
<code>\divide</code>	1002, 1722, 1743, 1759, 1771, 1786
<code>\do@actions</code>	899
<code>\do@actions@fixedcodeR</code>	927
<code>\do@actions@nextR</code>	927
<code>\do@actionsR</code>	880, 927
<code>\do@ballast</code>	900
<code>\do@ballastR</code>	881, 918
<code>\do@insidelineLhook</code>	789, 828, 831
<code>\do@insidelineRhook</code>	829, 831
<code>\do@lineL</code>	745 , 1596, 1901
<code>\do@lineLhook</code>	751, 826, 831
<code>\do@lineR</code>	836 , 1607, 1943
<code>\do@lineRhook</code>	827, 831 , 843
<code>\do@lockoff</code>	337
<code>\do@lockoffR</code>	337
<code>\do@lockon</code>	312
<code>\do@lockonR</code>	312
<code>\doinsidelinehook</code>	1748, 1776
<code>\doinsidelineLhook</code>	826
<code>\doinsidelineRhook</code>	826
<code>\dolineLhook</code>	826
<code>\dolineRhook</code>	826
<code>\dolistloop</code>	481, 1335, 1355, 1362, 2019
<code>\dummy@ref</code>	367
<code>\dump@pstartL@pc</code>	486 , 697

\dump@pstartR@pc 486, 718

E

\edlabel 1306
 \edtext 436
 \eled@sectioningR@out 78, 106, 2323
 \eled@sections@@ 771, 803, 1597, 1902
 \eled@sectionsR@@ 75, 863, 1608, 1944
 \eledpar@error 17
 \eledsection@correcting@skip 823, 1572, 1843, 2322
 \eledsectmark 16, 2320
 \eledsectnotoc 16, 2318
 \empty 91, 94, 497, 503, 608, 658, 987, 1069, 1077, 1175–1177, 1188, 1199, 2053, 2059, 2066, 2072
 \endashchar 1210
 \endgraf 692, 713, 1578, 1856
 \endline@num 377, 383
 \endlock 436, 1397, 1405, 1415
 \endnumbering 11, 81, 138, 559
 \endnumberingR 53, 81, 121, 133, 147, 559
 \endpage@num 376, 383
 \endsub 436
 \endsubline@num 378, 384
 \enlargethispage 1695
 environments:
 astanza 14, 1392
 Leftside 11, 533
 pages 8, 508
 pairs 6, 508
 Rightside 11, 553
 \expandonce 397
 \extensionchars 70, 127, 143, 152

F

\f@x@l@cks 767
 \f@x@l@cksR 859, 997
 \finish@Pages@notes 1997, 2001
 \first@linenum@out@Rfalse 415, 421
 \first@linenum@out@Rtrue 415
 \firstlinenum 12, 204
 \firstlinenum* 12, 204
 \firstsublinenum 12, 204
 \firstsublinenum* 12, 204
 \fix@page 309
 \flag@end 436
 \flag@start 436
 \flush@notes 1647, 1987
 \flush@notesR 1197, 1648, 1988
 \footnoteXmk 10
 \footnoteXnomk 10
 \fullstop 237, 1207, 1209, 1211, 1213

G

<code>\get@nextboxL</code>	1918, 2204
<code>\get@nextboxR</code>	1960, 2204
<code>\getline@numL</code>	757, 897
<code>\getline@numR</code>	849, 878
<code>\getlinesfrompagelistL</code>	1878, 1980, 2065
<code>\getlinesfrompagelistR</code>	1879, 1981, 2065
<code>\getlinesfromparlistL</code>	1870, 2052
<code>\getlinesfromparlistR</code>	1871, 2052
<code>\gl@p</code>	460, 498, 504, 990, 991, 1181, 1185, 1200, 2056, 2062, 2069, 2075
<code>\goalfraction</code>	2189

H

<code>\hb@xt@</code>	764, 778, 788, 856, 871, 1616, 1798, 1820, 1907, 1916, 1949, 1958
<code>\hsize</code>	627, 676, 1616, 1717–1720, 1723, 1737, 1739, 1740, 1742, 1744, 1755, 1766–1769, 1772, 1782, 1798, 1820, 1908, 1916, 1950, 1958

I

<code>\if@firstcolumn</code>	1084, 1110, 1131, 1341
<code>\if@nobreak</code>	587, 637
<code>\if@noeled@sec</code>	76, 105
<code>\if@nopb</code>	1680, 1695
<code>\if@pb</code>	1679, 1696
<code>\if@pstarts</code>	1530 , 1585, 1867, 1883
<code>\if@RTL</code>	816
<code>\ifaraw@text</code>	1542 , 1594, 1894, 1979
<code>\ifautopar</code>	619, 669
<code>\ifbool</code>	1251, 1294
<code>\ifboolexpr</code>	1223, 1267, 1792, 1814
<code>\ifbypage@R</code>	969
<code>\ifbypstart@</code>	403, 1637, 2219
<code>\ifbypstart@R</code>	406, 1641, 2255
<code>\ifcsempy</code>	1218, 1262
<code>\ifcsstring</code>	1224, 1226, 1268, 1270, 1793, 1794, 1815, 1816
<code>\ifcsundef</code>	394
<code>\ifdefempty</code>	1795, 1817
<code>\ifdefstring</code> ...	800, 801, 810, 1599, 1610, 1617, 1627, 1799, 1807, 1821, 1829, 1898, 1940
<code>\ifdim</code>	1660, 1666, 1907, 1949, 2028, 2035, 2044, 2106, 2127
<code>\ifdimgreater</code>	2402, 2405
<code>\ifdimless</code>	2415
<code>\ifeledmaccompat@</code>	2428
<code>\iffirst@linenum@out@R</code>	415 , 419
<code>\ifhbox</code>	1619, 1624
<code>\ifinserthangingsymbol</code>	1372, 2170
<code>\ifinserthangingsymbolR</code>	1370, 1381, 2181
<code>\ifinstanzaL</code>	531, 531 , 1373, 2168
<code>\ifinstanzaR</code>	531 , 532, 1382, 2179
<code>\ifl@d@dash</code>	1210
<code>\ifl@d@elin</code>	1212, 1213
<code>\ifl@d@esl</code>	1213

\ifl@pnum	1207, 1211
\ifl@dssub	1209
\ifl@dhidenumber	765, 857
\ifl@dpagfull	1928, 1970, 2098
\ifl@dpaging	13
\ifl@dpairing	13, 88, 1568, 1840
\ifl@dprintingpages	2295, 2312
\ifl@dsamepage	1897, 1938, 2098
\ifl@dskipnumber	1060
\ifl@dskipversenumberR	928, 1082
\ifl@dusedbabel	1449
\iflabelpstart	629, 678
\ifledgroupnotesL@	901
\ifledgroupnotesR@	882, 1059
\iflednopbinverse	2169, 2180
\ifledplinenum	1208
\ifledRcol	13, 1470, 1482
\ifluatex	593, 643, 790, 806
\ifnocritical@	163, 171, 179, 441, 471, 2003, 2431
\ifnofamiliar@	437, 445, 474, 2006
\ifnoteschanged@	98
\ifnumberedpar@	602, 652, 689, 710
\ifnumbering	156, 598, 686
\ifnumberingR	51, 82, 123, 648, 707
\ifnumberline	883, 902, 1059
\ifnumberpstart	488, 493, 620, 670, 698, 719, 746, 837
\ifnumequal	1328
\ifnumgreater	1336, 1356, 1363
\ifnumodd	1224, 1226, 1268, 1270
\ifodd	1094, 1120, 1141, 1351, 2026, 2032, 2037, 2046, 2296
\ifparledgroup	2391, 2423
\ifprint@last@after@pendL	730, 1920, 2116
\ifprint@last@after@pendR	730, 1962, 2137
\ifpstartedL	47, 606
\ifpstartedR	47, 656
\ifpstartnum	1151, 1156
\ifpstartnumR	1106
\ifsameparallelpagenunder	10, 2277, 2294
\ifshiftedpstarts	6, 1906, 1948, 2190
\ifsidepstartnum	621, 671, 1108, 1129
\ifstrempy	631, 680, 702, 723
\IfStrEq	755, 847, 1681, 1689, 2145, 2149, 2157, 2161, 2172, 2173, 2183, 2184, 2339, 2340, 2347, 2348, 2360, 2361, 2368
\ifstrequal	2394, 2400
\ifsublines@	235, 298, 317, 341, 382, 384, 884, 903, 976, 1063
\ifvbox	748, 840, 1545, 1548, 2205, 2241
\ifvoid	534, 554, 2009, 2010
\ifwidthliketwocolumns	9
\ifwrittenlinesL	2201, 2213
\ifwrittenlinesR	2202, 2249

\init@series@par	479
\initnumbering@quote	511, 523
\initnumbering@sectcountR	74, <u>110</u> , 130, 1939
\InputIfFileExists	77
\insert@countR	358
\insert@noterule@ledgroup	2343, 2351, <u>2359</u>
\inserthangingsymbolfalse	761
\inserthangingsymbolL	794, <u>1370</u>
\inserthangingsymbolR	<u>1370</u>
\inserthangingsymbolRfalse	853
\inserthangingsymbolRtrue	851
\inserthangingsymboltrue	759
\insertlines@listR	91, <u>245</u> , 258, 363, 1177, 1181
\inserts@list	607
\inserts@listR	657, <u>1172</u> , 1175, 1185, 1199, 1200
\istanzaLfalse	1656, 1994
\istanzaLtrue	545
\istanzaRfalse	1657, 1995
\istanzaRtrue	571
\itemcount@	1326, 1328, 1333, 1336, 1354, 1356, 1361, 1363

L

\l@d@set	<u>312</u>
\l@dbfnote	<u>1370</u>
\l@dc@maxchunks	613, 615, 663, 665, <u>1489</u> , 1499, 1507, 1514
\l@dcal@maxoftwo	1872, <u>2085</u> , 2227, 2261
\l@dcal@minoftwo	1880, 1982, <u>2085</u>
\l@dcalcnum	<u>997</u>
\l@dchecklang	<u>1450</u>
\l@dchset@num	265, 268
\l@dcsnotetext	1335, 1338
\l@dcsnotetext@l	1338, 1355
\l@dcsnotetext@r	1338, 1362
\l@emptyd@ta	752, 844
\l@dend@stuff	71, 128, 144, 153
\l@dgetsidenote@margin	1317
\l@dhidenumberfalse	766, 858
\l@dhidenumbertrue	956
\l@dld@ta	785, 1085, 1097, 1111, 1123, 1132, 1144
\l@dleftbox	<u>732</u> , 763, 778, 1618, 1907, 1908, 1916
\l@dlinenumR	<u>227</u>
\l@dlsn@te	787
\l@dmake@labelsR	<u>1306</u>
\l@dminpagelines	<u>1835</u> , 1881, 1983, 2107, 2128
\l@dnumpstartsL	488, 612, 613, 615, 617, 632, 633, 703, 704, <u>1493</u> , 1525, 1533, 1573, 1574, 1652, 1847, 1848, 1992, 2217
\l@dnumpstartsR	55, 85, 493, 662, 663, 665, 667, 681, 682, 724, 725, <u>1493</u> , 1526, 1536, 1573, 1574, 1653, 1847, 1848, 1993, 2253
\l@doldbbl@set@language	1467
\l@doldselectlanguage	1466, 1469, 1474

<code>\l@dpagfullfalse</code>	2098, 2147, 2152, 2159, 2164
<code>\l@dpagfulltrue</code>	2098, 2146, 2151, 2158, 2163
<code>\l@dpagingfalse</code>	510, 528
<code>\l@dpagingtrue</code>	522
<code>\l@dpairingfalse</code>	514, 527
<code>\l@dpairingtrue</code>	509, 521
<code>\l@dprintingcolumnfalse</code>	1654
<code>\l@dprintingcolumnstrue</code>	1571
<code>\l@dprintingpagesfalse</code>	1996
<code>\l@dprintingpagestrue</code>	1853
<code>\l@dpscL</code>	746, 748, 753, 771, 802, 812, 815, 1495, 1527, 1533, 1545, 1582, 1588, 1597, 1600, 1602, 1650, 1793, 1801, 1810, 1815, 1823, 1832, 1860, 1868, 1873, 1876, 1884, 1899, 1900, 1902, 1921, 1922, 1990, 2205, 2207, 2210, 2217, 2225–2227, 2229, 2231, 2338, 2395
<code>\l@dpscR</code>	837, 840, 845, 863, 1496, 1528, 1536, 1548, 1583, 1589, 1608, 1611, 1613, 1651, 1794, 1805, 1811, 1816, 1827, 1833, 1861, 1869, 1877, 1885, 1941, 1942, 1944, 1963, 1964, 1991, 2241, 2243, 2246, 2253, 2259–2261, 2263, 2265, 2346, 2397
<code>\l@drd@ta</code>	795, 1087, 1095, 1113, 1121, 1134, 1142
<code>\l@drightbox</code>	732, 855, 871, 1623, 1949, 1950, 1958
<code>\l@drsn@te</code>	796
<code>\l@dsamepagefalse</code>	2098, 2146, 2151, 2158, 2163
<code>\l@dsamepagetrue</code>	2098, 2147, 2152, 2159, 2164
<code>\l@dsetupmaxlinecounts</code>	1506, 1523
<code>\l@dsetupprawboxes</code>	1498, 1522
<code>\l@dskipnumberfalse</code>	1061
<code>\l@dskipnumbertrue</code>	952
<code>\l@dskipversenumberfalse</code>	1919
<code>\l@dskipversenumberR</code>	927
<code>\l@dskipversenumberRfalse</code>	1961
<code>\l@dskipversenumberRtrue</code>	954
<code>\l@dunhbox@line</code>	795, 818, 821
<code>\l@dusedbabelfalse</code>	1449, 1463
<code>\l@dusedbabeltrue</code>	1449, 1465
<code>\l@duselanguage</code>	1456, 1595, 1606, 1896, 1937
<code>\l@dzeromaxlinecounts</code>	1506, 1524
<code>\l@dzeropenalties</code>	618, 668, 1577, 1855
<code>\l@luatexbodydir@L</code>	596, 809
<code>\l@luatexbodydir@R</code>	646
<code>\l@luatexpardir@L</code>	595, 808
<code>\l@luatexpardir@R</code>	645
<code>\l@luatextextdir@L</code>	594, 791, 807, 810
<code>\l@luatextextdir@R</code>	644
<code>\l@prev@nopbR</code>	62, 2325, 2332, 2333
<code>\l@prev@pbR</code>	61, 2324, 2330, 2331
<code>\l@pscL</code>	1495
<code>\l@pscR</code>	1495
<code>\labelref@listR</code>	1304
<code>\languagenam</code>	1451, 1452
<code>\last@page@numR</code>	309
<code>\lastbox</code>	756, 848
<code>\lcolwidth</code>	7, 9, 13, 524, 627, 764, 778, 788, 1603, 1718, 1739, 1756, 1767, 1783, 1800, 1822, 1850

\led@err@BadLeftRightPstarts	<u>21</u> , 1574, 1848
\led@err@Columns@InsideEnv	<u>32</u> , 1569
\led@err@LeftOnRightPage	<u>24</u> , 2038
\led@err@Leftside@PreviousNotPrinted	<u>28</u> , 535
\led@err@LineationInNumbered	157
\led@err@ManyLeftnotes	1356
\led@err@ManyRightnotes	1363
\led@err@ManySidenotes	1336
\led@err@NumberingNotStarted	102
\led@err@numberingShouldHaveStarted	132
\led@err@NumberingStarted	52
\led@err@NumberingWithoutPstart	86
\led@err@Pages@InsideEnv	<u>32</u> , 1841
\led@err@PendNoPstart	690, 711
\led@err@PendNotNumbered	687, 708
\led@err@PstartInPstart	603, 653
\led@err@PstartNotNumbered	599, 649
\led@err@RightOnLeftPage	<u>24</u> , 2047
\led@err@Rightside@PreviousNotPrinted	<u>28</u> , 555
\led@err@TooManyPstarts	<u>18</u> , 614, 664
\led@error@fail@patch@@outputpage	<u>42</u> , 2307
\led@error@fail@patch@pagenumbering	<u>39</u> , 2288
\led@error@fail@patch@thepage	<u>36</u> , 2281
\led@mess@NotesChanged	99
\led@mess@SectionContinued	126, 142, 151
\led@nopbnumR	2329, <u>2330</u>
\led@nopbR	2328, <u>2330</u>
\led@pb@setting	1681, 1689, 2145, 2149, 2157, 2161, 2172, 2173, 2183, 2184
\led@pbnumR	2327, <u>2330</u>
\led@pbR	2326, <u>2330</u>
\led@warn@BadAction	958
\led@warn@BadLineation	183
\led@warn@DuplicateLabel	1308
\ledgroupnotesL@false	2354
\ledgroupnotesL@true	2342
\ledgroupnotesR@false	2357
\ledgroupnotesR@true	2350
\ledllfill	788
\lednopb	567
\lednopbnum	2172, <u>2326</u>
\lednopbnumR	2183, <u>2326</u>
\lednopbR	567, 2328
\ledpb	566
\ledpbnum	2173
\ledpbnumR	2184, <u>2326</u>
\ledpbR	566, <u>2326</u>
\ledRcol@false	874
\ledRcol@true	838
\ledRcolfalse	537, 573
\ledRcoltrue	557

<code>\ledrlfill</code>	795, 1727, 1749
<code>\ledsectnomark</code>	801
<code>\ledsectnotoc</code>	800, 1898, 1940
<code>\ledstrutL</code>	1908, 1916, 2021
<code>\ledstrutR</code>	1950, 1958, 2021
<code>\ledthegoal</code>	2106, 2127, 2189
<code>\leftlinenumR</code>	227, 1085, 1097
<code>\leftpstartnumL</code>	1106
<code>\leftpstartnumR</code>	1106
<code>Leftside (environment)</code>	11, 533
<code>\Leftsidehook</code>	543, 548
<code>\Leftsidehookend</code>	547, 548
<code>\letcs</code>	395, 396, 453, 459, 462, 746, 837
<code>\line@list@stuff</code>	143
<code>\line@list@stuffR</code>	70, 127, 152, 417
<code>\line@list@version</code>	423
<code>\line@listR</code>	94, 245, 257, 384
<code>\line@margin</code>	1116
<code>\line@marginR</code>	192, 1090, 1137
<code>\line@num</code>	403, 909, 2220
<code>\line@numR</code>	63, 234, 241, 270, 304, 377, 381, 406, 890, 970, 979, 1068, 1070, 1072, 1073, 2256
<code>\lineation</code>	188, 189, 568
<code>\lineation*</code>	12, 188
<code>\lineationR</code>	12, 155, 190, 568
<code>\linenum@out</code>	433, 1556, 1638, 2079
<code>\linenum@outR</code>	414, 420, 422, 423, 428, 429, 435, 1561, 1642, 2082, 2326–2329
<code>\linenumberlist</code>	1069, 1073
<code>\linenumincrement</code>	12, 204
<code>\linenumincrement*</code>	12, 204
<code>\linenummargin</code>	192
<code>\linenumr@p</code>	1208, 1212
<code>\linenumrepR</code>	224, 234
<code>\linenumsep</code>	229, 231, 1153, 1156, 1165, 1168
<code>\linesinpar@listL</code>	252, 404, 2053, 2056
<code>\linesinpar@listR</code>	252, 261, 407, 2059, 2062
<code>\lineskip</code>	1256, 1299
<code>\linesonpage@listL</code>	410, 2066, 2069
<code>\linesonpage@listR</code>	262, 412, 2072, 2075
<code>\list@clear</code>	257–262, 607, 657
<code>\list@clearing@regR</code>	256
<code>\list@create</code>	245–248, 252–254, 385, 393, 448, 484, 485, 1172, 1304
<code>\list@pstartL@pc</code>	484, 487, 497, 498
<code>\list@pstartR@pc</code>	484, 492, 503, 504
<code>\listbreak</code>	2012, 2016
<code>\listxadd</code>	2330–2333
<code>\lock@disp</code>	1029, 1033, 1038
<code>\lock@off</code>	337
<code>\luatexbodydir</code>	596, 646, 809
<code>\luatexpardir</code>	595, 645, 808
<code>\luatextextdir</code>	594, 644, 791, 807

M

\managestanza@modulo	1425
\maxchunks	6, 1489
\maxlinesinpar@list	252
\memorydump	11, 542, 562
\memorydumpL	137, 542
\memorydumpR	137, 562
\message	69
\multiply	1003, 1721, 1770

N

\n@num	358
\namebox	748, 753, 840, 845, 1433, 1545, 1548, 2205, 2241
\NeedsTeXFormat	2
\new@line	818, 821
\new@lineL	432, 793
\new@lineR	434
\newbool	439, 442
\newbox	578, 732, 733, 1434
\newcommandx	586, 636, 685, 706, 1414
\newcounter	110–113, 195, 197, 199, 201, 581, 583, 2309
\newhookcommand@series	472, 475
\newif	6, 10, 48, 415, 531, 532, 730, 731, 928, 1160, 1370, 1449, 1530, 1542, 1679, 1680, 2098, 2100, 2201, 2202
\newlength	1707, 1710
\newmarks	2334–2336
\newnamebox	1433, 1501, 1502, 2004, 2007
\newnamecount	1444, 1509
\newsavebox	1565, 1566
\newseries@par	436, 480
\newwrite	414, 2323
\next@absline	1682, 1684, 1686, 2150–2152
\next@abslineR	1683, 1685, 1687, 2162–2164
\next@actionlineR	921, 966, 988, 990
\next@actionR	922, 967, 968, 973, 974, 982, 991
\next@insert	608
\next@insertR	658, 1176, 1179, 1181, 1184, 1188
\next@page@numR	67, 273, 275
\noindent	633, 682, 704, 725
\normal@page@breakR	60
\normal@pars	84, 611, 661
\normalbfnoteX	1370
\noteschanged@true	92, 95, 1178
\notesXwidthliketwocolumns	8
\num@lines	692, 1578, 1856
\num@linesR	577, 713, 1579, 1857
\numberedpar@true	628, 677
\numberingRfalse	83
\numberingRtrue	57, 121, 148
\numberingtrue	139

<code>\numberpstartfalse</code>	12
<code>\numberpstarttrue</code>	12
<code>\numdef</code>	454, 802, 1354, 1361, 1682, 1683, 2150, 2162, 2395, 2397
<code>\numgdef</code>	1326, 1333, 2171, 2182
<code>\numlabfont</code>	234
<code>\numpagelinesL</code>	<u>1835</u> , 1905, 1929, 1933, 2107, 2173, 2415
<code>\numpagelinesR</code>	<u>1835</u> , 1947, 1971, 1975, 2128, 2184

O

<code>\old@footnote</code>	462, 466
<code>\old@otherlanguage</code>	1479
<code>\old@startstanza</code>	544, 545, 570, 571
<code>\one@line</code>	753, 756, 795, 818, 821
<code>\one@lineR</code>	<u>577</u> , 845, 848
<code>\onlysideX</code>	10
<code>\onlyXside</code>	2432
<code>\openout</code>	78, 422, 429
<code>\otherlanguage</code>	1479, 1480

P

<code>\p@pstartL</code>	630
<code>\p@pstartR</code>	679
<code>\PackageError</code>	17
<code>\page@action</code>	274, <u>312</u> , 369
<code>\page@num</code>	1118
<code>\page@numR</code>	<u>250</u> , 376, 381, 968, 1092, 1139, 1349
<code>\pagebreak</code>	1696
<code>\pagegoal</code>	2197
<code>\pagenumbering</code>	40, 2284
<code>\Pages</code>	8, 29, 31, 33, <u>1839</u>
<code>pages (environment)</code>	8, 508
<code>\pagetotal</code>	1910, 1912, 1952, 1954, 2028, 2035, 2044, 2106, 2127
<code>pairs (environment)</code>	6, <u>508</u>
<code>\paperwidth</code>	817, 820
<code>\par@line</code>	693, 1580, 1858
<code>\par@lineR</code>	<u>577</u> , 714, 1581, 1859
<code>\par@patch@thepage</code>	<u>2276</u>
<code>\parbox</code>	1603, 1614, 1800, 1804, 1822, 1826
<code>\parledgroup@</code>	755, 847, <u>2334</u> , 2360
<code>\parledgroup@beforenotes@save</code>	701, 722, <u>2422</u>
<code>\parledgroup@beforenotesL</code>	<u>2420</u>
<code>\parledgroup@beforenotesR</code>	<u>2420</u>
<code>\parledgroup@correction@notespacing</code>	1915, 1957, 2409
<code>\parledgroup@correction@notespacing@final</code>	2235, 2269, <u>2390</u>
<code>\parledgroup@correction@notespacing@init</code>	1934, 1976, <u>2385</u> , 2393
<code>\parledgroup@notes@endL</code>	2208, 2211, 2234, 2353
<code>\parledgroup@notes@endR</code>	2244, 2247, 2268, 2356
<code>\parledgroup@notes@startL</code>	755, <u>2337</u> , <u>2353</u>
<code>\parledgroup@notes@startR</code>	847, <u>2337</u> , <u>2353</u>
<code>\parledgroup@notespacing@correction</code>	<u>2379</u> , 2412–2414

\parledgroup@notespacing@correction@accumulated	... 2386, 2392, 2413
\parledgroup@notespacing@correction@modulo 2387, 2414–2416
\parledgroup@notespacing@set@correction 1844, <u>2379</u>
\parledgroup@series 2335, 2339, 2340, 2347, 2348, 2363, 2364, 2370, 2371
\parledgroup@type 2336, 2339, 2340, 2347, 2348, 2361, 2368
\parledgroupseries@ <u>2334</u>
\parledgrouptrue 8
\parledgrouptype@ <u>2334</u>
\patchcmd 1450, 2278
\pausenumbering 560
\pausenumberingR <u>120</u> , 560
\pend 12, 541, 565, 604, 1416
\pendL 541, <u>685</u>
\pendR 565, 654, <u>706</u>
\pretocmd 2311
\prev@abslineverse 2171–2173, 2182–2184
\prev@nopbR <u>2324</u>
\prev@pbR <u>2324</u>
\prevgraf 692, 713, 1578, 1579, 1856, 1857
\print@columnseparator 1622, <u>1659</u> , 1803, 1825
\print@eledsectionL <u>798</u> , 1603, 1903
\print@eledsectionR <u>878</u> , 1614, 1945
\print@last@after@pendLfalse 2119
\print@last@after@pendRfalse 2140
\print@last@after@pendRtrue 2271
\print@lineL 773, <u>783</u>
\print@lineR 865, 878
\print@notesX 1865
\print@notesX@forpages <u>1217</u> , 1865
\print@Xnotes 1864
\print@Xnotes@forpages <u>1217</u> , 1864
\printlinesR <u>1205</u>
\ProcessOptions 12
\protected@csxdef 464
\protected@edef 629, 678
\providebool 386
\ProvidesPackage 3
\pst@rtedLfalse 47
\pst@rtedLtrue 140, 609
\pst@rtedRfalse 56, 89
\pst@rtedRtrue 124, 149, 659
\pstart 12, 19, 23, 539, 564, 1410
\pstartL 539, <u>580</u>
\pstartnumfalse 1153, 1158
\pstartnumRfalse 1165, 1170
\pstartnumRtrue 1161, 1587, 2266
\pstartnumtrue 1586, 2232
\pstartR 564, <u>580</u>

R

\Rcolwidth	7, 9, <u>13</u> , 525, 676, 856, 871, 1614, 1719, 1740, 1757, 1768, 1784, 1804, 1826, 1851
\read@linelist	264, 418
\reledpar@error	17, 19, 22, 25, 27, 29, 31, 33, 35, 37, 40, 43
\rem@inder	1073, 1075–1077
\RequirePackage	5
\resetprevline@	1639, 1643, 2221, 2257
\restore@pstartL@pc	<u>496</u> , 1590, 1886, 2233
\restore@pstartR@pc	<u>496</u> , 1591, 1887, 2267
\resumenumbering	561
\resumenumberingR	<u>120</u> , 561
\rightlinenumR	<u>227</u> , 1087, 1095
\rightpstartnumL	<u>1106</u>
\rightpstartnumR	<u>1106</u>
Rightside (environment)	<u>11</u> , <u>553</u>
\Rightsidehook	548, 569
\Rightsidehookend	548, 574
\rlap	1087, 1095, 1113, 1121, 1134, 1142
\Rlineflag	<u>220</u>
\rule	1699

S

\sameparallelpagenumbertrue	9, 11
\savebox	1603, 1614
\sc@n@list	1074, 1076
\section@num	141–143
\section@numR	<u>45</u> , 58, 69, 70, 77, 78, 125–127, 150–152
\select@language	1451, 1452, 1485
\selectlanguage	<u>1456</u>
\set@line	<u>436</u>
\set@line@action	267, <u>312</u> , 371
\setgoalfraction	9, <u>2189</u>
\sethangingsymbol	<u>14</u>
\setl@dlp@rbox	1342, 1357, 1359
\setl@drp@rbox	1344, 1352, 1364
\setline	<u>436</u>
\setlinenum	<u>436</u>
\setnamebox	617, 667, <u>1433</u>
\setnotepositionliketwocolumns@C	<u>1713</u>
\setnotepositionliketwocolumns@L	<u>1713</u>
\setnotepositionliketwocolumns@R	<u>1713</u>
\setnotespositionliketwocolumns@C	1752
\setnotespositionliketwocolumns@L	1730
\setnotespositionliketwocolumns@R	1779
\setparledgroupnotespacing	2377
\setpositionliketwocolumns@C	<u>1713</u> , 1747
\setpositionliketwocolumns@L	<u>1713</u> , 1726
\setpositionliketwocolumns@R	<u>1713</u> , 1775
\setprintlines	1206
\setRlineflag	<u>12</u> , 220, 223

<code>\setwidthliketwocolumns@C</code>	1713 , 1734
<code>\setwidthliketwocolumns@L</code>	1713, 1713
<code>\setwidthliketwocolumns@R</code>	1713 , 1764
<code>\shiftedpstartstrue</code>	7
<code>\sidenote@margin</code>	1319, 1322
<code>\sidenote@marginR</code>	1316 , 1347
<code>\sidenotecontent@</code>	1325, 1330, 1331, 1342, 1344, 1352, 1353, 1357, 1359, 1360, 1364
<code>\sidenotemargin</code>	1316
<code>\sidenotemargin*</code>	1316
<code>\sidenotesep</code>	1331
<code>\skip</code>	1233, 1241, 1277, 1285, 2363, 2370
<code>\skip@lockoff</code>	337
<code>\skipnumbering</code>	12, 436
<code>\smash</code>	1699
<code>\splitbotmarks</code>	2360, 2361, 2363, 2364, 2368, 2370, 2371
<code>\splitfirstmarks</code>	755, 847, 2339, 2340, 2347, 2348
<code>\splittopskip</code>	750, 842, 1253, 1296
<code>\stanza@count</code>	1394, 1407, 1420
<code>\stanza@hang</code>	1396, 1428
<code>\stanza@modulo</code>	1394, 1423
<code>\stanzaindentbase</code>	1375, 1384, 1421, 1424
<code>\startlock</code>	436
<code>\startsub</code>	436
<code>\stepcounter</code>	2297, 2300, 2303
<code>\sub@action</code>	283, 312 , 370
<code>\sub@change</code>	68, 277, 278, 284
<code>\sub@lock</code>	904
<code>\sub@lockR</code>	65, 292, 294, 296, 299, 313, 319, 320, 322, 323, 343, 344, 346, 885, 944, 946, 947, 949, 1010, 1050, 1052, 1054
<code>\subline@num</code>	236, 905, 910
<code>\subline@numR</code>	237, 241 , 300, 304, 378, 382, 886, 891, 970, 977, 1064, 1065
<code>\sublinenumincrement</code>	12, 204
<code>\sublinenumincrement*</code>	12, 204
<code>\sublinenumr@p</code>	1209, 1213
<code>\sublinenumrepR</code>	224 , 237
<code>\sublines@false</code>	66, 281, 934
<code>\sublines@true</code>	279, 932
<code>\sublock@disp</code>	1012, 1016, 1021
<code>\sympplinum</code>	1208
<code>\sza@penalty</code>	1402, 1406

T

<code>\temp</code>	1716, 1717, 1720–1723, 1736, 1737, 1742–1744, 1753, 1755, 1758–1761, 1765, 1766, 1769–1772, 1780, 1782, 1785–1788
<code>\temp@</code>	802, 803
<code>\temp@spacing</code>	2382, 2383
<code>\tempa</code>	1754, 1756–1758, 1781, 1783–1785
<code>\textwidth</code>	14, 16, 524, 525, 1850, 1851
<code>\theledlanguageL</code>	1456 , 1595, 1896
<code>\theledlanguageR</code>	1456 , 1606, 1937

`\thepage` 433, 435, 2278
`\thepstart` 540, 563
`\thepstartL` 12, 540, 582, 623, 630, 1152, 1157
`\thepstartR` 12, 563, 584, 672, 679, 1164, 1169
`\thesameparallepage@number` 2309
`\this@line@list@version` 423
`\thr@@` 322, 331, 344, 351, 939, 947
`\togglefalse` 452
`\toggletrue` 450
`\topskip` 2028

U

`\unhbox` 1438, 1618, 1623, 1908, 1916, 1950, 1958
`\unhnamebox` 1433
`\unless` 163, 171, 179, 437, 441, 445, 471, 474, 2431
`\unvbox` 756, 848, 1257, 1300, 1440
`\unvnamebox` 1433
`\usebox` 1620, 1625
`\usenamecount`
 . . 1395, 1401, 1444, 1516, 1873, 2207, 2210, 2227, 2229, 2243, 2246, 2261, 2263, 2338, 2346

V

`\value` 1419
`\vbadness` 749, 841
`\vbox` 617, 667, 1254, 1297, 1603, 1614, 2313
`\vsplit` 753, 845, 1255, 1298

W

`\widthliketwocolumns` 8
`\widthliketwocolumnstrue` 9
`\WithSuffix` 188, 204, 208, 212, 216, 1316
`\writtenlinesLfalse` 1862, 2218
`\writtenlinesLtrue` 2215
`\writtenlinesRfalse` 1863, 2254
`\writtenlinesRtrue` 2251

X

`\xifinlist` 771, 803, 863, 1597, 1608, 1902, 1944
`\xifinlistcs` 1684–1687, 1690–1693, 2146, 2147, 2151, 2152, 2158, 2159, 2163, 2164
`\Xnoteswidthliketwocolumns` 8
`\Xonlyside` 10, 2432
`\xpg@main@language` 1486, 1487
`\Xpstart` 164, 172, 180
`\xright@appenditem`
 316, 318, 327, 340, 342, 349, 363, 380, 397, 404, 407, 410, 412, 455, 487, 492
`\xspace` 467

Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
General: Added section of babel related code	64
Fix babel problems	1
\Columns: Added \l@dchecklang and \l@duselanguage to \Columns	68
\Pages: Added \l@duselanguage to \Pages	77
v0.3.0.	
General: Added \do@lineLhook and \do@lineRhook	45
Reorganize for ledarab	1
\affixline@numR: Changed \affixline@numR to match new eledmac	49
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR	47
\do@lineL: Added \do@lineLhook to \do@lineL	43
Simplified \do@lineL by using macros for some common code	43
\do@lineR: Changed \do@lineR similarly to \do@lineL	45
Leftside: Added hooks into Leftside environment	37
\flag@end: Removed extraneous spaces from \flag@end	32
\ifledRcol: Moved \ifl@dpairing to eledmac	18
\ifpstartedR: Moved \ifpstartedL to eledmac	20
\l@dlinenumR: Simplified \leftlinenumR and \rightlinenumR by introducing \l@dlinenumR	25
\l@dnumpstartsR: Moved \l@dnumpstartsL to eledmac	65
\ledstrutR: Added \ledtrutL and \ledstrutR	80
\normalbfnotex: Removed extraneous spaces from \normalbfnotex	61
\Pages: Added \ledstrutL to \Pages	77
Added \ledstrutR to \Pages	78
\printlinesR: Simplified \printlinesR by using \setprintlines	55
\Rightsidehookend: Added \Leftsidehook, \Leftsidehookend, \Rightsidehook and \Rightsidehookend	37
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR	25
v0.3.a.	
General: Minor \linenummargin fix	1
\line@marginR: Do not just set \line@marginR in \linenummargin	24
v0.3.b.	
General: Improved parallel page balancing	1
\Pages: Added \l@dminpagelines calculation for succeeding page pairs	79
v0.3.c.	
General: Compatibilty with Polyglossia	1
v0.4.0.	
General: No more ledparpatch. All patches are now in the main file.	1
v0.5.0.	
General: Corrections about \section and other titles in numbered sections	1
v0.6.0.	
General: Be able to us \chapter in parallel pages.	1
v0.7.0.	
General: Option ‘shiftedverses’ which make there is no blank between two parallel verses with inequal length.	1

v0.8.0.	
General: Possibility to have a symbol on each hanging of verses, like in the french typography. Redefine the commande <code>\hangingsymbol</code> to define the character.	1
v0.9.0.	
General: Possibility to number <code>\pstart</code>	12
Possibility to number the <code>pstart</code> with the commands <code>\numberpstarttrue</code>	1
<code>\iflledRcol</code> : Moved <code>\iflledRcol</code> and <code>\ifnumberingR</code> to <code>eledmac</code>	18
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.	
General: Change <code>\do@lineL</code> and <code>\do@lineR</code> to allow line numbering by <code>pstart</code> (like in <code>eledmac</code> 0.15).	43
Lineation can be by <code>pstart</code> (like in <code>eledmac</code> 0.15).	22
New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . .	61
<code>\affixline@numR</code> : Changed <code>\affixline@numR</code> to allow to disable line numbering (like in <code>eledmac</code> 0.15).	49
<code>\Columns</code> : Line numbering by <code>pstart</code>	69
<code>\get@nextboxR</code> : Change <code>\get@nextboxL</code> and <code>\get@nextboxR</code> to allow to disable line numbering (like in <code>eledmac</code> 0.15).	85
<code>Pstart</code> number can be printed in side	86
<code>\inserthangingsymbolR</code> : Prevent the column separator for hanging verse from shifting	61
v0.12.0.	
General: New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	61
v1.0.0.	
General: Compatibility with <code>eledmac</code> . Change name to <code>eledpar</code>	1
Debug in lineation by <code>pstart</code>	22
v1.0.1.	
General: Correction on <code>\numberonlyfirstinline</code> with lineation by <code>pstart</code> or by page.	1
v1.1.0.	
General: <code>Shiftedverses</code> becomes <code>shiftedpstarts</code>	1
<code>\pstartR</code> : Add <code>\labelpstarttrue</code> (from <code>eledmac</code>).	39
v1.1.1.	
<code>\pstartR</code> : Correct <code>\pstartR</code> bug introduced by 1.1.	39
v1.1.2.	
<code>\affixside@noteR</code> : Remove spurious space between line number and line content .	60
v1.2.0.	
General: Support for <code>\led<section></code> commands in parallel texts.	1

v1.2.1.	
<code>\initnumbering@sectcountR</code> : For the right section, the counter is defined only once.	21
v1.3.0.	
<code>\edtext</code> : Manage RTL language.	33
v1.3.1.	
<code>\l@dbfnote</code> : Compatibility of standard footnotes with <code>eledmac</code> when theses footnotes contain any commands.	61
v1.3.2.	
General: Debug with some classes.	1
v1.3.3.	
General: Debugging the left notes of the right column.	60
<code>\l@dbfnote</code> : Spurious space with footnote in right column.	61
v1.3.4.	
General: Allow use of commands in sidenotes, as introduced by <code>eledmac 1.0</code>	60
v1.3.5.	
<code>\normalbfnoteX</code> : Allows one to redefine <code>\thefootnoteX</code> with <code>\alph</code> when some packages are loaded.	61
v1.4.0.	
General: Added <code>\do@insidelineLhook</code> and <code>\do@insidelineRhook</code>	45
v1.4.1.	
<code>\normalbfnoteX</code> : Fix bug with normal familiar footnotes when mixing RTL and LTR text.	61
<code>astanza</code> : Enable the use of <code>stanzaindentsrepetition</code> within <code>astanza</code> environment. ...	62
v1.4.3.	
General: Corrects a false hanging verse when a verse is exactly the length of a line.	1
<code>\inserthangingsymbolR</code> : Hanging verse is no longer automatically flush right.	61
<code>\pendL</code> : Spurious spaces in <code>\pendL</code>	41
<code>\pendR</code> : Spurious spaces in <code>\pstartR</code>	42
<code>\pstartR</code> : Spurious spaces in <code>\pstartL</code> and <code>\pstartR</code>	39
v1.5.0.	
General: Add, as in <code>eledmac</code> , features to manage page breaks.	1
<code>\sublinenumincrement*</code> : Add starred version of <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> to change both Left and Rightside.	24
v1.6.0.	
General: Add tool and documentation for parallel ledgroups	15
v1.7.0.	
General: Add, as in <code>eledmac</code> , features to make crossrefs with <code>pstart</code> numbers.	1
v1.8.0.	
General: <code>\beginnumbering</code> is defined only on <code>eledmac</code> , not on <code>eledpar</code>	20
<code>\l@dlsnote</code> , <code>\l@drsnote</code> and <code>\l@dcsnote</code> defined only one time, in <code>eledmac</code>	60
Add <code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code>	8
Add <code>\columnspanposition</code>	7
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.	43
Compatibility with <code>musixtex</code>	1

Debug eledmac sectioning command after using <code>\resumenumbering</code>	1
New sectioning commands, as in eledmac.	16
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.	64
<code>\Columns</code> : Modify <code>\Columns</code> to enable to add section's title.	68
Suppress <code>\l@dchecklang</code> from <code>\Columns</code>	68
<code>\l@dchecklang</code> : Suppress <code>\l@dchecklang</code> which did not work and was not logical, because both columns could have the same language but not the main language of the document.	64
<code>\Pages</code> : Modify <code>\Pages</code> to enable to add section's title.	75
<code>\pendL</code> : As in eledmac, <code>\pendL</code> can have an optional argument.	41
<code>\pendR</code> : As in eledmac, <code>\pendR</code> can have an optional argument.	42
<code>\print@columnseparator</code> : Move some code of <code>\Columns</code> to <code>\print@columnseparator</code>	70
<code>\pstartR</code> : As in eledmac, <code>\pendL</code> and <code>\pendR</code> can have an optional argument.	39
<code>\sidenotemargin*</code> : <code>\sidenotemargin</code> is now directly defined in eledmac to be able to manage eledpar.	60
Add <code>\sidenotemargin*</code>	60
<code>\theledlanguageR</code> : Correct left/right language setting with polyglossia.	65
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) (?).	43
<code>\do@lineR</code> : Fix a bug with critical notes at the beginning of a page, added by v1.8.0 (?).	45
v1.8.2.	
General: Debug <code>\eledxxx</code> with some paper sizes	1
Debug left and side note (bugs added by 1.8.0)	1
<code>\eledpar@error</code> : Errors specific to eledpar send to eledpar handbook	18
<code>\flag@end</code> : <code>\flag@start</code> and <code>\flag@end</code> are now defined only one time for eledmac and eledpar	32
<code>\lineation*</code> : Add <code>\lineation*</code>	23
v1.8.3.	
General: Add <code>\noeledxxx</code> , as in eledmac	1
<code>\doinsidelineRhook</code> : Added <code>\dolineLhook</code> , <code>\dolineRhook</code> , <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code>	45
<code>\Pages</code> : Debug blank pages when using optional argument in the last <code>\pend</code>	75
<code>\resumenumberingR</code> : Debug <code>\resumenumberingR</code>	22
v1.9.0.	
General: Add <code>\AtBeginPairs</code> macro.	7
Compatibility with <code>\Xnoteswidthliketwocolumns</code> and <code>\notesXwidthliketwocolumns</code>	1
<code>\ifwidthliketwocolumns</code> : Added <code>widthliketwocolumns</code> option	18
<code>\theledlanguageR</code> : Debug left/right language switching with polyglossia. Do not write in .aux file when setting left/right lines.	65
v1.9.1.	
<code>\ifledRcol</code> : Moved <code>\ifl@dpaging</code> to eledmac	18
v1.10.0.	
General: Compatibility with <code>\AtEveryPstart</code> and <code>\AtEveryPend</code>	1
Restore critical notes in <code>\eledsection</code> in parallel columns (this bug was added in 1.8.2).	1

\Pages: Debug wrong pages splitting when no optional argument is used in last \pend (bug was added in v.1.8.3).	75
Debug wrong parallel pages synchronization when an \edtext falls across two pages.	75
v1.10.1.	
\line@list@stuffR: Revert modification of 1.4.2, which makes bugs with number- ing. Leave vertical mode to solve spurious space before minipage.	32
v1.11.0.	
General: Compatibility of standard footnotes with some biblatex styles.	1
\edtext: \critext and \edtext are now defined only in eledmac.	33
v1.12.0.	
General: Compatibility with Lua \TeX RTL languages.	1
\Columns: Add \l@dprintingcolumnstrue	68
\edlabel: \edlabel and \edindex works now with hyperref when using eled- par.	59
\edlabel is now defined only one time for both eledmac and eledpar	59
\Pages: Add \l@dprintingpagestrue	75
\print@eledsectionL: Compatibility with Lua \TeX RTL languages.	44
\print@eledsectionR: Compatibility with Lua \TeX RTL languages.	46
\print@lineL: Compatibility with Lua \TeX RTL languages.	44
v1.12.1.	
\print@eledsectionL: Fixes bug with Lua \TeX RTL \eledsection.	44
v1.13.0.	
General: 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
\clearl@drihtpage: Use \newpage instead of \clearpage.	81
\ifledRcol: Remove false boolean settings which are not needed.	18
\Pages: Prevent false overfull hboxes when using \Pages outside of pages environ- ment.	75
When using shiftedpstarts option, a \l@dleftbox with a null height will advance the \pagetotal in any case.	75
astanza: Enable the use of optional argument of & in astanza environment.	62
v1.13.1.	
\correct@footinsX@box: Call \correct@footinsX@box and \correct@Xfootins@box directly in \print@notesX@forpages and \print@Xnotes@forpages.	55
Correct \correct@footinsX@box and \correct@Xfootins@box	55
\Pages: Prevent false empty page after \Pages (bug added in 1.13.0)	75
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.	
General: Add \AtEveryPstartCall.	1
Add sameparallelpagenumber option.	9
Fix vertical spurious space before right \eledchapter (bug added in v1.13.0).	1
Prevent vertical space when using \AtEveryPstart or \AtEveryPend with a command which prints nothing	1

\do@actions@nextR: Add action 1008 and 1009	47
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting when a verse is hanging	61
\lineationR: As \lineation, \lineationR automatically set the \pstart-infootnote.	23
\n@num: \n@num defined only one time for both Eledmac and Eledpar.	29
\skipnumbering: \skipnumbering defined only one time for both Eledmac and Eledpar	33
v1.16.0.	
General: Error message when calling \Pages inside ‘pages’ environment and \Columns inside ‘pairs’ environment.	1
Error message when starting a Leftside/a Rightside while the previous one has not been yet typeset.	1
Error message when using \beginnumbering...\endnumbering without \pstart.	1
Fix bug with nofamiliar / nocritical option of eledmac.	1
New package option sameparallelpagenummer to have the same page number for both left and right side.	1
\newseries@par: Fix bug with \onlysideX.	33
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
v2.0.0.	
\@adv: \@adv defined only in reledmac.	28
\@lab: \@lab defined only in eledmac.	60
\@ref@regR: \@ref defined only in reledmac, code specific to right side moved in \ref@regR.	29
\@set: \@set defined only in reledmac.	28
General: \@nl is now defined only in reledmac.	27
\ifbypage@ and \ifbypstart@R defined in eledmac.	23
Fix some bugs with ‘sameparallelpagenummer’ option.	1
Many code refactored and moved to reledmac.	1
Package’s name becomes reledpar.	1
Totally new implementation of ‘sameparallelpagenummer’ option.	1
\advanceline: \advanceline defined only in reledmac.	32
\bbl@set@language: Patch \bbl@set@language instead of redefining it ..	64
\do@lockonR: \do@lockon defined only in reledmac.	28
\endlock: \startlock and \endlock defined only in reledmac.	33
\endsub: \startsub and \endsub defined only in reledmac.	32
\fix@page: \fix@page is defined only once in reledmac	28
chapterinpages: Deleting the old system of managing parallel chapter, keep only the new one with \patchcmd.	37
\l@d@set: \l@d@set defined only in reledmac.	28
\l@dbfnote: \l@dbfnote defined only in reledmac.	61
\line@marginR: \linenummargin now defined only once time in reledmac. ..	24

<code>\normalbfnoteX</code> : <code>\normalbfnoteX</code> defined only in <code>reledmac</code>	61
<code>\page@action</code> : <code>\page@action</code> defined only in <code>reledmac</code>	28
<code>\read@linelist</code> : <code>\read@linelist</code> is defined only once time in <code>\reledmac</code>	27
<code>\set@line</code> : <code>\set@line</code> defined only in <code>reledmac</code>	33
<code>\set@line@action</code> : <code>\set@line@action</code> defined only in <code>reledmac</code>	28
<code>\setline</code> : <code>\setline</code> defined only in <code>reledmac</code>	32
<code>\setlinenum</code> : <code>\setlinenum</code> defined only in <code>reledmac</code>	33
<code>\skip@lockoff</code> : <code>\do@lockoff</code> defined only in <code>reledmac</code>	29
<code>\sub@action</code> : <code>\sub@action</code> defined only in <code>reledmac</code>	28
<code>\sublinenumincrement*</code> : <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\first-</code> <code>sublinenum</code> , <code>\sublinenumincrement</code> are now defined only in <code>reledmac</code>	24
<code>\theledlanguageR</code> : Patch <code>\otherlanguage</code> instead of redefining it.	65