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 6
  1.1 Aim of this package 6
  1.2 Historical overview 6

2 Options 7
  2.1 Synchronization's options 7
  2.2 Other options 7

3 General 8

4 Parallel columns 9
  4.1 Basic use 9
  4.2 Setting 9
    4.2.1 Column's width 9
    4.2.2 Column's separator 9
    4.2.3 Column's positions 10

∗This file (reledpar.dtx) has version number v2.25.3, last revised 2022/04/22.
†maieul at maieul dot net
‡herries dot press at earthlink dot net
4.2.4 Mixing two columns and one column texts ........................................ 10
4.2.5 Schemas of column typesetting ....................................................... 10
4.2.6 Continuous numbering between single and double column texts .......... 10

4.3 Note about \AtEveryPstart* and \AtEveryPend* ................................. 11

5 Keeping translation in the outside/inside column ................................. 11

6 Facing pages ......................................................................................... 12

6.1 Basic usage ....................................................................................... 12
6.2 Setting ............................................................................................... 12

6.2.1 Text width ...................................................................................... 12

6.2.2 Way of synchronizing ................................................................. 12
6.2.3 Page number ................................................................................. 14
6.2.4 Page breaking .............................................................................. 14
6.2.5 Right page before \Pages .............................................................. 14
6.2.6 Notes about \mainmatter .............................................................. 14

6.3 Critical and familiar footnotes ........................................................... 14

6.3.1 Notes height setting .................................................................... 14
6.3.2 About the numbering of familiar footnotes .................................... 15
6.3.3 Using perpage package ............................................................... 15
6.3.4 Notes for one side only ............................................................... 15
6.3.5 Familiar notes called on the right side, but to be printed on the left side ......................................................................................................................... 16
6.4 Critical notes called on leftside but printed on right side ..................... 16
6.5 Using line flag ................................................................................... 16

7 Left and right texts ................................................................................ 17

7.1 Environments .................................................................................... 17

7.2 Numbering text lines and paragraphs .............................................. 17
7.3 First line number and line number increment .................................. 18
7.4 Lineation system ................................................................................ 18
7.5 Line number style ............................................................................ 18
7.6 Line number margin ......................................................................... 19
7.7 Line flag ............................................................................................ 19
7.8 Chunks ............................................................................................... 20
7.9 \AtEveryPstart and \AtEveryPstartCall .............................................. 20
7.10 Language setting ............................................................................ 20
7.11 Executing code at each line ............................................................. 21
7.12 Executing code at each page ............................................................. 21

8 Verse ..................................................................................................... 21

9 Side notes ............................................................................................. 22

10 Parallel ledgroups ............................................................................... 22

10.1 General ............................................................................................ 22
10.2 Parallel ledgroups and setspace package ....................................... 23

11 Sectioning commands ........................................................................ 23

12 Notes about page number ...................................................................) 23
<table>
<thead>
<tr>
<th>Contents</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VIII.7 Penalties</td>
<td>77</td>
</tr>
<tr>
<td>VIII.8 Printing leftover notes</td>
<td>78</td>
</tr>
<tr>
<td>IX Footnotes</td>
<td>79</td>
</tr>
<tr>
<td>IX.1 Footnotes output specific to \Pages</td>
<td>79</td>
</tr>
<tr>
<td>IX.2 Critical footnote printed in right side</td>
<td>82</td>
</tr>
<tr>
<td>X Cross referencing</td>
<td>84</td>
</tr>
<tr>
<td>XI Sidenotes</td>
<td>85</td>
</tr>
<tr>
<td>XII Verse</td>
<td>87</td>
</tr>
<tr>
<td>XIII Fixing babel and polyglossia</td>
<td>89</td>
</tr>
<tr>
<td>XIV Counts and boxes for parallel texts</td>
<td>91</td>
</tr>
<tr>
<td>XV Checking text to be processed</td>
<td>93</td>
</tr>
<tr>
<td>XVI Parallel columns</td>
<td>94</td>
</tr>
<tr>
<td>XVII Parallel pages</td>
<td>105</td>
</tr>
<tr>
<td>XVII.1 Specific counters</td>
<td>105</td>
</tr>
<tr>
<td>XVII.2 Main macro</td>
<td>105</td>
</tr>
<tr>
<td>XVII.3 Ensure all notes are printed at the end of parallel pages</td>
<td>111</td>
</tr>
<tr>
<td>XVII.4 Struts</td>
<td>112</td>
</tr>
<tr>
<td>XVII.5 Page clearing</td>
<td>112</td>
</tr>
<tr>
<td>XVII.6 Lines managing</td>
<td>113</td>
</tr>
<tr>
<td>XVII.7 Page break managing</td>
<td>115</td>
</tr>
<tr>
<td>XVII.8 Getting boxes content</td>
<td>118</td>
</tr>
<tr>
<td>XVIII The mainmatter option of \Pages</td>
<td>121</td>
</tr>
<tr>
<td>XIX Sections’ titles’ commands</td>
<td>122</td>
</tr>
<tr>
<td>XX Page break/no page break, depending on the specific line</td>
<td>123</td>
</tr>
<tr>
<td>XXI Parallel ledgroup</td>
<td>123</td>
</tr>
<tr>
<td>XXII Compatibility with eledmac</td>
<td>127</td>
</tr>
<tr>
<td>XXIII The End</td>
<td>127</td>
</tr>
<tr>
<td>Appendix A Some things to do when changing version</td>
<td>128</td>
</tr>
<tr>
<td>A.1 Migration to eledpar 1.4.3</td>
<td>128</td>
</tr>
<tr>
<td>A.2 Migration from eledpar to reledpar</td>
<td>128</td>
</tr>
<tr>
<td>A.2.1 Deprecated options</td>
<td>128</td>
</tr>
<tr>
<td>A.2.2 \renewcommand replaced with command</td>
<td>128</td>
</tr>
<tr>
<td>A.2.3 Commands the names of which have changed</td>
<td>129</td>
</tr>
<tr>
<td>A.3 Migration to reledpar 2.2.0</td>
<td>129</td>
</tr>
<tr>
<td>A.4 Migration to reledpar 2.3.0</td>
<td>129</td>
</tr>
<tr>
<td>A.5 Migration to reledpar 2.4.0</td>
<td>129</td>
</tr>
</tbody>
</table>
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 \texttt{reledpar} package is an extension to \texttt{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: \url{https://github.com/maieul/ledmac/issues/}.

This manual contains a general description of how to use \texttt{reledpar} starting in section \ref{firstsection}, the complete source code for the package, with extensive documentation (in sections \ref{xxi} through \ref{xxiii}), and an Index to the source code. As \texttt{reledpar} is an adjunct to \texttt{reledmac} we assume that you have read the \texttt{reledmac} manual. Also \texttt{reledpar} requires \texttt{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 \ref{firstsection} unless you are particularly interested in the innards of \texttt{reledpar}.

1.2 Historical overview

Many of the code of this package is based on the \texttt{eledpar} package, which was based on the \texttt{ledpar}, created as an extension of the \texttt{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 \texttt{reledmac}’s handbook in order to understand this evolution.
2 Options

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

2.1 Synchronization’s options

Please read the paragraph on synchronization’s option on 6.2.2 p. 12 to understand better those options.

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

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

nomaxlines allows facing pages to have different numbers of lines.

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

2.2 Other options

parledgroup allows the use of \texttt{\textbackslash ledgroup} environment with \texttt{\textbackslash reledpar}.

\texttt{\textbackslash widthliketwocolumns} set the width of the text printed in a single column to be the same as the width of the text printed in two parallel columns with \texttt{\textbackslash reledpar}. This is useful when alternating between normal and parallel typesetting.

\texttt{\textbackslash continuousnumberingwithcolumns} allow you to alternate between normal typesetting and parallel column typesetting without resetting the line number (see 4.2.6 p. 10).

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

\texttt{\textbackslash sameparallelpagernumber} sets page numbers on facing pages to the same value.

\texttt{\textbackslash prevpgonnumbered} enables that the page before facing pages (the one automatically inserted to start parallel pages on a left page) is not counted. This applies only if the page is empty.

\texttt{\textbackslash movecolumnspositiononrightpage} make the left column on the right page become the right column, and the left column become the right column. It allows to have a text running on the inner column, and an other one on the outer column.

\footnote{This option can either be used on \texttt{reledmac} or \texttt{reledpar}.}

\footnote{This option can either be used on \texttt{reledmac} or \texttt{reledpar}.}
3 General

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

The \relpar package lets you typeset two numbered texts in parallel.\footnote{You can use, anyway, \numberlinefalse to disable printing of line numbers.} This can be done either as setting the “Leftside” and “Rightside” texts in two columns or on facing pages. In the paired pages case footnotes are placed at the bottom of the page on which they are called out — that is, footnotes belonging to the left are set at the foot of a left (even numbered) page, and those for right texts are at the bottom of the relevant right (odd numbered) page. However, in the columnar case, all footnotes are set at the bottom left of the page on which they are called out — they are not set below the relevant column.

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

\relpar 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.

It is possible to have multiple chunks in the left and right texts before printing them. The macro \maxchunks{\langle num \rangle} 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 \relpar 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, \relmac is a \TeX resource hog, and \relpar only makes things worse in this respect.
4 Parallel columns

4.1 Basic use

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

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

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

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

\AtBeginPairs{\sloppy}

There is no required pagebreak before or after the columns.

4.2 Setting

4.2.1 Column’s width

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

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

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

4.2.2 Column’s separator

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

\setlength{\columnrulewidth}{0.4pt}

You can also modify \columnseparator if you want more control.
4.2.3 Column’s positions

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

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

\beforecolumnseparator
\aftercolumnseparator

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

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

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

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

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

4.2.4 Mixing two columns and one column texts

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

\Xnoteswidthliketwocolumns
\notesXwidthliketwocolumns

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

4.2.5 Schemas of column typesetting

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

4.2.6 Continuous numbering between single and double column texts

If you want to have continuous line numbers between multiple columns and single columns, use the continuousnumberingwithcolumns option when loading reledmac or reledpar. You will need to use \pausenumbering...\resumenumbering in addition to \beginnumbering...\endnumbering (see 5.2.7 p.19).

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

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

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

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

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

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

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

4.3  Note about \AtEveryPstart* and \AtEveryPend*

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

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

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

\AtEveryPstart*{
  \if@printingscolumns
    \vspace{0.125\baselineskip}
  \else
    \vspace{0.25\baselineskip}
  \fi
}

5  Keeping translation in the outside/inside column

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

The \Leftside environment will correspond to the inner column, and the \Rightside environment will correspond to the outer column.
6 Facing pages

6.1 Basic usage

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.

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

```latex
\begin{pages}
  \begin{Leftside} reledmac numberied text commands \end{Leftside}
  \begin{Rightside} reledmac numbered text commands \end{Rightside}
  \begin{Leftside} reledmac numbered text commands \end{Leftside}
  \ldots
  \end{pages}
\Pages
```

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

6.2 Setting

6.2.1 Text width

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

6.2.2 Way of synchronizing

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

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

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

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

The aim of the default setting is twofold:

---

*There is a French version of this article on [http://geekographie.maieul.net/185](http://geekographie.maieul.net/185)*
6.2 Setting

- To ensure that left pages contain what is to be on left side and that right pages contain what is to be on right side.

- To ensure that every chunk starts on the page that is facing its corresponding chunk.

As regards the latter, \texttt{reledpar} checks that both of the following rules are respected:

- The numbers of lines of every pair of chunks must be identical. To keep this rule, \texttt{reledpar} may insert some blank lines at the bottom of the chunk that is shorter so that it may eventually have the same number of lines as the one that is longer.

- The main content of two facing pages, apart from critical and familiar footnotes, must have the same numbers of lines, including those that may be blank. Consequently, if one left page contains more notes than the corresponding right page, the bottom of the right page must be left blank.

Each of these rules can be modified by a number of optional synchronization settings in \texttt{reledpar}:

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

   (a) 'shiftedpstarts' setting merely moves any added blank lines from the bottom of the chunks to the bottom of the page. It does not allow to have more lines on a given page as it just removes the blank lines between the chunks and does nothing more. To understand better how this work, you may compare the total amounts of lines of text on a given page whether you have activated this setting or not: you will see that both amounts are the same.

   (b) 'advancedshiftedpstarts' prevents any blank lines from being inserted at the bottom of the chunks, also taking them away from the total amount of lines the page may have. This allows to get more lines on the pages. However, please note that:

   - Blank lines are taken into account as \texttt{reledpar} moves from one to the following chunk of text, so that every pair of chunks may always start on the same facing pages.

   - Consequently, blank lines continue to be taken into account in the calculation of the amount of lines a given pair of pages may have. This is why when a longer chunk runs from one page to another the shorter corresponding one also runs across pages, even if this may result in some blank vertical space being left on the first page.

2. As regards the number of lines per page, including blank ones, the \texttt{nomaxlines} setting disregards the rule that forces two facing pages to have the same numbers of lines. So it allows to have more text on the pages. Then, by a complex mechanism it is ensured that two corresponding chunks may always start on the same facing pages, provided that \texttt{shiftedpstarts} or \texttt{advancedshiftedpstarts} settings shall not be activated.

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

Please note that every change of synchronization setting resets the content of the "numbered" auxiliary files to make sure that \texttt{reledpar} does not try to make the synchronization with wrong calculations.
6.2.3 Page number

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

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

Please also read advising in 12 p. 23.

6.2.4 Page breaking

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

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

6.2.5 Right page before \Pages

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

\texttt{reledpar} provides two options to customize this (eventual) right page.

\texttt{prevpgstyle=⟨style⟩} in order to set the style of this page. A common value of ⟨style⟩ is empty. Use \texttt{prevpgstyle=empty} will suppress header and footer in this page.

Please also read advising in 12 p. 23.

\texttt{prevpnotnumbered} will make this page won’t be counted in the page counter.

6.2.6 Notes about \texttt{mainmatter}

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

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

6.3 Critical and familiar footnotes

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

6.3.1 Notes height setting

Since \texttt{reledpar v1.13.0}, long notes in facing pages can flow from left to right pages, and vice-versa.
6.3 Critical and familiar footnotes

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

So, you MUST change the default setting via \texttt{\textbackslash Xmaxhnotes} (for critical notes) and \texttt{\textbackslash maxhnotesX} (for familiar notes). Both commands are explained in \textit{reledmac} handbook (7.12.6 p. 52). As an advisable setting:

\begin{verbatim}
\AtBeginDocument{%
  \Xmaxhnotes{0.6\textwidth}
  \maxhnotesX{0.6\textwidth}
}
\end{verbatim}

6.3.2 About the numbering of familiar footnotes

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

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

6.3.3 Using \texttt{perpage} package

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

So do not set:

\begin{verbatim}
\MakePerPage{footnote}
\MakePerPage{footnoteA}
\MakePerPage{footnoteB}
\end{verbatim}

But set:

\begin{verbatim}
\MakePerPage{footnote@typeset}
\MakePerPage{footnoteA@typeset}
\MakePerPage{footnoteB@typeset}
\end{verbatim}

6.3.4 Notes for one side only

\begin{verbatim}
\onlyside
\onlysideX
\end{verbatim}

You may want to typeset notes on one side only (either left or right). Use \texttt{\textbackslash onlyside[⟨s⟩]{⟨p⟩}} to set critical notes, and \texttt{\textbackslash onlysideX[⟨s⟩]{⟨p⟩}} to set familiar notes. \texttt{⟨p⟩} must be set to

\footnote{The same applies to \LaTeX normal notes. Read\url{http://tex.stackexchange.com/a/228283/7712} for technical informations.}
L for notes to be confined only on the left side and to R for notes to be confined only on the right side.

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

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

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 \footnotenomk\{⟨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 \footnotemk 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\footnotenomk{A note.}. And so one ...
\pend
\end{numbering}
\end{Leftside}
```

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

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

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

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

### 6.5 Using line flag

Use \Xlineflag[⟨s⟩] to add right line flag (7.7 p. 19) to right critical footnotes and
\Xendlineflag{[s]} to add it to right critical endnotes.

7 Left and right texts

7.1 Environments

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

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

7.2 Numbering text lines and paragraphs

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.

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

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

\Pages
\begin{pages}
\begin{Leftside}
  \memorydump
  ...
\end{Leftside}
\begin{Rightside}
  \memorydump
\end{pages}
It is possible to insert a number at every \pstart command. You must use the \numberpstarttrue command to have it. You can stop the numbering with \numberpstartfalse.

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

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

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

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

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

### 7.3 First line number and line number increment

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

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

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

### 7.4 Lineation system

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

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

\reledmac allows you to define a \linenumberlist to explicitly define (5.3.2 p. 20) in which line the line number will be printed. In parallel typesetting, this command affects only left side. For right side, you have to define \linenumberlistR.

### 7.5 Line number style

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

\linenumberstyle is the equivalent of \reledmac \linenumberstyle right text. The starred version are for both side.
7.6 Line number margin

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

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

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

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

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

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

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

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

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

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

7.7 Line flag

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{\langle\text{flag}\rangle}`. Use `\setRlineflag{}` to empty it.
7.8 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.

The \autopar macro can be used, instead of manually inserting \pstart...\pends.

Please read relemac’s handbook [5.2.2 p. 17].

7.9 \AtEveryPstart and \AtEveryPstartCall

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

Consequently:

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

- The argument of \AtEveryPstartCall is called before every \pstart.

7.10 Language setting

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

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

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

About these two hooks, read reledmac’s handbook (5.6 p. 26).

7.12 Executing code at each page

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

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

8 Verse

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

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

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

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

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

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

• Add optional argument (in brackets) after \begin{astanza}, as the optional argument of \pstart.

• Use optional argument after the last \& as optional argument of \pend.

Like in reledmac, you could use the \sethangingsymbol command to insert a character in each hanging line. If you use it, you must run \LaTeX 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.

When using \stanzanumtrue (9.10 p. 57) in parallel typesetting, stanza counter is replaced by stanzaL counter in left side and by stanzaR counter in right side. Consequently, you can redefine \thestanzaL and \thestanzaR to change their aspect.

9 Side notes

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

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

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

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

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

10 Parallel ledgroups

10.1 General

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

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

See the following example:

\begin{pages}
  \begin{Leftside}
    \begin{numbering}
      \pstart
      \begin{ledgroup}
        ledgroup content
      \end{ledgroup}
      \pend
      \pstart
      \begin{ledgroup}
        ledgroup content
      \end{ledgroup}
      \pend
    \end{numbering}
  \end{Leftside}
  \begin{Rightside}
    \begin{numbering}
      \pstart
      \begin{ledgroup}
        ledgroup content
      \end{ledgroup}
      \pend
    \end{numbering}
  \end{Rightside}
\end{pages}
10.2 Parallel ledgroups and setspace package

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

\setparledgroupnotespacing{\singlespacing}

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

11 Sectioning commands

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

\eledsectnotoc

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

\eledsectmark

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

12 Notes about page number

If you use sameparallelnumber option (6.2.3 p. 14) or prevpgnotnumbered option (6.2.5 p. 14), please read the following paragraph if you want to manipulate page numbers manually.

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

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

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

Notes that reledpar automatically do it when you use \frontmatter and \mainmatter commands.
I Implementation overview

\LaTeX{} 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, \LaTeX{}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 \LaTeX{} 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.

\begin{verbatim}
%<*code>
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{reledpar}[2022/04/22 v2.25.3 reledmac extension for parallel texts]%

\end{verbatim}

II.2 Package’s requirement

Few commands use \texttt{xspace} command.

\begin{verbatim}
\RequirePackage{xspace}%
\end{verbatim}

II.3 Package’s options

We use \texttt{xkeyval} in order to manage options with arguments.

\begin{verbatim}
\RequirePackage{xkeyval}%
\end{verbatim}
II.4 Package’s options

II.4.1 Synchronization’s options

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

\begin{verbatim}
\def\par@this@sync@option{}%
\end{verbatim}

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

\begin{verbatim}
\ifshiftedpstarts
\newif\ifshiftedpstarts
\DeclareOptionX{shiftedpstarts}{% 
\shiftedpstartstrue%
\apptocmd{\par@this@sync@option}{shifted}{}}%
\fi
\end{verbatim}

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

\begin{verbatim}
\ifshiftedpstarts
\newif\ifadvancedshiftedpstarts
\DeclareOptionX{advancedshiftedpstarts}{% 
\advancedshiftedpstartstrue\shiftedpstartstrue%
\apptocmd{\par@this@sync@option}{advancedshifted}{}}%
\fi
\end{verbatim}

With the option \texttt{nomaxlines}, \texttt{reledpar} allows facing pages to have not the same number of lines.

\begin{verbatim}
\ifnomaxlines
\newif\ifnomaxlines%
\DeclareOptionX{nomaxlines}{% 
\nomaxlinestrue%
\apptocmd{\par@this@sync@option}{nomax}{}}%
\fi
\end{verbatim}

With the option \texttt{nosyncpstarts}, \texttt{reledpar} only alternate between left and right side, and does not try to obtain the same number of line in corresponding page.
\textbf{II Preliminaries}  

\section*{II.4.2 Position options}  
The \texttt{\ifmovecolumnspositiononrightpage} boolean is set to true if, for a right page, we want the left column become a right column and the right column become a left column.

\section*{II.4.3 Other options}  
The \texttt{parledgroup} can be called either on \texttt{reledmac} or \texttt{reledpar}.

\section*{if@sidenotesmarginpage}  
With the \texttt{sidenotesmarginpage}, a left sidenote called in a right column will be printed at the left of the page, and a right sidenote called in a left column will be printed at the right of the page.

Options related to page numbering. The booleans are defined in \texttt{reledmac}.

\section*{prevpgstyle}  
We store on \texttt{\prevpgstyle} the argument of the option \texttt{prevpgstyle}.
II.5 Determining side and category of parallel processing

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

\ifl@dpairing \ifl@dpaging \ifledRcol

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

II.6 Text’s width

\Lcolwidth \Rcolwidth

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

\newdimen\Lcolwidth
\Lcolwidth=0.45\textwidth
\newdimen\Rcolwidth
\Rcolwidth=0.45\textwidth
\newdimen\Lcolwidth@pages
\newdimen\Rcolwidth@pages

II.7 Messages

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

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

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

\led@err@TooManyPstarts \newcommand*{\led@err@TooManyPstarts}{% \reledpar@error(Too many space without printing. Some text will be lost){\@ehc}}
\newcommand{\led@err@polyglossiaTooOld}{}%  
\reledpar@error{You use the polyglossia package.}  
\MessageBreak However, reledpar requires version 2022/04/10 v1.55 or later of the polyglossia package.  
\MessageBreak Please update your polyglossia version}{\@ehc}  

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

\providebool{syntax@}  
\newcommand*{\led@err@LeftOnRightPage}{}%  
\notbool{syntax@}{}%  
\reledpar@error{The left page has ended on a right page}{\@ehc}  

\newcommand*{\led@err@RightOnLeftPage}{}%  
\notbool{syntax@}{}%  
\reledpar@error{The right page has ended on a left page}{\@ehc}  

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

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

\newcommand*{\led@err@Pages@InsideEnv}{}%  
\reledpar@error{\string\Pages\space must be called *outside* of the `pages` environment}{\@ehc}  

\newcommand*{\led@err@Columns@InsideEnv}{}%  
\reledpar@error{\string\Columns\space must be called *outside* of the `pairs` environment}{\@ehc}  

\newcommand*{\led@err@Pages@WithoutEnv}{}%  
\reledpar@error{\string\Pages\space called without previous `pages` environment}{\@ehc}  

\newcommand*{\led@err@Columns@WithoutEnv}{}%  
\reledpar@error{\string\Columns\space called without previous `pairs` environment}{\@ehc}
II.7 Messages

\newcommand*{\led@err@Pages@WithoutRightside}{% \reledpar@error{string\space Pages\space called without any preceding `Rightside` environment (or environment without content)}{\@ehc}}% \newcommand*{\led@err@Pages@WithoutLeftside}{% \reledpar@error{string\space Pages\space called without any preceding `Leftside` environment (or environment without content)}{\@ehc}}% \newcommand*{\led@err@Columns@WithoutRightside}{% \reledpar@error{string\space Columns\space called without any preceding `Rightside` environment (or environment without content)}{\@ehc}}% \newcommand*{\led@err@Columns@WithoutLeftside}{% \reledpar@error{string\space Columns\space called without any preceding `Leftside` environment (or environment without content)}{\@ehc}}% \newcommand*{\led@error@fail@patch@thepage}{% \reledpar@error{Fail to patch \string\@thepage\space command.}{\@ehc}}% \newcommand*{\led@error@fail@patch@pagenumbering}{% \reledpar@error{Fail to patch \string\pagenumbering\space command.}{\@ehc}}% \newcommand*{\led@error@note@called@onrightside}{% \reledpar@error{#1 called on right side, despite your configuring it to be for the left side only}{\@ehc}}% \newcommand*{\led@error@note@called@onleftside}{% \reledpar@error{#1 called on left side, despite your configuring it to be for the right side only}{\@ehc}}% \newcommand*{\led@error@fail@patch@@mempnum}{% \reledpar@error{Fail to patch \string\@mempnum\space command.}{\@ehc}}% \newcommand*{\led@error@fail@patch@@outputpage}{% \reledpar@error{Fail to patch \string\@outputpage\space command.}{\@ehc}}%
II.8 Naming macros

The \TeX kernel provides \texttt{@namedef} and \texttt{@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. A set of macros for creating and using “named” boxes; the macros are called after the regular box macros, but including the string “name”.

\begin{verbatim}
\providecommand*{\newnamebox}[1]{\expandafter\newbox\csname #1\endcsname}
\providecommand*{\setnamebox}[1]{\expandafter\setbox\csname #1\endcsname}
\providecommand*{\unhnamebox}[1]{\expandafter\unhbox\csname #1\endcsname}
\providecommand*{\unvnamebox}[1]{\expandafter\unvbox\csname #1\endcsname}
\providecommand*{\namebox}[1]{\csname #1\endcsname}
\end{verbatim}
The `\ifcsboxvoid` command comes on top of the etoolbox package. The first argument is a box name, the second is the (true) value, the third the (false) value. If the first argument refers to a non-existing box, or if it refers to a box that is void, the (true) value is returned. otherwise, (false) is returned.

```latex
\newcommand*{\ifcsboxvoid}[1]{%
  \ifcsname#1\endcsname%
  \expandafter\ifvoid\csname #1\endcsname%
  \expandafter\expandafter
  \expandafter\@firstoftwo
  \else%
  \expandafter\expandafter
  \expandafter\@secondoftwo
  \fi%
  \else%
  \expandafter\@firstoftwo
  \fi%
}%
```

Macros for creating and using “named” counts.

```latex
\newcommand*{\newnamecount}[1]{% 
  \expandafter\newcount\csname #1\endcsname}
\newcommand*{\usenamecount}[1]{% 
  \csname #1\endcsname}
```

III Sectioning commands

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

```latex
\newcount\section@numR
\section@numR=\z@%
```

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

```latex
\ifpst@rtedLfalse
\newif\ifpst@rtedR
```

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

```latex
\beginnumberingR
```
\newcommand*{\beginnumberingR}{% 
  \ifnumberingR 
    \led@err@NumberingStarted 
\endnumberingR 
\fi 
\global\l@dnumppstartsR \z@ 
\global\pst@rteRfalse 
\global\numberingRtrue 
\global\afterendnumberingRfalse% 
\global\advance\section@numR \@ne 
\global\absline@numR \z@ 
\gdef\normal@page@breakR{} 
\gdef\l@prev@pbR{} 
\gdef\l@prev@nopbR{} 
\global\line@numR \z@ 
\global\sublines@false 
\global\let\next@page@numR\relax 
\global\let\this@section@next@page@numR\relax% 
\global\let\sub@change\relax 
\global\last@page@numR=-10000% 
\global\stopmsdata@inserted@true% 
\global\let\@msdata@list\relax% 
\ifcontinuousnumberingwithcolumns% 
  \ifnumbering% 
    \resumenumberingR@starttrue% 
  \fi% 
\fi 
\ifnumberpstart% 
  \addtocounter{pstartR}{1}% 
\fi% 
\else% 
  \setcounter{pstartR}{1}% 
\fi% 
\message{Section \the\section@numR R}% 
\line@list@stuffR{\jobname.\extensionchars\the\section@numR R}% 
\l@dend@stuff 
\begingroup 
\initnumbering@sectcountR 
\gdef\eled@sectionsR@@{}% 
\if@noeled@sec\else% 
\makeatletter% 
\InputIfFileExists% 
{\l@auxdir\jobname.eledsec\the\section@numR R}{}% 
\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@numR R}% 
\makeatother% 
\immediate\openout\eled@sectioningR@out=\l@auxdir\jobname.eledsec\the\section@numR R\relax% 
\fi% 
}% 
}
This is the left text version of the regular \endnumbering and must follow the last text for a left text numbered section. It sets \ifpsted 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.

\def\endnumberingR{% 
  \ifnumberingR 
    \global\numberingRfalse 
    \global\afterendnumberingRtrue% 
    \normalpars 
    \ifnum\l@dnumpstartsR=0% 
      \l@ed@err\@NumberingWithoutPstart% 
    \fi% 
    \global\page@numR=\this@section@page@numR% 
    \global\last@page@numR=\this@section@last@page@numR% 
    \global\let\next@page@numR\this@section@next@page@numR% 
    \if\@dpairing 
      \global\pstedRfalse 
    \else 
      \ifx\insertlines@listR\empty\else 
        \global\noteschanged@true 
      \fi 
      \ifx\line@listR\empty\else 
        \global\noteschanged@true 
      \fi 
    \fi 
    \ifnoteschanged@ 
      \l@ed@err@NotesChanged 
    \fi 
  \else 
    \l@ed@err@NumberingNotStarted 
  \fi 
\endgroup 
\immediate\closeout\eled@sectioningR@out% }% 

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

\setsectcountR \save@section@number \newcounter{chapterR} \newcounter{sectionR} \newcounter{subsectionR} \newcounter{subsubsectionR} \newcounter{old@chapterR} \newcounter{old@sectionR} \newcounter{old@subsectionR} \newcounter{old@subsubsectionR}
These are the right text equivalents of \pausenumbering and \resumenumbering.

\pausenumberingR
\resumenumberingR
\resumenumberingR@start
\pausenumbering@page@numR
\newcount\pausenumbering@page@numR%
\newcommand*{\pausenumberingR}{%
  \ifx\this@section@next@page@numR\relax%
    \global\pausenumbering@page@numR=0%
  \else%
    \global\pausenumbering@page@numR=\this@section@next@page@numR%
  \fi%
  \endnumberingR%
  \global\numberingRtrue%
  \global\afterendnumberingRfalse%
}%
\newif\ifresumenumberingR@start
\newcommand*{\resumenumberingR}{%
  \ifnumberingR
    \global\pst@rtedRtrue
    \global\resumenumberingR@starttrue%
    \led@mess@SectionContinued{\the\section@numR R}%
    \line@list@stuffR/{\jobname\extensionchars\the\section@numR R}%
    \l@end@stuff
    \begingroup%
    \initnumbering@sectcountR%
  \else
    \led@err@NumberingShouldHaveStarted
  \fi
}
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{⟨word⟩} is the macro used to select the lineation system for right texts. Its argument is a string: either page, pstart or section.
IV Line counting

\global\bypstart@Rtrue
\unless\ifnocritical@%
\Xpstart%
\fi%
else
\def\@tempb{section}
\ifx\@tempa\@tempb
\global\bypage@Rfalse%
\global\bypstart@Rfalse%
\unless\ifnocritical@%
\Xpstart[][false]%
\fi%
else
\led@warn@BadLineation
\fi%
\fi}}
%
\lineation*
\lineation* change the lineation system for both sides.
\WithSuffix\newcommand\lineation*[1]{%
\lineation{#1}%
\lineationR{#1}%
}%
%
IV.2 Setting line number margin
\linenummargin\line@marginR
\linenummargin You call \linenummargin{⟨word⟩} to specify which margin you want your right text’s line numbers in; it takes one argument, a string. You can put the line numbers in the same margin on every page using left or right; or you can use inner or outer to get them in the inner or outer margins. You can change this within a numbered section, but the change may not take effect just when you would like; if it is done between paragraphs nothing surprising should happen.

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

It is defined only once time, in reledmac.
\newcount\line@marginR
%
By default put right text numbers at the right.
\line@marginR=\@ne
%
\linenummargin\linenummargin R\linenummargin You call \linenummarginR{⟨word⟩} to specify which margin you want your right text’s line numbers in; it takes one argument, a string. You can put the line numbers in the same margin on every page using left or right; or you can use inner or outer to get them in the inner or outer margins. You can change this within a numbered section, but the change may not take effect just when you would like; if it is done between paragraphs nothing surprising should happen.

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

It is defined only once time, in reledmac.
\newcommand{\linenummarginR}{[1]{%
\l@getline@margin{#1}%;
\linenummargin You call \linenummargin{⟨word⟩} to specify which margin you want your right text’s line numbers in; it takes one argument, a string. You can put the line numbers in the same margin on every page using left or right; or you can use inner or outer to get them in the inner or outer margins. You can change this within a numbered section, but the change may not take effect just when you would like; if it is done between paragraphs nothing surprising should happen.

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

It is defined only once time, in reledmac.
\newcommand{\linenummarginR}[1]{%
\l@getline@margin{#1}%;
\linenummargin You call \linenummargin{⟨word⟩} to specify which margin you want your right text’s line numbers in; it takes one argument, a string. You can put the line numbers in the same margin on every page using left or right; or you can use inner or outer to get them in the inner or outer margins. You can change this within a numbered section, but the change may not take effect just when you would like; if it is done between paragraphs nothing surprising should happen.

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

It is defined only once time, in reledmac.
\newcommand{\linenummarginR}[1]{%
\l@getline@margin{#1}%;
IV.2 Setting line number margin

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

\linenumOnlyPagesForColumns and `\linenumOnlyPagesForColumnsR` make the line numbers be printed only on left/right page for the left/right column.
IV.3 Setting lineation start and step

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.

\begin{verbatim}
\newcounter{firstlinenumR}
\setcounter{firstlinenumR}{5}
\newcounter{linenumincrementR}
\setcounter{linenumincrementR}{5}
\end{verbatim}

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

\begin{verbatim}
\newcounter{firstsublinenumR}
\setcounter{firstsublinenumR}{5}
\newcounter{sublinenumincrementR}
\setcounter{sublinenumincrementR}{5}
\end{verbatim}

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

\begin{verbatim}
\WithSuffix\newcommand\firstlinenum*[1]{\setcounter{firstlinenumR}{#1}\setcounter{firstlinenum}{#1}}
\WithSuffix\newcommand\linenumincrement*[1]{\setcounter{linenumincrementR}{#1}\setcounter{linenumincrement}{#1}}
\WithSuffix\newcommand\firstsublinenum*[1]{\setcounter{firstsublinenumR}{#1}\setcounter{firstsublinenum}{#1}}
\WithSuffix\newcommand\sublinenumincrement*[1]{\setcounter{sublinenumincrementR}{#1}\setcounter{sublinenumincrement}{#1}}
\end{verbatim}
And the “R” suffixed version.

\firstlinenumR
\linenumincrementR
\firstsublinenumR
\sublinenumincrementR
\newcommand\firstlinenumR[1]{% \setcounter{firstlinenumR}{#1} %}
\newcommand\linenumincrementR[1]{% \setcounter{linenumincrementR}{#1} %}
\newcommand\firstsublinenumR[1]{% \setcounter{firstsublinenumR}{#1} %}
\newcommand\sublinenumincrementR[1]{% \setcounter{sublinenumincrementR}{#1} %}
\newcommand\setRlineflag[1]{% \gdef\@Rlineflag{#1} %}
\setRlineflag{R}

\IV.4 Setting line flag

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

\newcommand\setRlineflag[1]{% \gdef\@Rlineflag{#1} %}
\setRlineflag{R}

\IV.5 Setting line number style

\linenumrepR \sublinenumrepR
\linenumrepR{⟨ctr⟩} typesets the right line number ⟨ctr⟩, and similarly \sublinenumrepR for subline numbers.

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

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

\newcommand*{\linenumberstyleR}[1]{% \def\linenumrepR##1{\@nameuse{@#1}{##1}} %}
\newcommand*{\sublinenumberstyleR}[1]{% \def\sublinenumrepR##1{\@nameuse{@#1}{##1}} %}

\linenumberstyle* \sublinenumberstyle*
And for both side.

\WithSuffix\newcommand\linenumberstyle*[1]{% \linenumberstyle[#1] % \linenumberstyleR[#1] %}
\WithSuffix\newcommand\sublinenumberstyle*[1]{% \sublinenumberstyle[#1] % \sublinenumberstyleR[#1] %}
IV.5 Print marginal line number

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

\newif\iflinenumberLevenifblank
\newif\iflinenumberRevenifblank

\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 \@dlinenumR.

\newcommand*{\leftlinenumR}{\@dlinenumR\kern\linenumsep\numlabfont\unless\iflinenumannotationotherside\ifdefstring{\Xlinenumannotationposition@side}{before}{}\l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{\csuse{annotR@\the\absline@numR @\the\section@numR}}\linenumrepR{\line@numR}\@Rlineflag\ifsublines@R\ifnum\subline@numR>\z@\unskip\fullstop\sublinenumrepR{\subline@numR}\fi\fi\unless\iflinenumannotationotherside\ifdefstring{\Xlinenumannotationposition@side}{after}{}\l@wrap@ifnotemptybox{\Xwraplinenumannotation@side}{\csuse{annotR@\the\absline@numR @\the\section@numR}}\fi}%

IV.7 Line-number counters and lists

IV.7.1 Correspond to those in \texttt{reledmac} for regular or left text

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

The count \texttt{\line@numR} stores the line number that is used in the right text’s marginal line numbering and in notes. The count \texttt{\subline@numR} stores a sub-line number that qualifies \texttt{\line@numR}. The count \texttt{\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.

\begin{verbatim}
\newcount\line@numR \newcount\subline@numR \newcount\absline@numR
\end{verbatim}

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 \texttt{reledmac} manual.

Here are the commands to create these lists:

\begin{verbatim}
\list@create{\line@listR} \list@create{\insertlines@listR} \list@create{\actionlines@listR} \list@create{\actions@listR} \list@create{\annot@listR}
\end{verbatim}

The right text page numbers.

\begin{verbatim}
\newcount\page@numR \newcount\this@section@page@numR
\end{verbatim}
IV.7.2 Specific to reledpar

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.

\list@create{\linesinpar@listL}
\list@create{\linesinpar@listR}
\list@create{\maxlinesinpar@list}

IV.8 Reading the line-list file

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

\newcommand{\list@clearing@regR}{\%
\list@clear{\line@listR}%%
\list@clear{\insertlines@listR}%%
\list@clear{\actionlines@listR}%%
\list@clear{\actions@listR}%%
\list@clear{\linesinpar@listR}%%
\list@clear{\linesonpage@listR}%%
}\%

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

\newcommand{\@par@sync@option}[1]{\%
\IfStrEq{#1}{\@par@this@sync@option}{\%
{\ifledRcol\%
\led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}%%
\else%%
\led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}%%
\fi%%
\endinput}%%
}\%

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

IV.9 Commands within the line-list file

This section defines the commands that can appear within a line-list file, except for \@lab which is in a later section among the cross-referencing commands it is associated with.
The macros with action in their names contain all the code that modifies the action-code list.

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

\newcommand{\@nl@regR}{%
\ifx\l@dchset@num\relax \else
\advance\absline@numR \@ne
\csedef{\@dchset@num@R\the\absline@numR}{}%To remember this line have been marked by a \setlinenum
\set@line@action
\let\l@dchset@num\relax
\advance\absline@numR \m@ne
\advance\line@numR \m@ne% % do we need this?
\fi
\reset@current@annot%
\advance\absline@numR \@ne
\ifx
ext@page@numR\relax \else
\page@action
\let\next@page@numR\relax
\fi
\ifx\sub@change\relax \else
\ifnum\sub@change>\z@ \sublines@Rtrue
\else \sublines@Rfalse
\fi
\sub@action
\let\sub@change\relax
\fi
\ifcase\@lockR
\or \@lockR \tw@
\or\or
\@lockR \z@
\fi
\ifcase\sub@lockR
\or \sub@lockR \tw@
\or\or
\sub@lockR \z@
\fi
\ifs@lines@R
\ifnum\sub@lockR<\tw@
\advance\subline@numR \@ne
\fi
\else
\ifnum\@lockR<\tw@
\advance\line@numR \@ne \subline@numR \z@\fi
\fi
\fi}
\last@page@numR holds the page number of the last right page. \this@section@last@page@numR holds the page number of the last right page of the current section. Its value is modified globally, contrary to \last@page@numR. Both are modified by \fix@page, defined by \reledmac.

\newcount\last@page@numR
\last@page@numR=-10000
\newcount\this@section@last@page@numR
\this@section@last@page@numR=-10000%

\@adv\ The \@adv\(\langle num\rangle\) 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\(\langle num\rangle\) 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\(\langle num\rangle\) macro sets the line number for the next \pstart... to the value specified as its argument. This is used to implement \setlinenum. It is defined in \reledmac.

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

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

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

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

\newcount\@lockR
\newcount\sub@lockR
\newcommand*{\do@lockonR}{%}
\xright@appenditem{\the\absline@numR}\to\actionlines@listR
\ifsublines@R
\xright@appenditem{-1005}\to\actions@listR
\ifnum\sub@lockR=\z@
 \sub@lockR \@ne
 \else
 \ifnum\sub@lockR=\thr@@
 \sub@lockR \@ne
 \fi
 \fi
\else
 \xright@appenditem{-1003}\to\actions@listR
\ifnum\@lockR=\z@
IV.9 Commands within the line-list file

\lockR \one
\else
  \ifnum\lockR=\textwidth
    \lockR \one
  \fi
\fi
}\fi

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

\do@lockoffR
\skip@lockoff
\newcommand{\do@lockoffR}{%
  \xright@appenditem{\the\absline@numR}\to\actionlines@listR
  \ifsublinesR
    \xright@appenditem{-1006}\to\actions@listR
  \else
    \sub@lockR \textwidth
  \fi
  \ifnum\sub@lockR=\textwidth
    \sub@lockR \textwidth
  \else
    \sub@lockR \z\textwidth
  \fi
  \fi
  \ifnum\@lockR=\textwidth
    \@lockR \textwidth
  \else
    \@lockR \z\textwidth
  \fi
}\fi
%

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

  • #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.

\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 or \refR, depending whether we are in left or right side. Here, we define only \refR, \ref is already defined in reledmac.

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

\begin{verbatim}
\newcommand*{\@refR}[2]{
\global\advance\@edtext@level by 1\% 
\global\insert@countR=#1\relax 
\loop\ifnum\insert@countR>\z@ 
\xright@appenditem{\the\absline@numR} to \insertlines@listR 
\global\advance\insert@countR \m@ne \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 execute its argument, so that nested \@ref commands are just skipped this time. Some other macros need to be temporarily redefined to suppress their action.
\begingroup
\let\@ref=\dummy@ref
\let\@lopR\@gobble
\let\page@action=\relax
\let\sub@action=\relax
\let\set@line@action=\relax
\let\@lab=\relax
\let\@lemma=\relax
\let\@sw\@gobblethree%
\let\store@annot@to@absline\@gobble%
#2
\global\endpage@num=\page@numR
\global\endline@num=\line@numR
\global\endsubline@num=\subline@numR
\global\let\endcurrent@annot=\current@annot%
\endgroup

Now store all the information about the location of the lemma's start and end in \line@list@R.
\xright@appenditem%
{%the\page@numR|the\line@numR|%
 \ifsublinesR the\subline@numR else 0\fi% 
 \the\endpage@num\the\endline@num%
 \ifsublinesR the\endsubline@num else 0\fi} to \line@listR
\xright@appenditem%
{%\current@annot|\endcurrent@annot} to annot@listR%
%
Create a list which will store all the second argument of each \@sw in this lemma, at this level.
\end{verbatim}
And now, call \ref{\ref{parsearg}}{\ref{reg}{\ref{parse}}}, which can be also called by \ref{\reg{\ref{parse}}}{\ref{later}}

\ref{\ref{reg}{\ref{parse}}}{\ref{#2}}%

Decrease edtext level counter.
\global\advance\@edtext@level by -1%
}

\ref{\ref{reg}{\ref{parse}}}
\newcommand{\ref{\ref{reg}{\ref{parse}}}}[1]{%

Declare and init boolean for lemma in this level.
\providebool{lemmacommand@\the\@edtext@level}%
\boolfalse{lemmacommand@\the\@edtext@level}%

Execute the second argument of \ref again, to perform for real all the commands within it.
#1%

% Now, we store the list of \protect\cs{sw} of this current \protect\cs{edtext} as an element of
% the global list of list of \protect\cs{sw} for a \protect\cs{edtext} depth.
% \begin{macrocode}
\ifnum\@edtext@level>0%
\def\create@this@edtext@level{\expandafter\list@create\expandafter{\csname sw@list@edtext@\the\@edtext@level\endcsname}}%
\ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level}{}%
\letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
\letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}%
\xright@appenditem{\expandonce\@tmpp}\to\@tmp%
\global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
\fi%
\end{macrocode}
%
\end{verbatim}

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

\@pend{⟨num⟩} \@pend{\@pend} \@pendR \@pend{\@pendR}

\renewcommand*{\@pend}{[1]{% 
\ifbypstart\%
\unless\ifinstanza\%
\global\line@num=0\%
\fi%
\fi%
\xright@appenditem{\expandonce\@tmp}\to\@tmp%
\global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
}}%

\renewcommand*{\@pendR}{[1]{% 
\ifbypstartR}%
IV Line counting

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

\newcommand{\@pstart}[3]{%
   \ifcsdef{minpage@pstart@#1}{%
      \ifnumgreater{#2}{\csuse{minpage@pstart@#1}}{%
         \csnumgdef{minpage@pstart@#1}{#2}}%
      {\csnumgdef{minpage@pstart@#1}{#2}}%
   }%
   \csnumgdef{afterlines@pstart@#1L}{#3}%
}%

\newcommand{\@pstartR}[3]{%
   \numdef{\@tmp}{#2-1}%Because we have not to know in which page the pstart starts, but in which pair of facing page
   \ifcsdef{minpage@pstart@#1}{%
      \ifnumgreater{\@tmp}{\csuse{minpage@pstart@#1}}{%
         \csnumgdef{minpage@pstart@#1}{\@tmp}}%
      {\csnumgdef{minpage@pstart@#1}{\@tmp}}%
   }%
   \csnumgdef{afterlines@pstart@#1R}{#3}%
}%

\@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.

\renewcommand*{\@lopL}[1]{%
   \xright@appenditem{#1}\to\linesonpage@listL}
\renewcommand*{\@lopR}[1]{%
   \xright@appenditem{#1}\to\linesonpage@listR}

IV.10 Writing to the line-list file

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

\linenum@outR The file for right texts will be opened on output stream \linenum@outR.
Once any file is opened on this stream, we keep it open forever, or else switch to another file that we keep open.

This is the right text version of the \line@list@stuff{⟨file⟩} macro. It is called by \beginnumberingR and performs all the line-list operations needed at the start of a section. Its argument is the name of the line-list file. \relemac and \relepar can fill the \next@line@list@stuffR hook between a \endnumberingR (associated with numbered file \textit{n}) and a \beginnumberingR (associated with numbered file \textit{n + 1}). It allows adding content to the numbered file \textit{n + 1}R and not \textit{n}R.

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

`\edtext` 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`. 

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

\texttt{newseries@par} also initializes commands which are linked to notes series.

V.1.1 Notes to be printed on one side only

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

V.1.2 Tools specific to familiar footnotes

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

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

1. A \LaTeX counter \texttt{footnoteX}. This the only one manipulated by user, and the only one finally printed.

2. A \LaTeX counter \texttt{footnoteX@reading}. Its value is incremented when reading the \texttt{footnoteX} command in left or right side environments. It is used to get the correct footnote number from the .aux file to be typeset in the main text. This counter is already defined in \texttt{reledmac}, as it is also used for hyperlink.

3. A \LaTeX counter \texttt{footnoteX@typeset}. Its value is increased when inserting footnotes. Its value is directly used in the footnote, and is stored in the aux files to be used on the next run for the main text.

So here, we only defined the new counter.
\newcounter{footnote\#1@typeset} %

\Familiar footnotes without marks\ The \texttt{\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 \texttt{\nomark@} to true, then call the \texttt{\footnoteX}, and finally add the footnote counter in the footnote counter list.

First, check the \texttt{nofamiliar} option of \texttt{reledmac}. So declare the list.

\begin{verbatim}
\expandafter\list@create\csname footnote\#1@mk\endcsname %
\end{verbatim}

Then, declare the \texttt{\footnoteXnomk} command.

\begin{verbatim}
\expandafter\newcommand\csname footnote\#1nomk\endcsname[1]{%
\toggletrue{nomk@}%
\csuse{footnote\#1}{##1}%
\togglefalse{nomk@}%
%
% First step: just call the normal \texttt{\footnoteX}, saying that we do not want to print the mark.
\end{verbatim}

\begin{verbatim}
\letcs{\old@footnote}{c@footnote\#1}%
\setcounter{footnote\#1}{\csuse{c@footnote\#1}}%
%
% Second, and last, step: store the footnote counter in the footnote counters list. We use some \texttt{\let}, because \texttt{\xright@appenditem} is difficult to use with \texttt{\expandafter}.\end{verbatim}

\begin{verbatim}
\letcs{\@tmp}{footnote\#1@mk}%
\global\xright@appenditem{\csuse{c@footnote\#1}}{\@tmp}%
\global\cslet{footnote\#1@mk}{\@tmp}%
%
% Then, declare the command which inserts the footnotemark in the right side.
\end{verbatim}

\begin{verbatim}
\expandafter\newcommand\csname footnote\#1mk\endcsname{%
\letcs{\@tmp}{footnote\#1@mk}%
\edef@tmpa{\csuse{c@footnote\#1}}%
% \gl@p\@tmpa\to\@tmp%
\global\cslet{footnote\#1@mk}{\@tmp}%
%
% Set the footnotecounter with it. For the sake of security, we make a backup of the previous value.
\end{verbatim}

\begin{verbatim}
\letcs{\old@footnote}{c@footnote\#1}%
\setcounter{footnote\#1}{\@tmp}%
%
% Define the footnote mark and print it
\end{verbatim}
V.2 Create hooks

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

```latex
\unless\ifnocritical@%
\newhookcommand{series}{Xonlyside}%
\fi%
\unless\ifnofamiliar@%
\newhookcommand{series}{onlysideX}%
\fi%
```

V.1.3 Get correct footnote number

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

```latex
\newcommand{\save@familiarfootnote@number}{%
\unless\ifnofamiliar@
\def\do##1{%\csxdef{saved@footnote##1}{\the\csname c@footnote##1\endcsname}}%
dolistloop{\@series}%
\fi%}
\newcommand{\get@familiarfootnote@number}{%
\unless\ifnofamiliar@
\def\do##1{%\setcounter{footnote##1@typeset}{\csuse{saved@footnote##1}}}
dolistloop{\@series}%
\setcounter{footnote@typeset}{\saved@footnote}%
\fi%}
```

```latex
\protected@csxdef{\thefnmark#1}{\csuse{thefootnote#1}}%
\csuse{@footnotemark#1}%
\setcounter{footnote#1}{\old@footnote}%
\xspace%
\fi%
```

Restore previous footnote counter and finally add space.

End of tools specific to familiar notes.

End of \texttt{\newseries@par}.

End of tools specific to familiar notes.
V.3  Init standards series (A,B,C,D,E,Z)

\init@series@par

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

\newcommand{\init@series@par}{%
\def{do##1}{\newseries@par{##1}}%
\dolistloop{\@series}%
}%
\init@series@par%

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

As users can use classical footnotes of \LaTeX (\footnote) in parallel texts, we must integrate the same tools to get correct number as for reledmac' footnotes [V.1.2 p. 51].

\footnote@reading
\footnote@typeset

\newcount{footnote@reading}%
\newcounter{footnote@typeset}%

VI  Pstart numbers dumping and restoration

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

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

\list@pstartL@pc
\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
VII Parallel environments

The initial set up for parallel processing is deceptively simple.

\newenvironment{pairs}{%}
\l@dpairingtrue
\l@dpagingfalse
\initnumbering@quote
\save@familiarfootnote@number%
%\if@ledgroup%
\get@familiarfootnote@number%
%\fi%
\save@section@number%
%\at@begin@pairs%
(%
%\l@dpairingfalse
%)
%
\textbf{\texttt{\AtBeginPairs}}\hspace{1em}The \texttt{\AtBeginPairs} macro just define a \texttt{\at@begin@pairs} macro, called at the beginning of each pairs environments.

\begin{verbatim}
\newcommand{\AtBeginPairs}[1]{\gdef\at@begin@pairs{#1}}%
}\def\at@begin@pairs{}%
\%
\end{verbatim}

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.

\begin{verbatim}
\newenvironment{pages}{%\l@dpairingtrue\l@dpagingtrue\initnumbering@quote\save@familiarfootnote@number%\if@ledgroup%
\get@familiarfootnote@number%
\fi%\save@section@number%
\Lcolwidth=\textwidth\Rcolwidth=\textwidth%\}%\l@dpairingfalse\l@dpagingfalse\global\Lcolwidth@pages=\Lcolwidth\%Does not use \setlength, in case of calc package redefinition (cf. https://github.com/latex3/latex2e/issues/58)\global\Rcolwidth@pages=\Rcolwidth%}{%\l@dpairingfalse\l@dpagingfalse\global\Lcolwidth@pages=\Lcolwidth%\}
\%
\end{verbatim}

\textbf{\texttt{Leftside}}\hspace{1em}Within the pairs and pages environments the left and right hand texts are within \texttt{Leftside} and \texttt{Rightside} environments, respectively. The \texttt{Leftside} environment is simple, indicating that right text is not within its purview and using some particular macros.

\begin{verbatim}
\newenvironment{Leftside}{%\expandafter\ifvoid\csname l@dLcolrawbox1\endcsname\else%\led@err@Leftside@PreviousNotPrinted%\fi%\ledRcolfalse\setcounter{pstartL}{1}\let\pstart\pstartL\let\thepstart\thepstartL\let\pend\pendL\let\memorydump\memorydumpL\Leftsidehook\let\old@startstanza\@startstanza%\def\@startstanza[#1][#2]{\global\instanzaLtrue\old@startstanza[#1][#2]}%\expandafter\ifvoid\csname l@dLcolrawbox1\endcsname%}{%\expandafter\ifvoid\csname l@dLcolrawbox1\endcsname%
\end{verbatim}
Hooks into the start and end of the Leftside and Rightside environments. These are initially empty.

\newcommand*{\Leftsidehook}{ }
\newcommand*{\Leftsidehookend}{ }
\newcommand*{\Rightsidehook}{ }
\newcommand*{\Rightsidehookend}{ }

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

\newcommand*{\Rightpagehook}{ }
\newcommand*{\Leftpagehook}{ }

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.
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

Here are numbers and flags that are used internally in the course of the paragraph decomposition.

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

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

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.

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.
VIII Paragraph decomposition and reassembly

Dump the optional arguments
\provide toggles{before@pstartL@the\l@dnumpstartsL @par}\
\ifboolexpr{\test {\ifstrempty{#1}}\and \test {\ifstrempty{#2}}}{\%}{\%}
\ifat@every@pstart@star@\global\togglesfalse{before@pstartL@the\l@dnumpstartsL @par}\else\global\togglestrue{before@pstartL@the\l@dnumpstartsL @par}\fi\csgdef{before@pstartL@the\l@dnumpstartsL}{\at@every@pstart}%
\ifstrempty{#1}{}\ifstrempty{#2}{}\csgdef{before@pstartL@the\l@dnumpstartsL}{\noindent#1}\fi\global\togglesfalse{before@pstartL@the\l@dnumpstartsL @par}\csgdef{before@pstartL@the\l@dnumpstartsL}{#2}\global\togglesfalse{before@pstartL@the\l@dnumpstartsL @par}\global\by@autoparfalse\ignorespaces\%{let \by@autopar to FALSE.}\global\by@autoparfalses%\%Gobble following space (automatically done if there is no optional argument)\igorespaces}\%The same for right side.\newcommandx*{\pstartR}[2][1,2,usedefault]{\ifnobreak\let@oldnobreak\nobreaktrue\else\let@oldnobreak\nobreakfalse\fi\nobreaktrue\ifluatex\xdef\l@luatextextdir@R{\the\textdir}\xdef\l@luatexpardir@R{\the\pardir}\xdef\l@luatexbodydir@R{\the\bodydir}\fi\ifnumberingR \else\led@err@PstartNotNumbered\beginnumberingR%
VIII.1 Boxes, counters, \pstart and \pend

\fi%
\ifnumberedpar%
\led@err@PstartInPstart%
\PENDR%
\fi%
\ifpst@rte
\list@clear{\insert@listR}%
\global\let\next@insertR=\empty%
\global\pst@rteRtrue%
\fi%
\begingroup\everypar{}%
\global\advance\l@dnumpstartsR \@ne%
\ifnum\l@dnumpstartsR>\l@dc@maxchunks%
\led@err@TooManyPstarts%
\global\l@dnumpstartsR=\l@dc@maxchunks%
\fi%
\global\setnamebox{l@d@colrawbox\the\l@dnumpstartsR}=\vbox{group%
\l@dzeropenalties%
\ifboolexpr{\bool{autopar}and\bool{by@autopar}}{
\ifnumberpstart%
\ifsidepstartnum\else%
\thepstartR%
\fi%
\fi%
\if@at@every@pstart%
\hsize=\Rcolwidth%
\numberedpartrue%
\iflabelpstart%
\protected@edef\@currentlabel{\p@pstartR\thepstartR}%
\fi%
\providetoggle{before@pstartR@par}%
\ifboolexpr{
\test{\ifstrempty{#1}}\and\test{\ifstrempty{#2}}%
\ifat@every@pstart@star%
\global\togglesfalse{before@pstartR@par}%
\global\togglesfalse{before@pstartR@par}%
\ifat@every@pstart@star%
\global\togglesfalse{before@pstartR@par}%
\global\togglesfalse{before@pstartR@par}%
\fi%
\ifat@every@pstart@star%
\global\togglesfalse{before@pstartR@par}%
\global\togglesfalse{before@pstartR@par}%
\fi%
\csgdef{before@pstartR@par}{\at@every@pstart}%
}%
\fi%
\ifstrempty{#1}{%
\global\togglesfalse{before@pstartR@par}%
\global\togglesfalse{before@pstartR@par}%
\csgdef{before@pstartR@par}{\noindent#1}%
}%
\pendL \pend must be used to end a numbered paragraph. Again we need a version that knows about left parallel texts.

\newcommandx{\pendL}[2][1,2,usedefault]{% 
  \ifnumbering \else\led@err@PendNotNumbered\fi\ifnumberedpar\else\led@err@PendNoPstart\fi% 
  \if@at@end@every@pend\endgraf\global\num@lines=\prevgraf\egroup\global\par@line=0% 
  \get@firsthalf@antilabe@wd% 
  \endgroup\ignorespaces\@oldnobreak\dump@pstartL@pc% 
  \ifnumberpstart \addtocounter{pstartL}{1}% \addtocounter{pstart}{1}% \fi\parledgroup@beforenotes@save{L}% 
  \providetoggle{after@pendL@the\l@d@numpstartsL@par}\ifboolexpr{test \ifstrempty{#1}\and test \ifstrempty{#2}}% 
  \ift@every@pend@star%
VIII.1  Boxes, counters, \pstart and \pend

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

\newcommandx{\pendR}[2][1,2,usedefault]{
  %
  \ifnumbering
    \else
      \led@err@PendNotNumbered
  \fi
  \ifnumberedpar
    \else
      \led@err@PendNoPstart
  \fi
  \at@every@pend
  \endgraf
  \global\num@linesR=\prevgraf
  \global\par@lineR=0
  \get@firsthalf@antilabe@wd
  \endgroup
  \ignorespaces
  \@oldnobreak
  \dump@pstartR@pc
  \ifnumberpstart
    \addtocounter{pstartR}{1}
  \fi
  \parledgroup@beforenotes@save{R}
  \providetoggle{after@pendR\the\l@dnumpstartsR @par}
  \ifboolexpr{test {\ifstrempty{#1}} and test {\ifstrempty{#2}}}
    %
    \ifat@every@pend@star@%
      \global\togglefalse{after@pendR\the\l@dnumpstartsR @par}%
    \else%
      \global\toggletrue{after@pendR\the\l@dnumpstartsR @par}%
    \fi%
    \csgdef{after@pendR\the\l@dnumpstartsR}{\at@every@pend}%
  %
  \else%
    %
    \global\toggletrue{after@pendR\the\l@dnumpstartsR @par}%
    \csgdef{after@pendR\the\l@dnumpstartsR}{\noindent#1}%
  %
  \fi%
  \ifstrempty{#2}{}
    \csgdef{after@pendR\the\l@dnumpstartsR}{#2}%
    \global\togglefalse{after@pendR\the\l@dnumpstartsR @par}%
\ifstrempty{#1}{}{%
    \csgdef{after@pendR@\the\l@dnumpstartsR}{\noindent#1}%
    \global\toggletrue{after@pendR@\the\l@dnumpstartsR @par}%
}\%
\ifstrempty{#2}{}{%
    \csgdef{after@pendR@\the\l@dnumpstartsR}{#2}%
    \global\togglefalse{after@pendR@\the\l@dnumpstartsR @par}%
}\%
\}
\%
\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.
\newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
\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.
\newif\ifprint@last@after@pendL%
\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
\l@drightbox
A line of left text will be put in the box \l@dleftbox, and analogously for a line of right text.
\newbox\l@dleftbox
\newbox\l@drightbox
\%
\countLline
\countRline
We need to know the number of lines processed.
\newcount\countLline
\countLline \z@%
\newcount\countRline
\countRline \z@%
\%
\@donereallinesL\@donetotallinesL\@donereallinesR\@donetotallinesR
We need to know the number of “real” lines output (i.e., those that have been input by the user), and the total lines output (which includes any blank lines output for synchronisation).
The \do@lineL macro is called to do all the processing for a single line of left text.
\texttt{\textbackslash print@line} \texttt{\textbackslash print@line} is for lines without a sectioning command. See \texttt{reledmac} definition of \texttt{\textbackslash print@line} for handbook.

\begin{verbatim}
def\print@lineL{\affixpstart@numL\l@dld@ta\add@apparatusL\l@dlsn@te\do@insidelineLhook\csuse{insidethis@\the\absline@num @\the@section@num}\global\csundef{insidethis@\the\absline@num @\the@section@num}\ifluatex\textdir\l@luatextextdir@L\fi\new@lineL\inserthangingsymbolL\l@dunhbox@line{\one@line}}%

\texttt{\textbackslash print@eledsectionL} \texttt{\textbackslash print@eledsectionL} is for line with macro code.

\begin{verbatim}
def\print@eledsectionL{\disable@edindex\addtocounter{pstartL}{-1}\ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}\ifdefstring{\@eledsectmark}{L}{}{\ledsectnomark}\numdef{\temp@}{\pstarts@typesetL-1}%
\end{verbatim}
VIII.2  Processing one line

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

\dolineLhook \dolineRhook \doinsidelineLhook \doinsidelineRhook  These high-level commands just redefine the low-level commands. They have to be used by user, without \makeatletter.
Paragraph decomposition and reassembly

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

The \do@lineR macro is called to do all the processing for a single line of right text.
The \addapparatusR macro adds apparatus for the right lines, apparatus is both footnotes and sidenotes.

\newcommand{\addapparatusR}{% 
  \if@firstlineofpageR% 
  \setXtxtbforenotesR% 
  \setXtxtbforenotesXR% 
  \global\@firstlineofpageRfalse% 
  \fi% 
  \ifdefstring{\ms@data@position}{msdata-regular}{% 
    \insert@msdataR 
    \add@insertsR 
    \add@groupbylineR 
  }% 
  \global\advance\@donetotallinesRoneR 
  \addpenaltiesR 
  \global\advance\@donereallinesRoneR 
  \fi% 
  \iflinenumberRevenifblankR 
  \setbox\l@drightbox \hb@xt@Rcolwidth{% 
    \newlineR 
    \l@demptyd@ta 
    \getline@numR 
    \affixline@numR 
    \l@dld@ta 
    \hspace*{Rcolwidth} 
    \ledrfill\l@drd@ta 
  }% 
  \else% 
  \setbox\l@drightbox \hb@xt@Rcolwidth{% 
    \ifcontinuousnumberingwithcolumnsR 
    \getline@numR 
    \fiR 
    \hspace*{Rcolwidth} 
  }% 
  \else% 
    \global\advance\@donetotallinesRoneR 
  \fi% 
  \ledRcolofalseR 
  \let\linenumrep\linenumrepL 
  \let\sublinenumrep\sublinenumrepL 
}
VIII.3 Line and page number computation

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.
The real work in the line macros above is done in \do@actions, but before we plunge into that, let is get \do@ballastR out of the way.

\newcommand*{\do@ballastR}{\global\ballast@count=\z@}
\begingroup
\advance\abline@numR \@ne
\ifnum\next@actionlineR=\abline@numR
\ifnum\next@actionR>-1001
\global\advance\ballast@count by -\c@ballast
\fi
\fi
\endgroup}
%
\l@dskipversenumberR
\do@actionsR
\do@actions@fixedcodeR
\do@actions@nextR

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.

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

\newif\if\l@dskipversenumberR
\newcommand*{\do@actions@fixedcodeR}{%\ifcase\@l@dtempcnta%
% 1001 = starting sublineation
\or% \global\sublines@Rtrue
% 1002 = ending sublineation
\or% \global\sublines@Rfalse
% 1003 = starting locking number
\or% \global\@lockR=\@ne
% 1004 = ending locking number
\or% \ifnum\@lockR=\tw@
\global\@lockR=\thr@@
\else
\global\@lockR=\z@
\fi
% 1005 = starting locking subnumber
\global\sub@lockR=\@ne
% 1006 = ending locking subnumber
\ifnum\sub@lockR=\tw@
\global\sub@lockR=\thr@@
\else
\global\sub@lockR=\z@
\fi
% 1007 = skipping numbering
\l@dskipnumbertrue
% 1008 = skipping numbering in stanza
\l@dskipversenumberRtrue
% 1009 = hiding number
\l@dhidenumbertrue
% 1010 = inserting msdata
\add@msdata%
\else%
\led@warn@BadAction
\fi%
VIII  

Paragraph decomposition and reassembly

\newcommand*{\do@actionsR}{% 
  \global\let\do@actions@nextR=\relax 
  \@l@dtempcntb=\absline@numR 
  \ifnum\@l@dtempcntb<\next@actionlineR\else 
    \ifnum\next@actionR>-1001\relax 
      \ifboolexpr{% 
        bool{resumenumberingR@start}% 
        and test {ifdimgreater{pagedepth}{\z@}}% 
      }{% 
        \{% 
        @@firstlineofpageRtrue% 
        \global\page@numR=\next@actionR 
        \ifcsdef{reset@lineR\the\absline@numR \the\section@numR}% 
          {% 
            \global\line@numR=\z@ \global\subline@numR=\z@% 
          }% 
        }% 
      \global\resumenumberingR@startfalse% 
      \add@msdata@firstlineofpage% 
    \else 
      \ifnum\next@actionR<-4999\relax% 
        \@l@dtempcnta=-\next@actionR 
        \advance\@l@dtempcnta by -5001\relax 
      \fi 
      \else 
        \@l@dtempcnta=-\next@actionR 
        \advance\@l@dtempcnta by -1000\relax 
        \do@actions@fixedcodeR 
      \fi 
    \fi 
  \else 
    \@l@dtempcnta=-\next@actionR 
    \advance\@l@dtempcnta by -1000\relax 
    \do@actions@nextR} 
}\do@actions@nextR

\l@dcalcnum 
\ch@cksub@l@ckR 
\ch@ck@l@ckR 
\f@x@l@cksR 
\affixline@numR is the right text version of the \affixline@num macro.

VIII.4  Line number printing
\newcommand*{\l@dcalcnum}[3]{% 
\ifnum #1 > #2\relax 
\@l@dtempcnta = #1\relax 
\advance\@l@dtempcnta by -#2\relax 
\divide\@l@dtempcnta by #3\relax 
\multiply\@l@dtempcnta by #3\relax 
\advance\@l@dtempcnta by #2\relax 
\else 
\@l@dtempcnta=#2\relax 
\fi} 
\newcommand*{\ch@cksub@l@ckR}{% 
\ifcase\sub@lockR 
\or 
\ifnum\sublock@disp=\@ne 
\@l@dtempcntb \z@ \@l@dtempcnta \@ne 
\fi 
\or 
\ifnum\sublock@disp=\tw@ 
\else 
\@l@dtempcntb \z@ \@l@dtempcnta \@ne 
\fi 
\or 
\ifnum\sublock@disp=\z@ 
\@l@dtempcntb \z@ \@l@dtempcnta \@ne 
\fi 
\fi} 
\newcommand*{\ch@ck@l@ckR}{% 
\ifcase\@lockR 
\or 
\ifnum\lock@disp=\@ne 
\@l@dtempcntb \z@ \@l@dtempcnta \@ne 
\fi 
\or 
\ifnum\lock@disp=\tw@ 
\else 
\@l@dtempcntb \z@ \@l@dtempcnta \@ne 
\fi 
\or 
\ifnum\lock@disp=\z@ 
\@l@dtempcntb \z@ \@l@dtempcnta \@ne 
\fi 
\fi} 
\newcommand*{\f@x@l@cksR}{% 
\ifcase\@lockR 
\or 
\ifnum\lock@disp=\@ne 
\@l@dtempcntb \z@ \@l@dtempcnta \@ne 
\fi 
\or 
\ifnum\lock@disp=\tw@ 
\else 
\@l@dtempcntb \z@ \@l@dtempcnta \@ne 
\fi 
\or 
\ifnum\lock@disp=\z@ 
\@l@dtempcntb \z@ \@l@dtempcnta \@ne 
\fi 
\fi}
VIII  Paragraph decomposition and reassembly

\global\sub@lockR \tw@
\or \or
\global\sub@lockR \z@
\fi

\let\linenumberlistR\empty%
\newcommand*{\affixline@numR}{%
\ifledgroupnotesR@\else\ifnumberline
\ifl@dskipnumberR
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\fi}
\global\l@dskipnumberfalse
\else
\ifsublines@R
\global\l@dskipnumberfalse
\else
\f@i
\fi
\fi}
\affixline@numR
\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.
\fi
\else
\ifboolexpr{bool \l@dprintingcolumns} and test {\ifnum\l@dnum
line@margin@columns}{\m@ne}}%  
{\l@dtempcntb=\line@margin@columns}%  
{\l@dtempcntb=\line@margin}%  
\ifnum\l@dtempcntb=\m@ne
\advance\l@dtempcntb \page@num
\fi
\fiodd\l@dtempcntb
\gdef\l@ddta{\rlap{{\leftpstartnumL}}}%
\else
\gdef\l@drtda{\rlap{{\rightpstartnumL}}}%
\fi
\fiodd\l@dtempcntb
\gdef\l@drtda{\rlap{{\rightpstartnumR}}}%
\else
\gdef\l@ddta{\llap{{\leftpstartnumR}}}%
\fi
\fiodd\l@dtempcntb
\gdef\l@drtda{\llap{{\rightpstartnumR}}}%
\else
\gdef\l@ddta{\llap{{\leftpstartnumR}}}%
\fi
\fiodd\l@dtempcntb
\gdef\l@drtda{\llap{{\rightpstartnumR}}}%
\else
\gdef\l@ddta{\rlap{{\leftpstartnumR}}}%
\fi
\fi
\newcommand*{\affixpstart@numR}{%  
\ifsidepstartnum  
\if@twocolumn  
\if@firstcolumn  
\gdef\l@ddta{\llap{{\leftpstartnumR}}}%
\else
\gdef\l@drtda{\rlap{{\rightpstartnumR}}}%
\fi
\else
\ifboolexpr{bool \l@dprintingcolumns and test {\ifnum\l@dnum
line@margin@columnsR}{\m@ne}}%  
{\l@dtempcntb=\line@margin@columnsR}%  
{\l@dtempcntb=\line@marginR}%  
\ifnum\l@dtempcntb=\m@ne
\advance\l@dtempcntb \page@numR
\fi
\fiodd\l@dtempcntb
\gdef\l@drtda{\rlap{{\rightpstartnumR}}}%
\else
\gdef\l@ddta{\llap{{\leftpstartnumR}}}%
\fi
\fiodd\l@dtempcntb
\gdef\l@drtda{\llap{{\rightpstartnumR}}}%
\else
\gdef\l@ddta{\rlap{{\leftpstartnumR}}}%
\fi
\fiodd\l@dtempcntb
\gdef\l@drtda{\llap{{\rightpstartnumR}}}%
\else
\gdef\l@ddta{\llap{{\leftpstartnumR}}}%
\fi
\fiodd\l@dtempcntb
\gdef\l@drtda{\rlap{{\rightpstartnumR}}}%
\else
\gdef\l@ddta{\llap{{\leftpstartnumR}}}%
\fi
\fi
\newcommand*{\leftpstartnumL}{%  
\ifpstartnum
\thepstartL
\kern\linenumsep\global\pstartnumfalse\fi
\newcommand*{\rightpstartnumL}{%  
\ifpstartnum\kern\linenumsep
\thepstartL
\kern\linenumsep\global\pstartnumfalse\fi
\newif\ifpstartnumR
\pstartnumRtrue
VIII.6 Add insertions to the vertical list

\newcommand*{\leftpstartnumR}{
  \ifpstartnumR\the\thepstartR
  \kern\linenumsep\global\pstartnumRfalsefi
}

\newcommand*{\rightpstartnumR}{
  \ifpstartnumR\kern\linenumsep
  \thepstartR
  \global\pstartnumRfalsefi
}

\inserts@listR\inserts@listR is the list macro that contains the inserts that we save up for one right text paragraph.

\list@create{\inserts@listR}

\add@insertsR The right text version.

\newcommand*{\add@insertsR}{%
  \global\let\add@inserts@nextR=\relax
  \ifx\inserts@listRempty \else
    \ifx\next@insertRempty
      \global\noteschangedRtrue
      \gdef\next@insertR{100000}%
    \else
      \gl@p\insertlines@listR\to\next@insertR
    \fi
  \fi
  \ifnum\next@insertR=\abslineR\gl@p\inserts@listR\to\@insertR
  \@insertR
  \global\let\@insertR=\undefined
  \global\let\next@insertR=\empty
  \global\let\add@inserts@nextR=\add@insertsR
  \fi
  \fi
}

\add@inserts@nextR

VIII.7 Penalties

\add@penaltiesL \add@penaltiesR \add@penaltiesL is the last macro used by \do@lineL. It adds up the club, widow, and interline penalties, and puts a single penalty of the appropriate size back into the paragraph; these penalties get removed by the \vsplit operation. \displaywidowpenalty and \brokenpenalty are not restored, since we have no easy way to find out where we should insert them.
In the code below, which is a virtual copy of the original \add@penalties, \num@lines is the number of lines in the whole paragraph, and \par@line is the line we are working on at the moment. The count \@l@dtempcnta 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 \nobreakdash-10000.

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

This is for a single chunk. However, as we are probably dealing with several chunks at a time, the above is nor really relevant. Peter Wilson thinks that it is likely with parallel text that there is no real need to add back any penalties; even if there was, they would have to match across the left and right lines. So, Peter Wilson ends up with the following.

\newcommand*{\add@penaltiesL}{}
\newcommand*{\add@penaltiesR}{}

\flush@notesR
The \flush@notesR macro is called after the entire right text has been sliced up and sent on to the vertical list.
IX Footnotes

IX.1 Footnotes output specific to \Pages

The \onlyside 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 \printXnotes@forpages, which replaces \printXnotes inside \Pages. Here is how we proceed:

- If notes are to be printed on both sides, we just proceed the usual way: print the foot starts for the series, then the foot group.

- If notes are to be printed in the left side, we do these prints only for even pages; if notes are to be printed in the right side, we do these prints only for odd pages.

- However, that is not enough. Because the problem does not only consists in printing notes in any particular page. It is also not to put aside room for notes in the pages where we do not want to print them. To take an example: if some note in the left side is too long by 160pt to be printed in full in the left page, we do not want to put aside 160pt a space for it in the following right page.

- To solve this problem, we change the magnification factor associated with notes before going to the next page. If we start a page where no notes are supposed to be printed, the magnification counter is set to 0. The dimension associated to footnote is set to \maxdimen, and so we can keep all the notes we want, without any break inside. We also set the note skip to 0pt. Before starting a new page where these notes are supposed to be printed, we reset these counter and skip to their default values. (About these counter, dimension and skip, read The TeXbook p. 122-125).

- In the output macro of the page where notes must NOT be printed, we store the content of the footnote box produced by \TeX to a temporary box.

- After going to the next page, before typesetting any thing in this page, we put the content of this temporary box the footnote insert box.

The code to print critical notes, when processing \Pages, called in the output routine.

\newcommand\printXnotes@forpages[1]{%

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

\par

\end{verbatim}

\footnotesize

\par

\emph{See} \url{http://tex.stackexchange.com/a/230332/7712}
Second case: notes are for one side only. First test if we are in a page where they must be printed.

\ifboolexpr{((\ifcsstring{Xonlyside@#1}{L} and not \ifnumodd{\c@page})
or\(\ifcsstring{Xonlyside@#1}{R} and \ifnumodd{\c@page}\))}{% 
  \ifcempty{Xonlyside@#1}{% 
    \csuse{#1footstart}{#1}\% 
    \csuse{#1footgroup}{#1}\% 
  }{% 
    \csuse{#1footstart}{#1}\% 
    \csuse{#1footgroup}{#1}\% 
  }% 
}{}

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

\ifboolexpr{(\ifcsstring{Xonlyside@#1}{L} and not \ifnumodd{\c@page})
or(\ifcsstring{Xonlyside@#1}{R} and \ifnumodd{\c@page})}{% 
  \csuse{#1footstart}{#1}\% 
  \csuse{#1footgroup}{#1}\% 
}{}

Then, set to not to keep room for notes in the next page. Also set to that, in the next page, notes are not to be split, using \maxdimen.

\global{\count\csuse{#1footins}=\z@\% 
  \skip\csuse{#1footins}=\z@\% 
  \dimen\csuse{#1footins}=\maxdimen\% 
}%

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

\restore@Xnotes@settings{#1}%

End of \print@Xnotes@forpages.

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

\newcommand{\restore@Xnotes@settings}{1}{% 
  \unless\ifnocritical@ 
  \global{\count\csuse{#1footins}=% \csuse{default@#1footins}\% 
    \skip\csuse{#1footins}=% \csuse{Xbeforenotes@#1}\% 
    \bgroup\% 
    \csuse{Notefontsizes@#1}\% 
    \dimen\csuse{#1footins}=% \csuse{Xmaxhnotes@#1}\% 
    \egroup\% 
    \setnamebox{#1footins@kept}=% \namebox{#1footins}\% 
  }% 
}%
And now, the same for familiar footnotes.

\newcommand{\printnotesforpages}[1]{%  
\ifsomething{onlysideX\#1}{%  \csuse{footstart#1}{#1}%  \csuse{footgroup#1}{#1}%  }%  \ifsboolexpr{%  ((\text{\ifcsstring{onlysideX\#1}{L}} \text{and} \text{not} \text{\ifnumodd{\c@page}})\text{or}\text{\ifcsstring{onlysideX\#1}{R}} \text{and} \text{\ifnumodd{\c@page}}))%  {\csuse{footstart#1}{#1}%  \csuse{footgroup#1}{#1}%  }%  \iftext{\ifcsstring{onlysideX\#1}{L}}{\text{L}}{\text{R}}{\text{\ifnumodd{\c@page}}}  {\csuse{footstart#1}{#1}%  \csuse{footgroup#1}{#1}%  }%  \global{\count\csuse{footins#1}=\z@}%  \global{\skip\csuse{footins#1}=\z@}%  \global{\dimen\csuse{footins#1}=\maxdimen}%  \restore@notesX@settings{#1}%  }%\newcommand{\restore@notesX@settings}{%  \unless{\ifnofamiliar@}%  \global{\count\csuse{footins#1}\text{=}\csuse{default@footins#1}}%  \global{\skip\csuse{footins#1}\text{=}\csuse{beforenotesX@#1}}%  \bgroup%  \csuse{Xnotefontsize@#1}%  \global{\dimen\csuse{footins#1}\text{=}\csuse{maxhnotesX@#1}}%  \egroup%  \global{\setnamebox{footins#1@kept}\text{=}\box\namebox{footins#1}}%  \fi%  }%\newcommand{\insertnotesforonlyside}{%  \def\do##1{%  \unless{\ifnocritical@}%  \ifvoid{\csuse{##1footins@kept}}%  \expandafter{\insert}\csname##1footins\endcsname%  \bgroup%  \unvnamebox{##1footins@kept}%  \egroup%  \fi%;  }%  \insertnotesforonlyside%}
IX.2 Critical footnote printed in right side

\texttt{\textbackslash edtextlater} and \texttt{\textbackslash edtextnow} are used to print critical footnotes on the right side, while referring to the text on the left side. First, we create two counters, one for \texttt{\textbackslash edtextlater} and the other for \texttt{\textbackslash edtextnow}.

\begin{verbatim}
\newcount\edtextnow
\newcount\edtextlater
\newcommand{\edtextlater}[2]{% #1 lemma, #2 critical notes
    \noexpand\edtext{#1}\noexpand\edtext{#2}
}
\end{verbatim}

First, increase the \texttt{\textbackslash edtextlater} counter.

\begin{verbatim}
\global\advance\edtextlater by \one%
\end{verbatim}

% As we are in a pseudo-\texttt{\textbackslash cs{edtext}}, we now need to get the \texttt{\textbackslash cs{sameword}} data stored in the auxiliary file for this \texttt{\textbackslash cs{edtextlater}}.

\begin{verbatim}
\advance\@edtext@level by \one%
\ifcsvoid{sw@list@edtext@\the\@edtext@level}{}
\else
    {\g@addto@macro{sw@inthisedtext}{\texttt{\textbackslash cs{\textbackslash the}\texttt{\textbackslash edtext@level}}\texttt{\textbackslash endsname}\texttt{to}\texttt{\textbackslash sw@inthisedtext}}}
\fi
\end{verbatim}

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

\begin{verbatim}
\csxdef{edtext@later@\the\edtext@later}{}\end{verbatim}

\texttt{\textbackslash edtextnow} will insert an empty \texttt{\textbackslash edtext}.
With a \lemma and \linenum defined by the current \edtextlater. Also with \sameword data gotten from the current \edtextlater

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

And the footnote command of this \edtextlater.

And now, we print the current lemma data. But we save the beginning and the starting line using the crossref mechanism. We also store information in the auxiliary file about the existence of a \edtextlater and, if required, about the use of a \lemma
We decrease the counter increased at the beginning.

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

It is defined on `reledmac`.

### XI Sidenotes

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

\begin{Verbatim}
\sidenote@marginR
\sidenotemargin*
\sidenotemarginR
\end{Verbatim}

Specifies which margin sidenotes can be in.

\begin{Verbatim}
\WithSuffix\newcommand\sidenotemargin*[1]{%\@dgetsidenote@margin(#1)%\global\sidenote@marginR=\@dtempcntb\global\sidenote@margin=\@dtempcntb}%\newcommand\sidenotemarginR*[1]{%\@dgetsidenote@margin(#1)%\global\sidenote@marginR=\@dtempcntb}%\newcount\sidenote@marginR\global\sidenote@marginR=\@ne%\ifmovecolumnspositiononrightpage%\sidenotemargin{inner}%\sidenotemarginR{outer}%\fi%
\end{Verbatim}

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

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

\begin{Verbatim}
\newdimen\@morespace@leftnote@rightcolumn\newdimen\@morespace@rightnote@leftcolumn\newcommand{\get@sidenote@morespace@columns}{%\newdimen\@morespace@leftnote@rightcolumn\newdimen\@morespace@rightnote@leftcolumn\newcommand{\get@sidenote@morespace@columns}{%\if@sidenotesmarginpage%\if@sidenotesmarginpage%\hfil%\fi%\fi%\if@sidenotesmarginpage%\if@sidenotesmarginpage%\hfil%\fi%\fi%}
\end{Verbatim}

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

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

\If\if@sidenotesmarginpage\IsFalse\else\fi

End of \get@sidenote@morespace@columns.

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

\newcommand*{\affixside@note}{}
\prepare@edindex@fornote{\the\page@numR|\the\line@numR|\the\subline@numR|\the\page@numR|\the\line@numR|\the\subline@numR|}
\def\sidenotecontent@{}%
\numgdef{\itemcount@}{0}%
\do##1{% Not print not separator before the 1st note
  \appto\sidenotecontent@{{@sidenotesep #1}}%}
  \numgdef{\itemcount@}{\itemcount@+1}%
\dolistloop{\l@dcsnotetext}
\ifnumgreater{\itemcount@}{1}{\le@err@ManySidenotes}{}
\gdef\@templ@d{}
\gdef\@templ@n{\l@dcsnotetext\l@dcsnotetext@l\l@dcsnotetext@r}
\ifx\@templ@d\@templ@n \else\fi\if@twocolumn\if@firstcolumn\setl@dlp@rbox{##1}{\sidenotecontent@}%\else\setl@drp@rbox{\sidenotecontent@}%\fi\else\ifodd\@l@dtempcntb\setl@drp@rbox{\sidenotecontent@}%\gdef\sidenotecontent@{}%\numdef{\itemcount@}{0}%\fi\fi\fi\fi\fi
XII Verse

Like in reledmac, the insertion of hangingsymbol is base on \ifinserthangingsymbol, and, for the right side, on \ifinserthangingsymbolR. Both commands also include the hanging space, to be sure the \one@line of hanging lines has the same width that the \one@line of normal lines and to prevent the column separator from shifting.

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.

\chardef\next=\catcode`\&
This is roughly an environmental form of \stanza, which treats its stanza-like contents as a single chunk.

\begin{Verbatim}
\catcode`\&=\active
\%
\newenvironmentx{astanza}[2][1,2,usedefault]{%
  \ifledRcol%
  \global\inastanzaRtrue%
  \else%
  \global\inastanzaLtrue%
  \fi%
  \catcode`\&\active
  \global\stanza@count\@ne\stanza@modulo\@ne
  \@advance\stanzanumber%
  \newdimen\parindent@beforestanza%
  \parindent@beforestanza=\parindent%Keep in memory the standard \parindent
  \ifnum\usenamecount{sza@0@}=#0%
  \let\stanza@hang\relax
  \let\endlock\relax
  \else
  \rightskip\z@ plus 1fil\relax
  \fi
  \ifnum\usenamecount{szp@0@}=#0%
  \let\sza@penalty\relax
  \fi
  \def&{%\endlock\mbox{}%\sza@penalty\global\advance\stanza@count\@ne\@astanza@line}%
  \def\&{%\@stopastanza}%
  \ifboolexpr{not test{\ifdefvoid{\at@every@stanza}}% and test{\ifstrempty{#1}}% and test{\ifstrempty{#2}}}%{
pstart\[
\]
\at@every@stanza}$\at@start@every@stanza
\@astanza@line
\@insertstanzanumber%
\let\par\relax\ignorespaces%No paragraph in verses
}%{
\global\undef\parindent@beforestanza%
\@insertstanzanumber%
\let\par\relax\ignorespaces%No paragraph in verses
}%
\%
\end{Verbatim}

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

\begin{Verbatim}
\newcommandx{\@stopastanza}[2][1,2,usedefault]{%
  \endlock\mbox{}%
  \ifboolexpr{% not test{\ifdefvoid{\at@every@stop@stanza}}% and test{\ifstrempty{#1}}%
    and test{\ifstrempty{#2}}%
    \{\pstart\[
    \]
    \at@every@stop@stanza}$\at@start@every@stop@stanza
  \{\pstart[#1][#2]$\at@start@every@stop@stanza
  \}@astanza@line
  \@insertstanzanumber%
  \let\par\relax\ignorespaces%No paragraph in verses
}%{
  \global\undef\parindent@beforestanza%
  \@insertstanzanumber%
  \%}
  \%
\end{Verbatim}
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.

Lastly reset the modified category codes.

And now, the left and right stanza counter.

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.

A flag for checking if babel has been used as a package.
\@checklang
\bbl@set@language
In \texttt{babel} the macro \texttt{\bbl@set@language{\textlangle lang\rangle}} does the work when the language \textlangle lang\rangle is changed via \texttt{\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 \texttt{\textbackslash lang} without it stripping the \.

\begin{verbatim}
\patchcmd{\bbl@set@language} %
  {\select@language{\languagename}} %
  {\edef\languagename{#1}\select@language{\languagename}} %
  {\textbackslash} %
  {} %
\end{verbatim}

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

\selectlanguage
\@duselanguage
\theledlanguageL
\theledlanguageR
\selectlanguage is a \texttt{babel} command. \texttt{\theledlanguageL} and \texttt{\theledlanguageR} are the names of the languages of the left and right texts. \texttt{\@duselanguage} is similar to \texttt{\selectlanguage}.

\begin{verbatim}
\newcommand*{\@duselanguage}[1]{}
\gdef\theledlanguageL{}
\gdef\theledlanguageR{}
\end{verbatim}

Now do the \texttt{babel} or \texttt{polyglossia} fix or, if necessary.

\AtBeginDocument{
  \@ifundefined{xpg@main@language}{% 
    \@ifundefined{bbl@main@language}{% 
      \let\l@doldselectlanguage\selectlanguage 
      \let\l@doldbbl@set@language\bbl@set@language
      \renewcommand{\selectlanguage}[1]{% 
        \l@doldselectlanguage{#1} 
        \ifledRcol \gdef\theledlanguageR{#1} \else \gdef\theledlanguageL{#1} \fi} 
    \else \l@dusedbabeltrue 
    \endgroup 
  }{% 
    \l@dusedbabelfalse 
  }% 
} %

Either \texttt{babel} has not been used or it has been used with no specified language.

Here we deal with the case where \texttt{babel} has been used. \texttt{\selectlanguage} has to be redefined to use our version of \texttt{\bbl@set@language} and to store the left or right language.
\l@duselanguage simply calls the original \selectlanguage so that theledlanguageL and theledlanguageR are unaltered.

```latex
\renewcommand*{\l@duselanguage}[1]{%
  \let\bbl@savelastskip\relax% Babel 3.61, which provokes spurious vertical space
  \let\bbl@restorelastskip\relax% Babel 3.61, which provokes spurious vertical space
  \expandafter\l@doldselectlanguage\expandafter{#1}%
}%
%
% Lastly, initialise the left and right languages to the current babel one.
\gdef\theledlanguageL{\bbl@main@language}%
\gdef\theledlanguageR{\bbl@main@language}%
%
% If use polyglossia
{%
  \@ifpackagelater{polyglossia}{2022/04/10}{\{\led@err@polyglossiaTooOld}
    %
    \let\old@otherlanguage\otherlanguage%
    \renewcommand{\otherlanguage}[2][]{%
      \selectlanguage[#1]{#2}%
      \xpg@set@language@only@aux[#1]{#2}%
      \ifledRcol \gdef\theledlanguageR{#2}%
      \else \gdef\theledlanguageL{#2}%
      \fi%
    %
    \renewcommand{\l@duselanguage}[1]{%
      \csuse{no\languagename @numbers}%
      \expandafter\@select@language\expandafter{#1}%
    %
    \xdef\theledlanguageL{\xpg@main@language}%
    \xdef\theledlanguageR{\xpg@main@language}%
%}
%
That is it.
%
%
XIV Counts and boxes for parallel texts

In sequential text, each chunk (that enclosed by \pstart ... \pend) is put into a box called raw@text and then immediately printed, resulting in the box being emptied and ready for the next chunk. For parallel processing multiple boxes are needed as printing is delayed. We also need extra counters for various things.

/maxchunks \l@dc@maxchunks The maximum number of chunk pairs before printing has to be called for. The default is 5120 chunk pairs.
\newcount\l@dc@maxchunks
\newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}
\maxchunks{5120}
%
\l@dnumpstartsL The numbers of left and right chunks. \l@dnumpstartsL is defined in \texttt{eledmac}.
\l@dnumpstartsR
\newcount\l@dnumpstartsR
%
\l@dpscL A couple of scratch counts to count left and right \texttt{pstart}, respectively.
\l@dpscR
\newcount\l@dpscL
\newcount\l@dpscR
%
\l@dsetuprawboxes This macro creates \texttt{\maxchunks} pairs of boxes for left and right chunks. The boxes are called \texttt{\ldcolrawbox1}, \texttt{\ldcolrawbox2}, etc.
\newcommand*{\l@dsetuprawboxes}{% 
\@l@dtempcntb=\l@dc@maxchunks 
\loop\ifnum\@l@dtempcntb>\z@ 
\newnamebox{\l@dLcolrawbox\the\@l@dtempcntb}
\newnamebox{\l@dRcolrawbox\the\@l@dtempcntb}
\advance\@l@dtempcntb \m@ne
\repeat}
%
\l@dsetupmaxlinecounts \l@dzeromaxlinecounts To be able to synchronise left and right texts we need to know the maximum number of text lines there are in each pair of chunks. \texttt{\l@dsetupmaxlinecounts} creates \texttt{\maxchunks} new counts called \texttt{\l@dmaxlinesinpar1}, etc., and \texttt{\l@dzeromaxlinecounts} zeroes all of them.
\newcommand*{\l@dsetupmaxlinecounts}{% 
\@l@dtempcntb=\l@dc@maxchunks 
\loop\ifnum\@l@dtempcntb>\z@ 
\newnamecount{\l@dmaxlinesinpar\the\@l@dtempcntb}
\advance\@l@dtempcntb \m@ne
\repeat}
\newcommand*{\l@dzeromaxlinecounts}{% 
\begin{group}
\@l@dtempcntb=\l@dc@maxchunks 
\loop\ifnum\@l@dtempcntb>\z@ 
\global\newnamecount{\l@dmaxlinesinpar\the\@l@dtempcntb}=\z@
\advance\@l@dtempcntb \m@ne
\repeat
\end{group}
%

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

\AtBeginDocument{\
\l@dsetuprawboxes\
\l@dsetupmaxlinecounts\
\l@dzeromaxlinecounts\
\l@dnumpstartsL=\z@\
\l@dnumpstartsR=\z@\
\l@dpscL=\z@\
\l@dpscR=\z@}

\Check@pstarts
\check@pstarts returns \pstartstrue if there are any unprocessed chunks.
\newif\if@pstarts
\newcommand*{\check@pstarts}{\@pstartstrue
\ifnum\l@dnumpstartsL>\l@dpscL \@pstartstrue \else \ifnum\l@dnumpstartsR>\l@dpscR \@pstartstrue \fi \fi}

\Checkraw@text
\checkraw@text checks whether the current Left or Right box is void or not. If one or other is not void it sets \araw@texttrue, otherwise both are void and it sets \araw@textfalse.
\newif\ifaraw@text
\newcommand*{\checkraw@text}{\araw@textfalse
\ifvbox\namebox{\l@dLcolrawbox\the\l@dpscL} \araw@texttrue \else \ifvbox\namebox{\l@dRcolrawbox\the\l@dpscR} \araw@texttrue \fi \fi}

\Write@linesinparL
\Write@linesinparR These write the number of text lines in a chunk to the section files, and then afterwards zero the counter.
\newcommand*{\@writelinesinparL}{%  
edef\next{%  
\write\linenum@out{\string@pend{\the@donereallinesL}}%  \next%  \global@donereallinesL \z@}%
\newcommand*{\@writelinesinparR}{%  
edef\next{%  
\write\linenum@out{\string@pendR{\the@donereallinesR}}%  \next%  \global@donereallinesR \z@}%
These write the pages where start the first line of a chunk.
\newcommand*{\@writepageofparL}{[0]{%  \ifnum@donereallinesL=\z@%  \edef\next{%  \write\linenum@out{\string@pstart{\the@l@dpscL}{\the@c@page}{\the@numpagelinesL}}%  \next%  \fi%}%  \newcommand*{\@writepageofparR}{[0]{%  \ifnum@donereallinesR=\z@%  \edef\next{%  \write\linenum@outR{\string@pstartR{\the@l@dpscR}{\the@c@page}{\the@numpagelinesR}}%  \next%  \fi%  \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.  
\newcommand*{\Columns}{%  \ifl@dpairing%  \led@err@Columns@InsideEnv%  \fi%  \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{l@dLcolrawbox1}}}{%  \ifcsboxvoid{l@dRcolrawbox1}{%  \Columns
\ifcsboxvoid{l@dLcolrawbox1}\%
  {\led@err@Columns@WithoutEnv}\%
  {\led@err@Columns@WithoutRightside}\%
}\%
{\led@err@Columns@WithoutLeftside}\%
{%)
{\global\l@dprintingcolumnstrue%
 {eledsection@correcting@skip=-\baselineskip% Correction for sections' titles
 {\ifnum\l@dnumpstartsL=\l@dnumpstartsR\else
 {\led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
{\fi
 {\% Start a group and zero counters, etc.
 {\begingroup
 {\l@dzeropenalties
 {\endgraf\global\num@lines=\prevgraf
 {\global\num@linesR=\prevgraf
 {\global\par@line=z@}
 {\global\par@lineR=z@}
 {\global\l@dpscL=z@}
 {\global\l@dpscR=z@}
 {\get@familiarfootnote@number%
 {\get@intercolumns@width%
 {\get@sidenote@morespace@columns%
 {\check@pstarts
 \loop\if@pstarts
 {\global\pstartnumtrue}
 {\global\pstartnumRtrue}
 {\% Increase \l@dpscL and \l@dpscR which here count the numbers of left and right chunks. Increase \pstarts@typeset@L which counts the number of \pstart typeset both in not parallel mode and in the left side of parallel mode. Also restore the value of the public pstart counters.
 {\global\advance\l@dpscL \@ne
 {\global\advance\l@dpscR \@ne
 {\global\advance\pstarts@typeset@L \@ne%
 {\restore@pstartL@pc%
 {\restore@pstartR@pc%
 {\% We print the optional argument of \pstart or the argument of \AtEveryPstart.
 {\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
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.
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.

\flush@notes
\flush@notesR
\endgroup

\global\l@dpscL=\z@
\global\l@dpscR=\z@
\global\l@dnumpstartsL=\z@
\global\l@dnumpstartsR=\z@
\global\@dprintingcolumnsfalse
\ignorespaces
\global\instanzaLfalse
\global\instanzaRfalse
)
%
%

\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).

\def\print@columnseparator{%
\ifdim\beforecolumnseparator<0pt%
\hfill%
\else%
\hspace{\beforecolumnseparator}%
\fi%
\columnseparator%
\ifdim\aftercolumnseparator<0pt%
\hfill%
\else%
\hspace{\aftercolumnseparator}%
\fi%
})%
%
%
\get@intercolumns@width
\intercolumns@width The \intercolumns@width is calculated by \get@intercolumns@width. This length depends of many parameters:
• Columns width;
• columns position;
• columns separator;
• space between columns and columns separator, which can be fixed by user or automatically calculated by \reledpar.

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

\begin{macrocode}
\newdimen\intercolumns@width
\newcommand {get@intercolumns@width} {%
  % begin(macrocode)
  \global\intercolumns@width=\z@%
  \ifdefstring {\columns@position} {C} %
  %
  \ifboolexpr{test {\ifdimless {\beforecolumnseparator} {\z@}} and test {\ifdimless {\aftercolumnseparator} {\z@}}} {
    \global\intercolumns@width=\dimexpr \columnrulewidth + \textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth)*1/2\relax
  }{
  %

  \global\intercolumns@width=\dimexpr \columnrulewidth + \beforecolumnseparator + \aftercolumnseparator \relax
  }%
}

\end{macrocode}

First case, the columns are centered.

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

\begin{macrocode}
\ifboolexpr{test {\ifdimless {\beforecolumnseparator} {\z@}} and test {\ifdimless {\aftercolumnseparator} {\z@}}} {
  \global\intercolumns@width=\dimexpr \columnrulewidth + \textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth)*1/2\relax
} {
% 
\ifboolexpr{test {\ifdimgreater {\beforecolumnseparator} {\z@}} and test {\ifdimgreater {\aftercolumnseparator} {\z@}}} {
  \global\intercolumns@width=\dimexpr \columnrulewidth + \beforecolumnseparator + \aftercolumnseparator \relax
} {
%
}

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

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

\ifboolexpr{\test{\ifdimless{\beforecolumnseparator}{\z@}}\and\test{\ifdimgreater{\aftercolumnseparator}{\z@}}}\
\global\intercolumns@width=\dimexpr\columnrulewidth + \beforecolumnseparator + (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \beforecolumnseparator) * 1/3\relax\
\}%
%

Now, we have finished with the case the columns are centered aligned.
%
%
Other case, the columns are left or right aligned.
%
%
First sub-case, the width between columns is automatically calculated.

\ifboolexpr{\test{\ifdimless{\beforecolumnseparator}{\z@}}\and\test{\ifdimlesst{\aftercolumnseparator}{\z@}}}\
\global\intercolumns@width=\dimexpr\columnrulewidth + \aftercolumnseparator + (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \aftercolumnseparator) * 1/3\relax\
\}%
%
%
The total of the width before and after column separator
\relax%
%
%
Second sub-case, the width between column is fully determined by user setting.
Third sub-case, the width before column separator is determined by user setting, but the width after column separator is automatically calculated.

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

We have finished with the case the columns are left or right aligned.
\print@leftcolumn and \print@rightcolumn print the line number of the left or right column respectively. Normally, it is defined by the content of the \Leftside or \Rightside environments. But if \movecolumnspositiononrightpage is set to TRUE, it also depends on the page number.

\newcommand{\print@leftcolumn}{%
\if\page@num\page@numR%
\@l@dtempcntb=\page@num%
\else%
\@l@dtempcntb=\page@numR%
\fi%
\ifboolexpr{%
   not bool \movecolumnspositiononrightpage%
   or test \ifnumodd{\@l@dtempcntb}%
}{%
\unhbox\l@dleftbox%
\ifhbox\@eledsectionL%
\usebox{\@eledsectionL}%
\fi%
\}
%
\unhbox\l@drightbox%
\ifhbox\@eledsectionR%
\usebox{\@eledsectionR}%
\fi%
\}
}%
%
\newcommand{\print@rightcolumn}{%
\if\page@num\page@numR%
\@l@dtempcntb=\page@num%
\else%
\@l@dtempcntb=\page@numR%
\fi%
\ifboolexpr{%
   not bool \movecolumnspositiononrightpage%
   or test \ifnumodd{\@l@dtempcntb}%
}{%
\unhbox\l@drightbox%
\ifhbox\@eledsectionR%
\usebox{\@eledsectionR}%
\fi%
\}
%
\unhbox\l@dleftbox%
\ifhbox\@eledsectionL%
\usebox{\@eledsectionL}%
\fi%
\}
}%
\checkpb@columns \checkpb@columns prevent or make pagebreaking in columns, depending on the use of \ledpb or \lednopb.

\newcommand{\checkpb@columns}{%\newif\if@pb \newif\if@nopb \IfStrEq{\led@pb@setting}{before}{\numdef{\next@absline}{\the\absline@num+1}% \numdef{\next@abslineR}{\the\absline@numR+1}% \xifinlist{\next@absline}{\l@prev@pb}{\@pbtrue}{} \xifinlist{\next@abslineR}{\l@prev@pbR}{\@pbtrue}{} \xifinlist{\next@absline}{\l@prev@nopb}{\@nopbtrue}{} \xifinlist{\next@abslineR}{\l@prev@nopbR}{\@nopbtrue}{} }{\IfStrEq{\led@pb@setting}{after}{\xifinlist{\the\absline@num}{\l@prev@pb}{\@pbtrue}{} \xifinlist{\the\absline@numR}{\l@prev@pbR}{\@pbtrue}{} \xifinlist{\the\absline@num}{\l@prev@nopb}{\@nopbtrue}{} \xifinlist{\the\absline@numR}{\l@prev@nopbR}{\@nopbtrue}{} }{\if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi \if@pb\pagebreak[4]\fi}} \%}

\columnseparator \columnrulewidth 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).
\newcommand*{\columnseparator}{\smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}} \newdimen\columnrulewidth \columnrulewidth=\z@ \%

\columnsposition \columns@position The position of the Columns in a page. Default value is R. Stored in \columns@position.
\newcommand*{\columnsposition}{1}{\xdef\columns@position{#1}} \xdef\columns@position{R} \%

\beforecolumnseparator \aftercolumnseparator \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.
\newlength{\beforecolumnseparator}\setlength{\beforecolumnseparator}{-2pt}\newlength{\aftercolumnseparator}\setlength{\aftercolumnseparator}{-2pt}
The \setwidthliketwocolumns macro is called in \begin{numbering} in a non-parallel typesetting context, to fix the width of the lines to be vertically aligned with parallel columns. It is also called at the beginning of a note’s group, if some options are enabled. The \setposition... macros are called in \begin{numbering} in a non-parallel typesetting context to fix the position of the lines. The \setnoteposition... macros are called in \xxxfootstart in a non-parallel typesetting context to fix the position of the notes block.

\newcommand{\setwidthliketwocolumns}{% 
\get@intercolumns@width% As \columns is not necessary called before \setwidthliketwocolumns 
\hsize=\dimexpr \Lcolwidth + \intercolumns@width + \Rcolwidth \relax% }
\newcommand{\setpositionliketwocolumns@L}{% 
\renewcommand{\ledrlfill}{\hfill}\ }
\newcommand{\setnotespositionliketwocolumns@L}{% 
\doinsidelinehook{\hfill}\ }
\newcommand{\setpositionliketwocolumns@C}{% 
\doinsidelinehook{\hfill}\renewcommand{\ledrlfill}{\hfill}\ }
\newcommand{\setnotespositionliketwocolumns@C}{% 
\newdimen\temp% \newdimen\tempa% \temp=\hsize% \tempa=\Lcolwidth% \advance\tempa\Rcolwidth% \advance\temp-\tempa% \divide\temp by 2% \leftskip=\temp% \rightskip=-\temp% }\ }
\newcommand{\setnotespositionliketwocolumns@R}{% 
\doinsidelinehook{\hfill}\ }

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

\newcommand{\Columns@print@before@pstart}{% 
\ifboolexpr{\test{ifcsstring{before@pstartL@the@\l@dpscL}{\at@every@pstart}}}
XVII Parallel pages

This is considerably more complicated than parallel columns.

XVII.1 Specific counters

Counts for the number of lines on a left or right page, and the smaller of the number of lines on a pair of facing pages.

\newcount\numpagelinesL
\newcount\numpagelinesR
\newcount\l@dminpagelines

XVII.2 Main macro

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.

\newcommandx*{\Pages}[1,usedefault]{% 
\iffl@dpairing% \le@d@err@Pages@InsideEnv% \fi% \ifboolexpr{test\{\ifcsboxvoid{l@dRcolrawbox1}\} or test\{\ifcsboxvoid{l@dLcolrawbox1}\}}{% \ifcsboxvoid{l@dRcolrawbox1}{% \ifcsboxvoid{l@dLcolrawbox1}% {\le@d@err@Pages@WithoutEnv}% {\le@d@err@Pages@WithoutRightside}% }% \ifcsboxvoid{l@dRcolrawbox1}% {\le@d@err@Pages@WithoutLeftside}% \ifstrequal{#1}{mainmatter}{\Pages@mainmattertrue}{\Pages@mainmatterfalse}% }% \le@dsection@correcting@skip=-2\baselineskip% line correcting for section titles. \par\le@dgroup@notespacing@set@correction\typeout{\textbf{PAGES}}% \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else% \le@d@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}\else% \le@d@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}% \fi% \global\l@dprintingpagestrue% Get onto an empty even (left) page, then initialise counters, etc.

\cleartol@devenpage%
\global\l@dprintingpagestrue%
As \texttt{\Pages} must be called outside of the pages environment, we have to redefine the \texttt{\Lcolwidth} and \texttt{\Rcolwidth} lengths, to prevent false overfull hboxes.

\begingroup
\Lcolwidth=\Lcolwidth@pages
\Rcolwidth=\Rcolwidth@pages
\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number
\l@dzeropenalties%\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number
\l@dzeropenalties%\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number

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

\let\print@Xnotes\print@Xnotes@forpages
\let\print@notesX\print@notesX@forpages
\check@pstarts\loop\if@pstarts\repeat
\global\l@dpscL=\z@\global\l@dpscR=\z@
\global\l@dmaxlinesinpar\z@
\get@familiarfootnote@number
\l@dzeropenalties%\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number
\l@dzeropenalties%\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number

Check if there are chunks to be processed.

\check@pstarts%\loop\if@pstarts%
\global\advance\l@dpscL \@ne
\global\advance\l@dpscR \@ne
\global\l@dmaxlinesinpar\z@
\get@familiarfootnote@number
\l@dzeropenalties%\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number
\l@dzeropenalties%\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number

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

\global\l@dmaxlinesinpar\z@
\get@familiarfootnote@number
\l@dzeropenalties%\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number
\l@dzeropenalties%\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number

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

\get@familiarfootnote@number
\l@dzeropenalties%\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number
\l@dzeropenalties%\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number

Zero the counts again, ready for the next bit.

\global\l@dmaxlinesinpar\z@
\get@familiarfootnote@number
\l@dzeropenalties%\endgraf
\global\num@lines=\prevgraf
\global\num@linesR=\prevgraf
\global\par@line=z@\par
\global\par@lineR=z@\par
\global\l@dpscL=z@\par
\global\l@dpscR=z@\par
\writtenlinesLfalse\writtenlinesRfalse\get@familiarfootnote@number

Get the number of lines on the first pair of pages and store the minimum in \texttt{\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.

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

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

Start a loop over the boxes (chunks).

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

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

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
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 synchronize with the facing page so use an impossibly large number.

\if@pagefull
\@writelinesonpageL{\the
umpagelinesL} \else \@writelinesonpageL{1000} \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.

\umpagelinesL \z@ \parledgroup@correction@notespacing@init \clearl@dleftpage

Now do the same for the right text.

\checkpageR \Rightpagehook
\l@duselanguage{\theledlanguageR}
{ \loop\if@samepage
\set@sectcountR
\ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{}\csuse{before@pstartR@\the\l@dpscR} \global\csundef{before@pstartR@\the\l@dpscR} \do@lineR
\xifinlist{\the\l@dpscR}{\eled@sectionsR@@}{\print@eledsectionR} \advance\numpagelinesR 1 \ifshiftedpstarts
\ifdim\ht\l@drightbox>0pt \parledgroup@correction@notespacing\l@drightbox \else \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}{\add@apparatusR} \unless\ifadvancedshiftedpstarts\dimen0=\pagetotal \advance\dimen0 by \baselineskip \global\pagetotal=\dimen0 \else}
The page is full, so move onto the next (left, odd) page and repeat left text processing.

\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.

\checkraw@text%
\ifraw@text%
\getlinesfrompagelistL%
\getlinesfrompagelistR%
\l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
{\l@dminpagelines}%
\fi%
\repeat%
% We have now output the text from all the chunks.
\textbf{XVII.3 Ensure all notes are printed at the end of parallel pages}

This macro ensures that all long notes are printed at the end of \texttt{\Pages} typesetting, and that there are no more long notes left for the next pages.

```
\newcommand{\finish@Pages@notes}{%
% \def\do##1{%
% \global\l@dpscL=\z@%
% \global\l@dpscR=\z@%
% \global\l@dnumpstartsL=\z@%
% \global\l@dnumpstartsR=\z@%
% \global\instanzaLfalse%
% \global\instanzaRfalse%
% \global\inastanzaLfalse%
% \global\inastanzaRfalse%
% \global\l@dprintingpagesfalse%
%
Check the consistency of \texttt{\edtext@later} and \texttt{\edtext@now}

```

```
\ifnum\edtext@later=\edtext@now%
% \else%
% \led@error{edtext@later@now}%
% \fi%
\global\edtext@later=\z@%
\global\edtext@now=\z@%
%
Prevent final notes from overlapping the line number

```

```
\finish@Pages@notes%
% \ignorespaces}}%
%
\textbf{XVII.3 Ensure all notes are printed at the end of parallel pages}

This macro ensures that all long notes are printed at the end of \texttt{\Pages} typesetting, and that there are no more long notes left for the next pages.

```
\newcommand{\finish@Pages@notes}{%
% \def\do##1{%
% \global\l@dpscL=\z@%
% \global\l@dpscR=\z@%
% \global\l@dnumpstartsL=\z@%
% \global\l@dnumpstartsR=\z@%
% \global\instanzaLfalse%
% \global\instanzaRfalse%
% \global\inastanzaLfalse%
% \global\inastanzaRfalse%
% \global\l@dprintingpagesfalse%
%
Check the consistency of \texttt{\edtext@later} and \texttt{\edtext@now}

```

```
\ifnum\edtext@later=\edtext@now%
% \else%
% \led@error{edtext@later@now}%
% \fi%
\global\edtext@later=\z@%
\global\edtext@now=\z@%
%
Prevent final notes from overlapping the line number

```

```
\finish@Pages@notes%
% \ignorespaces}}%
%
\textbf{XVII.3 Ensure all notes are printed at the end of parallel pages}

This macro ensures that all long notes are printed at the end of \texttt{\Pages} typesetting, and that there are no more long notes left for the next pages.

```
\newcommand{\finish@Pages@notes}{%
% \def\do##1{%
% \global\l@dpscL=\z@%
% \global\l@dpscR=\z@%
% \global\l@dnumpstartsL=\z@%
% \global\l@dnumpstartsR=\z@%
% \global\instanzaLfalse%
% \global\instanzaRfalse%
% \global\inastanzaLfalse%
% \global\inastanzaRfalse%
% \global\l@dprintingpagesfalse%
%
Check the consistency of \texttt{\edtext@later} and \texttt{\edtext@now}

```

```
\ifnum\edtext@later=\edtext@now%
% \else%
% \led@error{edtext@later@now}%
% \fi%
\global\edtext@later=\z@%
\global\edtext@now=\z@%
%
Prevent final notes from overlapping the line number

```

```
\finish@Pages@notes%
% \ignorespaces}}%
%
\textbf{XVII.3 Ensure all notes are printed at the end of parallel pages}

This macro ensures that all long notes are printed at the end of \texttt{\Pages} typesetting, and that there are no more long notes left for the next pages.

```
\newcommand{\finish@Pages@notes}{%
% \def\do##1{%
% \global\l@dpscL=\z@%
% \global\l@dpscR=\z@%
% \global\l@dnumpstartsL=\z@%
% \global\l@dnumpstartsR=\z@%
% \global\instanzaLfalse%
% \global\instanzaRfalse%
% \global\inastanzaLfalse%
% \global\inastanzaRfalse%
% \global\l@dprintingpagesfalse%
%
Check the consistency of \texttt{\edtext@later} and \texttt{\edtext@now}

```

```
\ifnum\edtext@later=\edtext@now%
% \else%
% \led@error{edtext@later@now}%
% \fi%
\global\edtext@later=\z@%
\global\edtext@now=\z@%
%
Prevent final notes from overlapping the line number

```

```
\finish@Pages@notes%
% \ignorespaces}}%
%
We must also restore the settings for the footnotes in case of using onlieside option.

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

XVII.4 Struts

\ledstrutL
\ledstrutR

Struts inserted into left and right text lines.

XVII.5 Page clearing

\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.
\clearleftpage and \clearrightpage 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.

\newcommand*{\clearleftpage}{\ifdim\pagetotal=0pt\hbox{}\fi\newpage% \insert@notes@for\onlyside% \ifodd\c@page\else \led@err@LeftOnRightPage \hbox{}\cleardoublepage \fi}
\newcommand*{\clearrightpage}{\ifdim\pagetotal=0pt\hbox{}\fi\newpage% \insert@notes@for\onlyside% \ifodd\c@page \led@err@RightOnLeftPage \hbox{}\cleartoevenpage \fi}

\getlinesfromparlistL \@cs@linesinparL
\getlinesfromparlistR \@cs@linesinparR
\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.

\newcommand*{\getlinesfromparlistL}{\ifx\linesinpar@listL\empty \gdef\@cs@linesinparL{0}\else \@gobbletwo{\linesinpar@listL}{\@cs@linesinparL}\fi}
\newcommand*{\getlinesfromparlistR}{\ifx\linesinpar@listR\empty \gdef\@cs@linesinparR{0}\else \@gobbletwo{\linesinpar@listR}{\@cs@linesinparR}\fi}
\getlinesfrompagelistL \@cs@linesonpageL \getlinesfrompagelistR \@cs@linesonpageR

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

\newcommand*{\getlinesfrompagelistL}{% 
  \ifx\linesonpage@listL\empty 
    \gdef\@cs@linesonpageL{1000}% 
  \else 
    \gl@p\linesonpage@listL\to\@cs@linesonpageL 
  \fi}
\newcommand*{\getlinesfrompagelistR}{% 
  \ifx\linesonpage@listR\empty 
    \gdef\@cs@linesonpageR{1000}% 
  \else 
    \gl@p\linesonpage@listR\to\@cs@linesonpageR 
  \fi}

\@writelinesonpageL \@writelinesonpageR

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

\newcommand*{\@writelinesonpageL}[1]{% 
  \edef\next{\write\linenum@out{\string\@lopL{#1}}}% 
  \next}
\newcommand*{\@writelinesonpageR}[1]{% 
  \edef\next{\write\linenum@outR{\string\@lopR{#1}}}% 
  \next}

% \l@dcalc@maxoftwo \l@dcalc@minoftwo

\l@dcalc@maxoftwo{⟨num⟩}{⟨num⟩}{⟨count⟩} sets ⟨count⟩ to the maximum of the two ⟨num⟩.

Similarly \l@dcalc@minoftwo{⟨num⟩}{⟨num⟩}{⟨count⟩} sets ⟨count⟩ to the minimum of the two ⟨num⟩.

\newcommand*{\l@dcalc@maxoftwo}[3]{% 
  \ifnum #2>#1\relax 
    #3=#2\relax 
  \else 
    #3=#1\relax 
  \fi}
\newcommand*{\l@dcalc@minoftwo}[3]{% 
  \ifnum #2<#1\relax 
    #3=#2\relax 
  \else 
    #3=#1\relax 
  \fi}
XVII.7  Page break managing

\ifl@dsamepage
\l@dsamepagetrue
\l@dsamepagefalse
\ifl@dpagefull
\l@dpagefulltrue
\l@dpagefullfalse
\checkpageL
\checkpageR
\\newif\ifl@dsamepage
\l@dsamepagetrue
\\newif\ifl@dpagefull
\\newcommand*{\checkpageL}{%
\l@dpagefulltrue
\l@dsamepagetrue
\check@goal
\ifdim\pagetotal<\ledthegoal
\ifnum\numpagelinesL<\l@dminpagelines
\else
  \ifnomaxlines%
    \else%
      \l@dsamepagefalse%
      \l@dpagefullfalse%
    \fi%
  \fi%
\else
  \l@dsamepagefalse
  \l@dpagefulltrue
\fi%
}\ifprint@last@after@pendL%
\l@dpagefullfalse%
\l@dsamepagefalse%
\print@last@after@pendLfalse%
\fi%
}\
ewcommand*{\checkpageR}{%
\l@dpagefulltrue
\l@dsamepagetrue
\check@goal
\ifdim\pagetotal<\ledthegoal
\ifnum\numpagelinesR<\l@dminpagelines
\else
  \ifnomaxlines%
    \else%
      \l@dsamepagefalse%
      \l@dpagefullfalse%
    \fi%
  \fi%
\else
  \l@dsamepagefalse
  \l@dpagefulltrue
\fi%}
\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.

\newcommand{\checkpbL}{
  \IfStrEq{\led@pb@setting}{after}{
    \xifinlist{\the\absline@num}{\l@prev@pb}{\l@dpagefulltrue}\l@dsamepagefalse}\{}
  \xifinlist{\the\absline@num}{\l@prev@nopb}{\l@dpagefullfalse}\l@dsamepagetrue}\{}
}
\newcommand{\checkpbR}{
  \IfStrEq{\led@pb@setting}{after}{
    \xifinlist{\the\absline@numR}{\l@prev@pbR}{\l@dpagefulltrue}\l@dsamepagefalse}\{}
  \xifinlist{\the\absline@numR}{\l@prev@nopbR}{\l@dpagefullfalse}\l@dsamepagetrue}\{}
}

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

\newcommand{\checkverseL}{%
  \ifinstanzaL%
  \iflednopbinverse%
  \ifinsertingsymbol%
  \numdef{\prev@abslineverse}{\the\absline@num-1}%
  \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{}
}
\newcommand{\setgoalfraction}[]{\def\@goalfraction{#1}}

\newcommand{\ledthegoal}{\@goalfraction\pagegoal}

\newcommand{\check@goal}{\ledthegoal=\@goalfraction\pagegoal \newcommand{\setgoalfraction}[1]{\xdef\@goalfraction{#1}}}

\newif\ifwrittenlinesL
\newif\ifwrittenlinesR

\setgoalfraction \ledthegoal \goalfractio \check@goal

\ifwrittenlinesL
\newif\ifwrittenlinesL
\newif\ifwrittenlinesR

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

\ifwrittenlinesL
%
XVII.8 Getting boxes content

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

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

\newif\if@getnextbox
\newcommand\get@nextboxL{\ifvbox\namebox{l@dLcolrawbox\the\l@dpscL}% box is not empty
\else% box is empty
\get@nextboxfalse%Should be local, but be cautious
\fi%
\unless\ifnosyncpstarts%
\ifnomaxlines%
\ifdim\pagetotal<\ledthegoal%
\numdef{\@tmp}{\l@dpscL+1}%
\ifsdef{afterlines@pstart@\@tmp R}{%
\ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp R}}%
\ifnumless{\numpagelinesL}{\csuse{minpage@pstart@\@tmp}}%
\@getnextboxtrue%
\fi%
\fi%
}}%
\{\getnextboxtrue\%
  
\}
\fi
\fi
\fi
\else
\ifnomaxlines
\numdef{\@tmp}{\the\l@dpscL+1}
\ifcsdef{minpage@pstart@\@tmp}{
\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}{
\ifdimgreater{\pagetotal}{\ledthegoal}{
\getnextboxtrue
\getnextboxfalse
\}{};
\getnextboxtrue
\}
\fi
\fi
\% Sufficient lines have been output.
\if@getnextbox
\ifnum\usenamecount{l@dmaxlinesinpar\the\l@dpscL}=@\donetotallinesL
\parledgroup@notes@endL
\fi
\ifwrittenlinesL\else
\% Write out the number of lines done, and set the boolean so this is only done once.
\writelinesinparL
\writtenlinesLtrue
\fi
\ifnum\l@dnumpstartsL>\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.
\writelinesLfalse
\ifbypstick
\global\line@num=0
\resetprevline0
\fi
\% Add the content of the optional argument of the previous \protect\cs{pend}.
\begin{macrocode}
\csuse{after@pendL@\the\l@dpscL}%
\global\csundef{after@pendL@\the\l@dpscL}%
\end{macrocode}
\% Check the number of lines
Go to the next pstart

\global\advance\l@dpscL \@ne
\global\advance\pstarts@typeset@L \@ne%
\global\pstartnumtrue%
\restore@pstartL@pc%

Add notes of parallel ledgroup.
\parledgroup@notes@endL
\parledgroup@correction@notes@spacing@final{L}
\else
\fi
\fi
\fi

\newcommand*{\get@nextboxR}{%
\ifvbox\namebox{l@dRcolrawbox\the\l@dpscR}% box is not empty
\else% box is empty
\@getnextboxtrue%
\ifnum\usenamecount{l@dmaxlinesinpar\the\l@dpscR} > \@donetotallinesR
\parledgroup@notes@endR
\unless\ifnosyncpstarts%
\@getnextboxfalse%
\ifnomaxlines%
\ifdim\pagetotal< \ledthegoal%
\numdef{\@tmp}{\l@dpscR+1}%
\ifcsdef{afterlines@pstart@\@tmp}{L}{%
\ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp}}%
\ifnumless{\numpagelinesR}{\l@dminpagelines}%
\@getnextboxtrue%
\fi%
\fi%
\@getnextboxtrue}%
\@getnextboxtrue%
\fi%
\fi%
\fi%
\fi%
The mainmatter option of \Pages

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

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

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

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

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

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.

We save the sectioning commands of the right side in the \eled@sectioningR@out file.
XX  Page break/no page break, depending on the specific line

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

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

\prev@pbR
\prev@nopbR
\l@prev@pbR
\l@prev@nopbR

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.

\ledpbR
\ledpbnumR
\lednopbR
\lednopbnumR

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.

\led@pbR
\led@pbnumR
\led@nopbR
\led@nopbnumR

XXI  Parallel ledgroup

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).

\parledgroup@notes@startL
\parledgroup@notes@startR

\parledgroup@notes@startL and \parledgroup@notes@startR are used to mark the beginning of a note series in a parallel ledgroup.
\newcommand{\parledgroup@notes@startL}{% 
  \ifnum\usenamecount{\thedmaxlinesinpar}\the{\thedpscL}>0% 
    \IfStrEq{\splitfirstmarks}{\parledgroup@type}{\footnoteX}{% 
      \csuse{bhooknoteX@\splitfirstmarks}{\parledgroup@series}}{% 
        \IfStrEq{\splitfirstmarks}{\parledgroup@type}{Xfootnote}{% 
          \csuse{bhooknote@\splitfirstmarks}{\parledgroup@series}}{% 
            \fi% 
            \global\ledgroupnotesL@true% 
            \insert@noterule@ledgroup{L}% 
  }% 
  }% 
} 
\newcommand{\parledgroup@notes@startR}{% 
  \ifnum\usenamecount{\thedmaxlinesinpar}\the{\thedpscR}>0% 
    \IfStrEq{\splitfirstmarks}{\parledgroup@type}{\footnoteX}{% 
      \csuse{bhooknoteX@\splitfirstmarks}{\parledgroup@series}}{% 
        \IfStrEq{\splitfirstmarks}{\parledgroup@type}{Xfootnote}{% 
          \csuse{bhooknote@\splitfirstmarks}{\parledgroup@series}}{% 
            \fi% 
            \global\ledgroupnotesR@true% 
            \insert@noterule@ledgroup{R}% 
  }% 
  }% 
% \parledgroup@notes@startL \parledgroup@notes@startR

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

\newcommand{\parledgroup@notes@endL}{% 
  \global\ledgroupnotesL@false% 
} 
\newcommand{\parledgroup@notes@endR}{% 
  \global\ledgroupnotesR@false% 
} 
% \parledgroup@notes@endL \parledgroup@notes@endR

\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

\newcommand{\insert@noterule@ledgroup}{% 
  \IfStrEq{\splitbotmarks}{\parledgroup@}{begin}{% 
    \IfStrEq{\splitbotmarks}{\parledgroup@type}{Xfootnote}{% 
      \csuse{ifledgroupnotes#1@}{% 
        \vskip\skip\csuse{mp\splitbotmarks}{\parledgroup@series footins} 
        \csuse{footnoterule\splitbotmarks}{\parledgroup@series}}{% 
          \fi% 
          \{}% 
    }{}% 
  }{}% 
} 
% \insert@noterule@ledgroup
\@parledgroupnotespacing
\@parledgroupnotespacing can be redefined by the user to change the interline spacing of ledgroup notes.

\newcommand{\setparledgroupnotespacing}{1}\{\gdef{\@parledgroupnotespacing}{#1}\}
\newcommand{\@parledgroupnotespacing}{}
%
\parledgroupnotespacing@correction
\parledgroupnotespacing@correction is the difference between a normal line skip and a line skip in a note. It is set by \parledgroupnotespacing@set@correction, called at the beginning of \Pages.

\dimdef{\parledgroupnotespacing@correction}{0pt}
\newcommand{\parledgroupnotespacing@set@correction}{%We suppose all the series has the same footnote size setup
\@parledgroupnotespacing\dimgdef{\temp@spacing}{\baselineskip}}%
\dimgdef{\parledgroupnotespacing@correction}{\baselineskip-\temp@spacing}%}
%
\parledgroupnotespacing@init
\parledgroupnotespacing@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.

\newcommand{\parledgroupnotespacing@init}{%\dimdef{\parledgroupnotespacing@correction@accumulated}{0pt}\dimdef{\parledgroupnotespacing@correction@modulo}{0pt}\}%

\parledgroupnotespacing@final
\parledgroupnotespacing@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.

\newcommand{\parledgroupnotespacing@final}{[1]}{%\ifparledgroup\vspace{\parledgroupnotespacing@correction@accumulated}\parledgroupnotespacing@init%\ifstrequal{#1}{L}{{\numdef{\@checking}{\the\l@dpscL-1}}}{\numdef{\@checking}{\the\l@dpscR-1}}}%\{\dimdef{\@beforenotes@current@diff}{\csuse{\@parledgroup@beforenotes@\@checking L}-\csuse{\@parledgroup@beforenotes@\@checking R}}%\ifstrequal{#1}{L}{%Left\ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{-\@beforenotes@current@diff}}%}}%
The macro \parledgroup@beforenotes@save dumps the space before notes of the current parallel ledgroup in a macro named with the current pstart number.

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

\parledgroup@correction@notespacing is used before each printed line. If it is a line of notes in parallel ledgroup, the space \parledgroup@notespacing@correction is decreased, to make interline space correct. The decreased space is added to \parledgroup@notespacing@correction@accumulated and \parledgroup@notespacing@correction@modulo. If \parledgroup@notespacing@correction@modulo is equal or greater than \baselineskip:

- It is decreased by \baselineskip.
- The total of line number in the current page is decreased by one.

For example, suppose an normal interline of 24 pt and interline for note of 12 pt. That means that the two lines of notes take the place of one normal line. For every two lines of notes, the line total for the current place is decreased by one.
XXII  Compatibility with eledmac

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

XXIII  The End

</code>
Appendix A  Some things to do when changing version

A.1 Migration to eledpar 1.4.3

Version 1.4.3 corrects a bug added in version 0.12, which made hanging verse always flush right, despite the value of the first element in the \setstanzainsdents command. However, if you want to return to automatic flushright margins for verses with hanging indents, you have to redefine the \hangingsymbol command.

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

See the following two examples:

With standard \hangingsymbol:
A very long verse should sometimes be hanging. The position of the hanging verse is fixed.

With the modification of the hangingsymbol:
A very long verse should sometimes be hanging. And we can see that a hanging verse is flush right.

A.2 Migration from eledpar to reledpar

As for migration from eledmac to reledmac:

• One option has been removed because it is deprecated.

• Some of the customizations previously made by \renewcommand have been replaced with commands.

• Some command names have been changed in order to have a more logical and uniform pattern.

A.2.1 Deprecated options

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

A.2.2 \renewcommand replaced with command

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

<table>
<thead>
<tr>
<th>Deprecated \renewcommand</th>
<th>Replaced with</th>
</tr>
</thead>
<tbody>
<tr>
<td>\goalfraction</td>
<td>\setgoalfraction</td>
</tr>
<tr>
<td>\parledgroupnotespacing</td>
<td>\setparledgroupnotespacing</td>
</tr>
<tr>
<td>\Rlineflag</td>
<td>\setRlineflag</td>
</tr>
</tbody>
</table>
A.2.3 Commands the names of which have changed

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

<table>
<thead>
<tr>
<th>Old command</th>
<th>New command</th>
</tr>
</thead>
<tbody>
<tr>
<td>\onlyXside</td>
<td>\Xonlyside</td>
</tr>
</tbody>
</table>

A.3 Migration to \texttt{reledpar} 2.2.0

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

A.4 Migration to \texttt{reledpar} 2.3.0

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

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

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

A.5 Migration to \texttt{reledpar} 2.4.0

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

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

A.6 Migration to \texttt{reledpar} 2.5.0

If you use either \texttt{\stanza} or \texttt{astanza} environment, please read A.12 p. 384.

A.7 Migration to \texttt{reledpar} 2.6.0

\texttt{\printlinenumR} was deleted. Use \texttt{\Xlineflag} instead.

A.8 Migration to \texttt{reledpar} 2.6.1

If you use \texttt{perpage} package to control footnote numbering, please read the handbook on 6.3.3 p. 15.

A.9 Migration to \texttt{reledpar} 2.14.0

The \texttt{\linenumberlist} command is reserved for the left side. For the equivalent feature on the right side, define \texttt{\linenumberlistR}. 
A.10 Migration to reledpar 2.14.1

The execution of the code sent to \doinsideLhook and \doinsideRhook has been moved to a place where \markboth can work.

Normally, it should not interfere with your prior use of \doinsideLhook and \doinsideRhook.

Please contact us if that causes problems.

A.11 Migration to reledpar v. 2.17.2

When changing the \Lcolwidth and \Rcolwidth length within an \pages environment, the contents were aligned to the right. Now, they are aligned to the left. Please contact us if you need to retain the older behaviour.

A.12 Migration to reledpar v. 2.24.3

There was an inconsistency with hanging verses. If you defined indentation of hanging verses as 2, with \setstanzaindents{2,...}, the indent was, in reality, typeset as equal to 4. This happened:

- only in parallel typesetting;
- only with the \stanza macro, not with the \astanza environment.

This inconsistency has been corrected. If you want to keep old indentation after upgrading the package, just multiply by two the first argument of \setstanzaindents.

References


[Wil04] Peter Wilson and Maïeul Rouquette. \texttt{eledmac} A presumptuous attempt to port EDMAC,TABMAC and EDSTANZA to LaTeX. December 2004. (Available from CTAN in macros/latex/contrib/eledmac)

Index

Symbols

\@adv .......................................................... 1
\@astanza@line ................................................ 1
\@cs@linesinparL .............................................. 1
\@cs@linesinparR .............................................. 1
\@cs@linesonpageL ............................................. 1
\@cs@linesonpageR ............................................. 1
\@donereallinesL .............................................. 1
\@donereallinesR .............................................. 1
Command@pend ................................................. 47
Command@pendR ........................................ 47
Command@pstart .......................................... 48
Command@pstarttrue .................................. 93
Command@ref ............................................. 45–47, 50, 152
Command@ref@late ...................................... 47
Command@ref@reg@parsegAR .................................. 47
Command@ref@regR ...................................... 46
Command@set .............................................. 44, 152
Command@sw ................................................. 46
Command@AtBeginPairs ................................ 9, 56, 151, 155
Command@AtEveryPend ................................ 11, 151, 152
Command@AtEveryPend* ................................ 11
Command@AtEveryPstart ................................ 3, 11, 20, 64, 95, 151, 152
Command@AtEveryPstart* ................................ 11
Command@AtEveryPstartCall .............................. 3, 20, 64, 152
Command@AtEveryStanza .................................. 154
Command@AtEveryStopStanza .................................. 154
Command@Clear the right lines for `read@linelist' ........................................... 42
Command@Columns ........................................ 8, 9, 20, 54, 94, 102, 148–150, 152–154
Command@Columns@print@after@pend ................. 103
Command@Columns@print@before@pstart ......... 103
Command@Lcolwidth ..................................... 9, 12, 106, 130
Command@Lcolwidth ..................................... 155
Command@Leftpagehook ................................. 21, 157
Command@Leftsidehook .................................. 148
Command@Leftsidehookend .............................. 148
Command@MakePerPage .................................... 15
Command@Pages ............................................. 5, 8, 12, 14, 20, 27, 53, 54, 75, 79, 81, 105, 106, 111, 121, 125, 148, 150–154, 156
Command@Pages@mainmatter .................................. 122
Command@Pairs ............................................. 53
Command@Rcolwidth ..................................... 9, 12, 106, 130
Command@Rcolwidth ..................................... 155
Command@Rightpagehook ................................. 21, 157
Command@Rightsidehook .................................. 148
Command@Rightsidehookend .............................. 148
Command@Rlineflag ...................................... 128
Command@Xendlineflag .................................. 17, 154
Command@Xlineflag ..................................... 16, 129, 154
Command@Xmaxnotes .................................... 15
Command@Xnoteswidthliketwocolumns ..................... 10, 151
Command@Xonlyside ..................................... 15, 79, 129, 154, 157
Command@Xparindent ................................... 157
Command@Xpstart ........................................ 156
Command@Xtxtbeforenotes .................................. 154
Command@& ................................................ 21
Command@absline@numR ..................................... 41
Command@add@apparatusL .................................. 67
Command@add@apparatusR .................................. 69
Command@add@penalties ..................................., 78
Command@add@penaltiesL ..................................., 77
Command@advance ........................................ 44, 50, 152
Command@affixline@num ..................................... 72
Command@affixline@numR ..................................... 72, 148, 149
<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>@dlcolrawbox2</td>
</tr>
<tr>
<td>@drcolrawbox</td>
</tr>
<tr>
<td>@dcalc@maxoftwo</td>
</tr>
<tr>
<td>@dcalc@minoftwo</td>
</tr>
<tr>
<td>@dchecklang</td>
</tr>
<tr>
<td>@dcsnote</td>
</tr>
<tr>
<td>@dleleftbox</td>
</tr>
<tr>
<td>@dlinenumR</td>
</tr>
<tr>
<td>@dlsnote</td>
</tr>
<tr>
<td>@dmake@labels</td>
</tr>
<tr>
<td>@dmaxlinesinpar</td>
</tr>
<tr>
<td>@dmaxlinesinpar1</td>
</tr>
<tr>
<td>@dminpagelines</td>
</tr>
<tr>
<td>@dnumpstartsL</td>
</tr>
<tr>
<td>@dprintingcolumnstrue</td>
</tr>
<tr>
<td>@dprintingpagestrue</td>
</tr>
<tr>
<td>@dpscl</td>
</tr>
<tr>
<td>@dpscr</td>
</tr>
<tr>
<td>@dssnote</td>
</tr>
<tr>
<td>@dsetupmaxlinecounts</td>
</tr>
<tr>
<td>@dseulanguage</td>
</tr>
<tr>
<td>@d@maxlinecounts</td>
</tr>
<tr>
<td>@dprev@uopbR</td>
</tr>
<tr>
<td>@dprev@pbR</td>
</tr>
<tr>
<td>@labels@starttrue</td>
</tr>
<tr>
<td>@label@ref@list</td>
</tr>
<tr>
<td>@label@ref@listR</td>
</tr>
<tr>
<td>@lang</td>
</tr>
<tr>
<td>@last@page@numR</td>
</tr>
<tr>
<td>@led</td>
</tr>
<tr>
<td>@led@nopbR</td>
</tr>
<tr>
<td>@led@nopbnumR</td>
</tr>
<tr>
<td>@led@pbR</td>
</tr>
<tr>
<td>@led@pbnumR</td>
</tr>
<tr>
<td>@led@innernote</td>
</tr>
<tr>
<td>@led@leftnote</td>
</tr>
<tr>
<td>@lednopb</td>
</tr>
<tr>
<td>@lednopbR</td>
</tr>
<tr>
<td>@lednopbnumR</td>
</tr>
<tr>
<td>@ledouternote</td>
</tr>
<tr>
<td>@ledpb</td>
</tr>
<tr>
<td>@ledpbR</td>
</tr>
<tr>
<td>@ledpbnumR</td>
</tr>
<tr>
<td>@ledrightnote</td>
</tr>
<tr>
<td>@ledsidenote</td>
</tr>
<tr>
<td>@ledstrutL</td>
</tr>
<tr>
<td>@ledstrutR</td>
</tr>
<tr>
<td>@ledthegoal</td>
</tr>
<tr>
<td>@led@strut</td>
</tr>
<tr>
<td>@leftlinenumR</td>
</tr>
<tr>
<td>@leftlinenumnotationR</td>
</tr>
<tr>
<td>@lemma</td>
</tr>
<tr>
<td>@let</td>
</tr>
<tr>
<td>@line@list@R</td>
</tr>
</tbody>
</table>
Index

COMMAND\print@Xnotes@forpages .......................................................... 79, 80, 152
COMMAND\print@columnseparator .......................................................... 97, 150
COMMAND\print@eledsectionL ............................................................... 66
COMMAND\print@leftcolumn ................................................................. 101
COMMAND\print@line ........................................................................ 66
COMMAND\print@lineL ....................................................................... 66
COMMAND\print@notesX@forpages ........................................................ 152
COMMAND\print@rightcolumn ............................................................... 101
COMMAND\print@linenumR ................................................................. 129
COMMAND\printlinesR ................................................................. 139
COMMAND\psstart ................................................................. 4,
          8, 11, 12, 18, 20–22, 35, 44, 50, 54, 58, 59, 62, 64, 91, 92, 95, 103, 148, 150, 152, 153, 155–157
COMMAND\psstartL ................................................................. 64, 150
COMMAND\psstartR ................................................................. 64, 149, 150
COMMAND\psstartinfootnote .............................................................. 152
COMMAND\psstarts@typeset@L ............................................................... 95
COMMAND\raw@text ................................................................. 91
COMMAND\read@linelist ................................................................. 42, 132, 153
COMMAND\ref@regR ................................................................. 46, 152
COMMAND\ref@reg ................................................................. 46, 152
COMMAND\relax ................................................................. 129
COMMAND\reledmac ............................................................... 153
COMMAND\renewcommand .............................................................. 21, 128
COMMAND\resumenumbering ............................................................ 10, 11, 34, 35, 150, 154
COMMAND\resumenumberingR ............................................................ 151
COMMAND\rightlinenumR ............................................................... 40, 148
COMMAND\rightlinenumannotationR .................................................. 41
COMMAND\sameword ................................................................. 83
COMMAND\section ................................................................. 148
COMMAND\section@num ............................................................... 31
COMMAND\selectlanguage ............................................................ 20, 90, 91
COMMAND\set@line ................................................................. 50, 153
COMMAND\set@line@action .............................................................. 44, 153
COMMAND\set@lineflag ................................................................. 19, 128
COMMAND\setgoalfraction ............................................................... 14, 128
COMMAND\set@linefract ................................................................. 14
COMMAND\sethangingsymbol ............................................................. 21
COMMAND\setline ................................................................. 44, 50, 153
COMMAND\settlinenum ............................................................... 44, 50, 153
COMMAND\set@msdataposition ........................................................ 155
COMMAND\setnote@position ............................................................ 103
COMMAND\setparledgroupnotespacing ............................................... 128, 153
COMMAND\setposition ............................................................... 103
COMMAND\setspan ................................................................ 103
COMMAND\set@stanzaindents ............................................................ 10, 21, 128, 130
COMMAND\setwidethkietwocolumns .................................................. 103
COMMAND\sidenotemargin .............................................................. 22, 150
COMMAND\sidenotemargin* .............................................................. 22, 150
COMMAND\sidenotemarginR .............................................................. 22, 155
COMMAND\skipnumbering .............................................................. 18, 152
COMMAND\sloppy ................................................................. 9
COMMAND\stanza ................................................................. 8, 10, 18, 21, 88, 129, 130, 149
COMMAND\stanzanumtrue ............................................................... 22
COMMAND\startlock ................................................................. 50, 152
COMMAND\startswith ............................................................... 50, 152
COMMAND\sub@action ............................................................... 44, 153
Index

COMMAND\sub@off ................................................. 85
COMMAND\sub@on ................................................ 85
COMMAND\subline@numR ......................................... 41
COMMAND\sublinenumberstyle .................................. 18, 129
COMMAND\sublinenumberstyle' ................................. 129
COMMAND\sublinenumberstyleR ................................. 129
COMMAND\sublinemincrement ................................. 150, 153, 154
COMMAND\sublinenumrepR ........................................ 39, 148
COMMAND\syntaxonly .................................. 154
COMMAND\sza@0@ ........................................... 21
COMMAND\textheight ........................................... 15
COMMAND\textwidth .............................................. 56
COMMAND\the@labelX ........................................... 154
COMMAND\theledlanguageL ..................................... 90, 91
COMMAND\theledlanguageR ..................................... 90, 91
COMMAND\thepage ................................................ 23
COMMAND\thepstartL ........................................... 18, 149
COMMAND\thepstartR ........................................... 18, 149
COMMAND\thestanzaL ............................................... 22
COMMAND\thestanzaR ............................................... 22
COMMAND\this@section@last@page@numR .................... 144
COMMAND\vbox ........................................... 59
COMMAND\vskip .............................................. 124
COMMAND\vsplit ............................................... 77
COMMAND\widthliketwocolumns ............................... 10
COMMAND\widthliketwocolumnsfalse ......................... 10
COMMAND\widthliketwocolumnstrue ......................... 10
COMMAND\xflagref ............................................. 154
COMMAND\xright@appenditem ................................ 52
COMMAND\xspace ............................................... 24
COMMAND\xxxfootstart ......................................... 103
ENVIRONMENTLeftside ..................................... 11, 56, 101
ENVIRONMENTRightside ....................................... 11, 57, 101
ENVIRONMENTstanzar ........................................ 21, 88, 129, 130, 153, 156, 157
ENVIRONMENTcolumns .......................................... 33, 154
ENVIRONMENTledgroup ......................................... 7, 154
ENVIRONMENTleft .............................................. 18
ENVIRONMENTpages ........................................... 27, 33, 55, 130, 152, 154, 155
ENVIRONMENTpairs ........................................... 55, 152, 154
PackageEDMAC .............................................. 130
PackageEDSTANZA ........................................... 130
PackageEleledmac ........................................... 152
PackageEleledpar ............................................. 152
PackageTABMAC .............................................. 130
Package babel .............................................. 20, 89–91, 154, 157
Packagecalc ................................................ 157
Packageeleledmac .............................................. 5, 92, 127, 128, 130, 150, 152, 153
Packageeleledpar .............................................. 5, 6, 14, 38, 128, 129, 150, 151
Packageetoolbox ............................................. 31, 97, 123
Packageledmac ................................................ 6
Packageledpar .................................................. 2, 6
Packagememoir ............................................... 130
Packagemusixtex ........................................... 150
Packageperpage ............................................. 3, 15, 129, 154
Index
141

Package polyglossia ........................................ 20, 89–91, 154, 157
Package reledmac ........................................... 2, 4, 6–8, 10, 11, 14, 15, 18–24, 26, 27, 31, 33, 36, 38, 41, 42, 44–54, 66, 75, 84, 85, 87, 106, 111, 128, 129, 152–157
Package reledpar ........................................... 2, 4–8, 10–14, 21–26, 35, 41, 42, 47–49, 51, 53, 54, 84, 98, 128–130, 153
Package setspace ........................................... 3, 23
Package syntonly ............................................ 154
Package xkeyval ............................................ 24
Package xspace ............................................. 154

A
\abline@numR ................................................ 1
\actionlines@listR ........................................ 1
\actions@listR ............................................. 1
\add@apparatusL ........................................... 1
\add@apparatusR ........................................... 1
\add@inserts@nextR ....................................... 1
\add@insertsR ............................................. 1
\addr@penaltiesL .......................................... 1
\addr@penaltiesR .......................................... 1
\advanceR .................................................. 1
\affixline@numR .......................................... 1
\affixpstarts@numR ....................................... 1
\affixpstarts@numL ...................................... 1
\affixside@noteR ......................................... 1
\aftercolumnseparators .................................. 1
\annot@listR .............................................. 1
\araw@textfalse ......................................... 1
\araw@texttrue ........................................... 1
\astanza (environment) .................................... 21
\atBeginPairs ........................................... 1
\atEveryPstartCall ...................................... 1
\autopar ................................................... 1

B
\bbil@set@language ........................................ 1
\beforecolumnseparators .................................. 1
\beginnumbering .......................................... 17
\beginnumberingR ......................................... 1

C
\c@firstlinenumR .......................................... 1
\c@firstsublinenumR ..................................... 1
\c@linenumincrementR ................................... 1
\c@sublinenumincrementR ................................ 1
\chk@ck@checkR .......................................... 1
\checksub@ck@checkR ..................................... 1
\chapter .................................................. 1
\chapterinpages ......................................... 1
\check@goal .............................................. 1
\check@ppstart ......................................... 1
\checkpageL ............................................. 1
\checkpageR ............................................. 1
\checkpb@o@columns .................................... 1
\checkpb@L ............................................... 1
Index

\checkpbR ........................................ 1
\checkrauvatext ................................ 1
\checkverseL ................................ 1
\checkverseR ................................ 1
\clearleftpage ................................ 1
\clearrightpage ................................ 1
\cleartevenpage ................................ 1
\cleartoevenpage .............................. 1
\columnrulewidth .............................. 1, 9
\Columns ......................................... 1, 9
\columns@position ................................ 1
\Columns@print@after@pend ..................... 1
\Columns@print@before@pstart ................ 1
\columnseparator ............................... 1, 9
\columns@position ................................ 1
\correct@footins@box .......................... 1
\correct@Xfootins@box ......................... 1
\countline ....................................... 1
\countRline ..................................... 1
\countLline ..................................... 1
\critext ......................................... 1

D
\do@actions@fixedcodeR ................. 1, 10
\do@actions@nextR ....................... 1, 10
\do@actionsR ................................ 1
\do@ballastR ................................ 1
\do@insidelineLhook ...................... 1
\do@insidelineRhook ...................... 1
\do@lineL ....................................... 1
\do@lineLhook ................................ 1
\do@lineR ....................................... 1
\do@lineRhook ................................ 1
\do@lockoff ................................... 1
\do@lockoffR ................................... 1
\do@lockon ..................................... 1
\do@lockonR .................................... 1
\doinsideLhook ................................ 1
\doinsideRhook ................................ 1
\do@lineLhook ................................ 1
\do@lineRhook ................................ 1
\dump@pstartL@pc ............................ 1
\dump@pstartR@pc ............................ 1

E
\edlabel ........................................ 1
\edtext .......................................... 1
\edtext@later .................................. 1
\edtext@now .................................... 1
\edtext@later .................................. 1
\edtextnow ..................................... 1
\eled@sectioningR@out ..................... 1
\eled@section@correcting@skip ............ 1
\eledsectmark ................................ 1
\eledsectnotoc ................................ 1

D
Index

\pstartL .................................................. 1
\pstartR .................................................. 1

R
\Rcolwidth .............................................. 1, 9, 12
\read@linelist ........................................... 1
\reledpar@error .......................................... 1
\reledpar@warning ....................................... 1
\restore@notesX@settings ................................ 1
\restore@pstartL@pc ..................................... 1
\restore@pstartR@pc ..................................... 1
\restore@Xnotes@settings ............................... 1
\resumenumberingR ....................................... 1
\rightlinenumannotationR ............................... 1
\rightlinenumR .......................................... 1
\Rightpagehook .......................................... 1, 21
\rightpstartnumL ........................................ 1
\rightpstartnumR ........................................ 1
\Rightside (environment) ................................ 17
\Rightsidehook ........................................... 1
\Rightsidehookend ....................................... 1
\RLineflag ............................................... 1

S
\save@familiarfootnote@number .......................... 1
\save@section@number .................................... 1
\section@numR ............................................ 1
\selectlanguage ......................................... 1
\set@line ................................................ 1
\set@line@action ........................................ 1
\set@sectcountR .......................................... 1
\setgoal@fraction ....................................... 1
\sethangingsymbol ...................................... 21
\setline ................................................ 1
\setlinenum ............................................ 1
\setnamebox .............................................. 1
\setnote@position@liketwocolumns@C .................. 1
\setnote@position@liketwocolumns@L .................. 1
\setnote@position@liketwocolumns@R .................. 1
\setposition@liketwocolumns@C ......................... 1
\setposition@liketwocolumns@L ......................... 1
\setposition@liketwocolumns@R ......................... 1
\setRLineflag ........................................... 19
\setwidethlike@twocolumns ................................ 1
\sidetnote@margi@n ..................................... 1
\sidetnote@marginR ...................................... 1
\sidetnote@margin* ...................................... 1
\skip@lockoff ............................................ 1
\skip@numbering .......................................... 1
\startlock ............................................... 1
\startsub ............................................... 1
\sub@action ............................................. 1
\subline@numR ........................................... 1
\sublinenum@style* ..................................... 1
Change History

v0.1.0.
General: First public release ............................................. 1

v0.2.0.
\Columns: Added \checklang and \duselanguage to \Columns ........... 95
\Pages: Added \duselanguage to \Pages .................................. 107
General: Added section of babel related code .......................... 89
Fix babel problems ....................................................... 1

v0.3.0.
\Pages: Added \ledstrutL to \Pages ..................................... 107
\Rightsidehookend: Added \Lsidelook, \Lsidelookend, \Rightsidehook
and \Rightsidehookend .................................................. 57
\affixline@numR: Changed \affixline@numR to match new eledmac ....... 72
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR ...... 71
\do@lineL: Added \do@linehook to \do@lineL .......................... 65
Simplified \do@lineL by using macros for some common code .......... 65
\do@lineR: Changed \do@lineR similarly to \do@lineL .................. 68
\flag@end: Removed extraneous spaces from \flag@end ............... 50
\ifledRcol: Moved \ifl@dpairing to eledmac .......................... 27
\ifpst@rtedR: Moved \ifpst@rtedL to eledmac ....................... 110
\do@linenumR: Simplified \leftlinenumR and \rightlinenumR by introducing
\do@linenumR ........................................................... 40
\do@numpstartL: Moved \do@numpstartL to eledmac .................. 92
\ledstrutR: Added \ledstrutL and \ledstrutR ........................ 112
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR ........ 39
General: Added \do@lineLhook and \do@lineRhook ..................... 68
Added hooks into Leftside environment ................................ 56
Reorganize for ledarab ................................................ 1

v0.3.a.
\line@marginR: Do not just set \line@marginR in \linenummargin ....... 36
General: Minor \linenummargin fix ................................... 1

v0.3.b.
\Pages: Added \ominumpagelines calculation for succeeding page pairs .... 110
General: Improved parallel page balancing ............................ 1

v0.3.c.
General: Compatibility with Polyglossia ............................... 1

v0.4.0.
General: No more ledparpatch. All patches are now in the main file. .... 1

v0.5.0.
General: Corrections about \section and other titles in numbered sections .... 1

v0.6.0.
General: Be able to us \chapter in parallel pages .................... 1

v0.7.0.
General: Add shiftedverses option which makes there is no blank between two
parallel verses with inequal length .................................. 1

v0.8.0.
General: Possibility to have a symbol on each hanging of verses, like in the french
typography. Redefine the commande \hangingsymbol to define the character. .... 1

v0.9.0.
\ifledRcol: Moved \ifledRcol and \ifnumberingR to eledmac ............... 27
General: Possibility to number \psstart .................................. 18
Possibility to number the \psstart with the commands \numberpsstarttrue.
v0.9.1.
  General: The numbering of the pstarts restarts on each \beginnumbering. ......... 1
v0.9.2.
  General: Debug: with \Columns, the hanging indentation now runs on the left columns
  and the hanging symbol is shown only when \stanza is used. ..................... 1
v0.9.3.
  General: \thepstartL and \thepstartR use now \bfseries and not \bf, which is
  deprecated and makes conflicts with memoir class. ............................... 1
v0.10.0.
  General: \edlabel commands on the right side are now correctly indicated. .... 1
  \edlabel commands which start a paragraph are now put in the right place. .... 1
v0.11.0.
  \Columns: Line numbering by pstart. .................................................. 96
  \affixline@numR: Changed \affixline@numR to allow to disable line numbering
  (like in eledmac 0.15). ............................................................... 72
  \getnextboxR: Change \getnextboxL and \getnextboxR to allow to disable line
  numbering (like in eledmac 0.15). ............................................... 118
  Pstart number can be printed in side ............................................. 119
  \inserthangingsymbolR: Prevent the column separator for hanging verse from shifting
  General: Change \do@lineL and \do@lineR to allow line numbering by pstart (like in
  eledmac 0.15). .......................................................... 65
  Lineation can be by pstart (like in eledmac 0.15). ................................ 35
  New management of hangingsymbol insertion, preventing undesirable insertions. . 87
v0.12.0.
  General: New management of hangingsymbol insertion, preventing undesirable
  insertions. ................................................................................. 87
v1.0.0.
  General: Compatibility with eledmac. Change name to eledpar. ................. 1
  Debug in lineation by pstart ....................................................... 35
v1.0.1.
  General: Correction on \numberonlyfirstinline with lineation by pstart or by page. 1
v1.1.0.
  \pstartR: Add \labelpstarttrue (from eledmac). ..................................... 58
  General: The shiftedverses options becomes shiftedpstarts. ................... 1
v1.1.1.
  \pstartR: Correct \pstartR bug introduced by 1.1. .................................. 58
v1.1.2.
  \affixside@noteR: Remove spurious space between line number and line content . 86
v1.2.0.
  General: Support for \led⟨section⟩ commands in parallel texts. ............... 1
v1.2.1.
  \set@sectcountR: For the right section, the counter is defined only once. .... 33
v1.3.0.
  \edtext: Manage RTL language. ....................................................... 50
v1.3.2.
  General: Debug with some classes. .................................................. 1
v1.3.3.
  General: Debugging the left notes of the right column. .......................... 86
v1.3.4.
  General: Allow use of commands in sidenotes, as introduced by eledmac 1.0. .. 86
v1.4.0.
  General: Added \do@insidelineLhook and \do@insidelineRhook .................. 68
v1.4.1.
  General: Enable the use of stanzaindentsrepetition within astanza environment. 88
v1.4.3.
\\inserthangingsymbolR: Hanging verse is no longer automatically flush right. ........ 87
\pendL: Spurious spaces in \pendL .............................................. 62
\ pendR: Spurious spaces in \pstartR ............................................ 63
\pstartR: Spurious spaces in \pstartL and \pstartR. ........................... 58
General: Corrects a false hanging verse when a verse is exactly the length of a line. .... 1

v1.5.0.
\sublinenumincrement*: Add starred version of \firstlinenum.
\linenumincrement, \firstsublinenum, \sublinenumincrement to change
both Left and Rightside. ................................................................ 38
General: Add, as in eledmac, features to manage page breaks. ................................ 1

v1.6.0.
General: Add tool and documentation for parallel ledgroups ................................. 22
v1.7.0.
General: Add, as in eledmac, features to make crossrefs with pstart numbers. .......... 1
v1.8.0.
\Columns: Modify \Columns to enable to add section's title. ............................... 94
\ Suppress \l@cchecklang from \Columns. ............................................ 95
\Pages: Modify \Pages to enable to add section's title. ...................................... 105
\l@cchecklang: Suppress \l@cchecklang which did not work and was not logical,
because both columns could have the same language but not the main language of
the document. ..................................................................................... 90
\ pendL: As in eledmac, \pendL can have an optional argument. ................. 62
\ pendR: As in eledmac, \pendR can have an optional argument. ................. 63
\ print@columnseparator: Move some code of \Columns to
\ print@columnseparator. ..................................................................... 97
\pstartR: As in eledmac, \pendL and \pendR can have an optional argument. .... 58
\sidenotemarginR: \sidenotemargin is now directly defined in eledmac to be able
to manage eledpar. ............................................................................. 85
Add \sidenotemargin*. ......................................................................... 85
\theledlanguageR: Correct left/right language setting with polyglossia. .......... 91
General: \beginnumbering is defined only on eledmac, not on eledpar. .......... 31
\l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in eledmac. .... 86
Add \beforecolumnseparator and \aftercolumnseparator. ......................... 10
Add \columnsposition. ........................................................................ 10
Add, as in eledmac, new system of sectioning commands. 1
Add, as in eledmac, option to insert something after \pend / verses. ............... 1
Add, as in eledmac, option to insert something between \pstart / verse. ......... 1
Change \do@lineR and \do@lineL to allow new sectioning commands. ............ 65
Compatibility with musixtex ............................................................... 1
Debug eledmac sectioning command after using \resumenumbering. ................. 1
New sectioning commands, as in eledmac. ................................................. 23
Suppress \ifl@dsamelang which did not work and was not logical, because both
columns could have the same language but not the main language of the document. 90

v1.8.1.
\do@lineL: Fix a bug with critical notes a the beginning of a page, (maybe added by
v1.8.0) (?). ......................................................................................... 65
\do@lineR: Fix a bug with critical notes a the beginning of a page, added by v1.8.0 (?). 68
v1.8.2.
\flag@end: \flag@start and \flag@end are now defined only one time for eledmac
and eledpar ................................................................. 50
\lineation*: Add \lineation* ............................................................. 36
\eledparerror: Errors specific to eledpar send to eledpar handbook ................. 27
General: Debug \eledxxx with some paper sizes ......................................... 1
Change History

153

Debug left and side note (bugs added by 1.8.0) .......................... 1

v1.8.3.
\Pages: Debug blank pages when using optional argument in the last \pend. .... 105
\doinsidelineRhook: Added \dolineLhook, \dolineRhook, \doinsidelineLhook
and \doinsidelineRhook ........................................ 67
\pausenumbering\page\numR: Debug \resumenumberingR .................. 34
General: Add \noeledxxx, as in eledmac ................................ 1

v1.9.0.
\ifwidthliketwocolumns: Added widthliketwocolumns option ............ 26
\theledlanguageR: Debug left/right language switching with polyglossia. Do not
write in aux file when setting left/right lines. .......................... 91
General: Add \AtBeginPairs macro. .................................... 9
Compatibility with \Xnoteswidthliketwocolumns and
notesXwidthliketwocolumns .......................................... 1

v1.9.1.
\ifledRcol: Moved \ifl@dpaging to eledmac ............................ 27

v1.10.0.
\Pages: Debug wrong pages splitting when no optional argument is used in last \pend
(bug was added in v.1.8.3). ........................................ 105
Debug wrong parallel pages synchronization when an \edtext falls across two
pages. ................................................................. 105
General: Compatibility with \AtEveryPstart and \AtEveryPend ............ 1
Restore critical notes in \eledsection in parallel columns (this bug was added in
1.8.2). ................................................................. 1

v1.10.1.
\next@line@list@stuffR: Revert modification of 1.4.2, which makes bugs with
numbering. Leave vertical mode to solve spurious space before minipage. .... 49

v1.11.0.
\edtext: \critext and \edtext are now defined only in eledmac. .......... 50
General: Compatibility of standard footnotes with some biblatex styles. ... 1

v1.12.0.
\Columns: Add \@dprintingcolumnstrue ................................. 94
\Pages: Add \@dprintingpagemstrue .................................. 105
\edlabel: \edlabel and \edindex works now with hyperref when using eledpar . 84
\edlabel is now defined only one time for both eledmac and eledpar .... 84
\print@eledsectionL: Compatibility with Lua\TeX RTL languages. ........ 66
\print@eledsectionR: Compatibility with Lua\TeX RTL languages. ........ 69
\print@lineL: Compatibility with Lua\TeX RTL languages. ................ 66
General: Compatibility with Lua\TeX RTL languages. ..................... 1

v1.12.1.
\print@eledsectionL: Fix bug with Lua\TeX RTL \eledsection. .......... 66

v1.13.0.
\Pages: Prevent false overfull hboxes when using \Pages outside of pages
environment. ...................................................... 106
When using shiftedpstarts option, a \@dleftbox with a null height will advance the
\pagetotal in any case ............................................. 105
\clearl@drightpage: Use \newpage instead of \clearpage ................ 113
\ifledRcol: Remove false boolean settings which are not needed. ...... 27
General: Enable the use of optional argument of & in astanza environment . 88
Fix a bug in shiftedpstarts when size difference between pstarts is very important. ... 1
With parallel pages, long notes can now flow from the Left to the right side and from
the Right to the left side. .......................................... 1

v1.13.1.
\Pages: Prevent false empty page after \Pages (bug added in 1.13.0) .... 105
Change History

v1.16.0.
General: Fix a bug with line number position when using \eledsection and similar commands for RTL texts with LuaLaTeX.
The \nesifs are not followed by boolean values set to false, because it is the \TeX default setting.

v1.15.0.
\do@actions@nextR: Add actions 1008 and 1009 ......................... 71
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting when a verse is hanging ......................... 87
\lineationR: As \lineation, \lineationR automatically set the \pstartinfootnote. ........................................... 35
\n@num: \n@num defined only one time for both Eledmac and Eledpar. ........ 45
\skipnumbering: \skipnumbering defined only one time for both Eledmac and Eledpar ........................................... 50
General: Add \AtEveryPstartCall. ..................................... 1
Add sameparallelpagenumber option. .................................. 14
Fix vertical spurious space before right \eledchapter (bug added in v1.13.0). ... 1
Prevent vertical space when using \AtEveryPstart or \AtEveryPend with a command which prints nothing ..................................... 1

v1.16.0.
\newseries@par: Fix a bug with \onlysideX .................................. 51
General: Error message when calling \Pages inside pages environment and \Columns inside pairs environment. ......................... 1
Error message when starting a Leftside/a Rightside while the previous one has not been yet typeset. ......................... 1
Error message when using \beginnumbering...\endnumbering without \pstart ... 1
Fix a bug with nofamiliar / nocritical option of eledmac. .................. 1
New package option sameparallelpagenumber to have the same page number for both left and right side. ......................... 1

v1.16.1.
General: Write information about line-list file version in the correct file. ......................... 1

v1.16.2.
General: Fix a bug when adding empty lines before a \pend in combination with some specific penalties setting. ......................... 1

v1.17.0.
General: Add compatibility of optional argument of \pstart/\pend and \AtEveryPstart/\AtEveryPend with two columns mode. ............... 1

v1.21.0.
General: Add \hidenumbering ................................. 18

v2.0.0.
\@adv: \@adv defined only in reledmac. ......................... 44
\@lab: \@lab defined only in eledmac. ........................... 85
\@ref@regR: \@ref defined only in reledmac, code specific to right side moved in \ref@regR. ........................... 45
\@set: \@set defined only in reledmac. ........................... 44
\advanceline: \advanceline defined only in reledmac. .................. 50
\@bl@set@language: Patch \@bl@set@language instead of redefining it ........... 90
\do@lockonR: \do@lockon defined only in reledmac. ................... 44
\endlock: \@startlock and \endlock defined only in reledmac. .............. 50
\endsub: \@startsub and \endsub defined only in reledmac. .............. 50
\textbackslash fixpage: \textbackslash fixpage is defined only once in \texttt{reledmac} ........................... 44
\textbackslash l\texttt{dd}set: \textbackslash l\texttt{dd}set defined only in \texttt{reledmac} ........................... 44
\textbackslash line@margin: \textbackslash line@margin now defined only once time in \texttt{reledmac} ........ 36
\textbackslash page@action: \textbackslash page@action defined only in \texttt{reledmac} ........................... 44
\textbackslash read@linelist: \textbackslash read@linelist is defined only once time in \texttt{reledmac} ........ 42
\textbackslash set@line: \textbackslash set@line defined only in \texttt{reledmac} ................................... 50
\textbackslash set@line@action: \textbackslash set@line@action defined only in \texttt{reledmac} ................. 44
\textbackslash setlinenum: \textbackslash setlinenum defined only in \texttt{reledmac} ........................... 50
\textbackslash skip@lockoff: \textbackslash do@lockoff defined only in \texttt{reledmac} .......................... 45
\textbackslash sub@action: \textbackslash sub@action defined only in \texttt{reledmac} ........................... 44
\textbackslash sublinenumincrement*: \textbackslash firstlinenum, \textbackslash linenumincrement, \\
\textbackslash firstsublinenum, \sublinenumincrement are now defined only in \texttt{reledmac}. 38
General: \textbackslash \texttt{\textbackslash qml} is now defined only in \texttt{reledmac} ........................... 43
\textbackslash ifbypage@ and \textbackslash ifbypstart@ defined in \texttt{eledmac} ............................... 35
\hspace{1.0em} Fix some bugs with \texttt{sameparallelpagenumber} option. ................................. 1
\hspace{1.0em} Many code refactored and moved to \texttt{reledmac}. ........................................... 1
\hspace{1.0em} Package's name becomes \texttt{reledpar} .................................................................... 1
\hspace{1.0em} Totally new implementation of \texttt{sameparallelpagenumber} option. ........................ 1
\hspace{1.0em} chapterinpages: Deleting the old system of managing parallel chapter, keep only the new one with \texttt{patchcmd}. ................................................................. 56

v2.1.0.

General: Fix a bug when using \texttt{eledsection} and related on right pages when page width is short. ................................................................. 1
\hspace{1.0em} Fix a bug when using \texttt{pagemumbering} with memoir (bug added in v2.0.0) ........ 1
\hspace{1.0em} Fix a bug with \texttt{setparledgroupnotespacing} with the shiftedpstarts option. .... 1
\hspace{1.0em} Fix incompatibility between optional argument of \texttt{pstart} and \texttt{numberpstarttrue} 1
\hspace{1.0em} Options to custom empty right page before \texttt{Pages}. ........................................... 1

v2.2.0.

General: \texttt{astanza} environment can take an optional argument, which will be the optional argument of \texttt{pstart} started by this environment. .............................. 1
\hspace{1.0em} New tools to number stanza ................................................. 1

v2.2.1.

General: Fix a bug with optional argument of last left \texttt{pend} ................. 1

v2.3.0.

\texttt{Pages}: Fix a bug when calling \texttt{Columns after a \texttt{Pages}} (bug added in v1.13.0) ..... 106
General: Change some internal codes in order to provide compatibility with \texttt{HypX} release of october 2015 ......................................................... 1
\hspace{1.0em} Fix a bug with title number in parallel columns ........................................... 1
\hspace{1.0em} New line setting command suffixed by R to set only the right side. ................. 1

v2.4.0.

\texttt{ledstrutR}: Deleted \texttt{ledstrutL} and \texttt{ledstrutR} ............................................. 112
\hspace{1.0em} Fix a bug with dotted letter ............................................................................ 112
\hspace{1.0em} General: New way of (not) synchronizing the parallel pages. .......................... 1
\hspace{1.0em} Option to switch to \texttt{mainmatter} when calling \texttt{Pages} ................................. 1

v2.5.0.

General: Disable empty lines as paragraph in \texttt{astanza}. ................................. 1
\hspace{1.0em} Fix a bug introduced in v1.15.0 which made hanging indentation in verse not work anymore ................................................................. 1
\hspace{1.0em} New commands \texttt{linenummarginR} and \texttt{linenummargin*} ...................... 1

v2.5.1.

General: Fix spurious space when using optional argument of \texttt{astanza} environment (introduced in v2.5.0). ........................................... 1
v2.5.2.
General: Fix a bug introduced in v2.5.0 with \linenummargin, \firstlinenum, \linenumincrement, \firstsublinenum, \sublinenumincrement. ........... 1

v2.6.0.
\@dmake@labelsR: \@rlineflag is not stored directly after the line number, but as a fifth argument of \the@labelX. Can be retrieved by \xflagref. ................. 84
General: \Xlineflag and \Xendlineflag added ......................................................... 1
Error message when calling \Pages or \Columns without previous pages or pairs environment. ................................................................. 1
Fix a bug with footnote numbering when using the same series of familiar footnotes on both sides. ................................................................. 1
Fix a bug with right side title number when using title commands before pages or columns environments. ......................................................... 1
Fix compatibility with babel (broken in v2.0.0). ......................................................... 1
No error messages about ends of left / right page when using the \syntxonly command of the syntonly package. ......................................................... 1

v2.6.1.
General: Fix a bug, introduced in v2.6.0, with footnote numbering when using perpage package. ......................................................... 1

v2.6.2.
\newseries@par: The \TeX counter \footnoteX@reading is defined in reledmac. .......... 51
General: Fix (again) bugs with footnote numbering in parallel typesetting while using ledgroup environments (bug added in v2.6.0). ......................................................... 1
Fix a bug (added in v2.6.0) with footnote numbering in parallel typesetting while using polyglossia with specific numbering systems (like Greek). ......................................................... 1

v2.6.3.
General: Fix spurious dot when using \linenummargin on right side (introduced in v2.5.0). ......................................................... 1

v2.7.0.
General: reledmac cross-referencing can take advantage of \xr package. ................. 1

v2.7.1.
General: Fix a bug added in reledmac 2.8.2, when typesetting parallel text just after a sectioning command ......................................................... 1

v2.8.0.
General: Allow continuing line numbering between normal text and parallel text, using \pausenumbering and \resumenumbering and the continuousnumberingwithcolumns options. ......................................................... 1
Add \linenumberLevenifblanktrue and \linenumberRevenifblank commands 1
Fix a bug when the right line number style is not the same to the left line number style 1

v2.9.0.
General: Add \AtEveryStanza and \AtEveryStopStanza. ......................................................... 1
More specific error messages. ......................................................... 1

v2.9.1.
General: Prevent \xtxtbeforenotes hook from causing notes to go beyond the bottom margin ......................................................... 1

v2.10.0.
\do@actions@nextR: Add action 1010 ......................................................... 71
General: Add new tools to make apparatuses of manuscripts ......................................................... 1

v2.11.0.
\correctfootinsX@box: Clarification in the handbook about the use of \xonlyside and onlysideX and error message if mesuse. ......................................................... 79
New implementation of \xonlyside and onlysideX hooks, prevent trouble with vertical spacing. ......................................................... 79
<table>
<thead>
<tr>
<th>Version</th>
<th>Change Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>v2.12.0</td>
<td>General: Compatibility with <code>reledmac</code>'s <code>auxdir</code> option.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with <code>\AtBeginPairs</code>.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with <code>\lineationR</code>.</td>
</tr>
<tr>
<td>v2.13.0</td>
<td>General: Compatibility with the new features of <code>reledmac</code> 2.15.0.</td>
</tr>
<tr>
<td>v2.13.1</td>
<td>General: Simplification of the code concerning the sectioning command.</td>
</tr>
<tr>
<td>v2.14.0</td>
<td>General: Add <code>\edtextlater</code> and <code>\edtextnow</code>.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with <code>\AtBeginPairs</code>.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with <code>\lineationR</code>.</td>
</tr>
<tr>
<td>v2.14.1</td>
<td>General: Fix a bug when using <code>\markboth</code> on <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code>.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with vertical space before sectioning command in optional argument of <code>\psstart</code> (bug added in v2.13.1).</td>
</tr>
<tr>
<td>v2.14.2</td>
<td>General: Fix spurious space with <code>\edtextlater</code>.</td>
</tr>
<tr>
<td>v2.14.3</td>
<td>General: Fix a bug in <code>\edtextnow</code>.</td>
</tr>
<tr>
<td>v2.14.4</td>
<td>General: Take into account <code>\linenumberlistR</code>, equivalent to <code>\linenumberlist</code> for the right side.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with <code>\lineationR</code> when left / right side are not defined.</td>
</tr>
<tr>
<td></td>
<td>More explicit error message when the stanza indentation is not defined.</td>
</tr>
<tr>
<td></td>
<td>New commands to have specific line margins for text running in parallel columns</td>
</tr>
<tr>
<td></td>
<td>When indexing texts in sidenotes with <code>\edtext</code>, referring to the line number where the sidenote is called.</td>
</tr>
<tr>
<td>v2.15.0</td>
<td>General: Add <code>\movecolumnspositiononrightpage</code> option.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug about space before sectioning commands after a pairs environment (bug added in v. 2.14.4).</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with <code>\msdata</code> when using multiple <code>\beginnumbering</code>...<code>\endnumbering</code>.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with page break when using <code>\columns</code> (bug added in v. 2.14.1).</td>
</tr>
<tr>
<td></td>
<td>New macro <code>\sidenotemarginR</code>, fix default value to right.</td>
</tr>
<tr>
<td>v2.16.0</td>
<td>General: Compatibility with <code>\setmsdataposition</code>.</td>
</tr>
<tr>
<td>v2.16.1</td>
<td>General: Restore sidenotes on left side, deleted by mistake in v2.16.0</td>
</tr>
<tr>
<td>v2.16.2</td>
<td>General: Change log message when numbered files still don’t exist, in order to improve compatibility with <code>latexmk</code>.</td>
</tr>
<tr>
<td>v2.16.3</td>
<td>General: Fix a bug when setting <code>\aftercolumnseparator</code>.</td>
</tr>
<tr>
<td>v2.17.0</td>
<td>General: Change code for compatibility with <code>reledmac</code> 2.18.0.</td>
</tr>
<tr>
<td>v2.17.1</td>
<td>General: Fix a bug with <code>\prevpgstyle</code> option when not using <code>\prevpgnotnumbered</code>.</td>
</tr>
<tr>
<td>v2.17.2</td>
<td>\Pages: Fix a bug when changing <code>\Lcolwidth</code> and <code>\Rcolwidth</code> in pages environment.</td>
</tr>
<tr>
<td>v2.17.3</td>
<td>General: Fix a bug when using multiple sidenotes on the same line, on right side.</td>
</tr>
<tr>
<td>v2.17.4</td>
<td>General: Fix bugs with sublines in parallel typesetting.</td>
</tr>
</tbody>
</table>
### Change History

<table>
<thead>
<tr>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>v2.18.0</td>
<td>General: Compatibility with v2.20.0 of reledmac.</td>
</tr>
<tr>
<td>v2.19.0</td>
<td>General: Compatibility with v2.22.0 of reledmac.</td>
</tr>
<tr>
<td>v2.19.1</td>
<td>General: Fix a bug with \edtext in sectioning command with shiftedpstarts mode.</td>
</tr>
<tr>
<td>v2.19.2</td>
<td>General: continuousnumberingwithcolumns also works for the pstart number.</td>
</tr>
<tr>
<td>v2.19.3</td>
<td>General: Fix a bug with \edtext in sectioning command with shiftedpstarts mode.</td>
</tr>
<tr>
<td>v2.19.4</td>
<td>General: Do not print footnotes at the first run.</td>
</tr>
<tr>
<td>v2.19.5</td>
<td>General: Optimisation of the code in order to not print footnotes at the first run.</td>
</tr>
<tr>
<td>v2.20.0</td>
<td>General: Add \linenumOnlyPagesForColumns and \linenumOnlyPagesForColumnsR to tell to print the line numbers associated with a column only on some specific pages (for example, only on right pages).</td>
</tr>
<tr>
<td></td>
<td>Add sidenotesmarginpage option to make the sidenotes refer to page margins and not column margins.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with \linenumOnlyPagesForColumns and \linenumOnlyPagesForColumnsR internal hook.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with \linenumOnlyPagesForColumns and inner / outer notes and line numbering.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with \linenumOnlyPagesForColumns when a \beginnumbering is called after another \beginnumbering...\endnumbering.</td>
</tr>
<tr>
<td></td>
<td>Fix a bug with \linenumOnlyPagesForColumns when a \beginnumbering is called after another \beginnumbering...\endnumbering.</td>
</tr>
<tr>
<td></td>
<td>Fix incompatibility between \linenumOnlyPagesForColumns and \linenumOnlyPagesForColumnsR option.</td>
</tr>
<tr>
<td>v2.20.1</td>
<td>General: Fix bug with \linenumOnlyPagesForColumns when left column and right column have not the same size.</td>
</tr>
<tr>
<td>v2.20.2</td>
<td>General: Fix bugs with \linenumOnlyPagesForColumns when going to a new page.</td>
</tr>
<tr>
<td>v2.21.0</td>
<td>General: Add compatibility with new features of reledmac.</td>
</tr>
<tr>
<td>v2.21.1</td>
<td>General: Fix bug with \linenumberannotation.</td>
</tr>
<tr>
<td>v2.21.2</td>
<td>General: Fix bug with \Xpstart and \edtextlater.</td>
</tr>
<tr>
<td>v2.21.3</td>
<td>General: Fix bug when using \numberpstarttrue and alternating between manual \pstart and automatic \pstart produced by \autopar.</td>
</tr>
<tr>
<td>v2.21.4</td>
<td>General: Fix bug with using \ledsection and related when alternate normal typesetting and parallel typesetting.</td>
</tr>
<tr>
<td>v2.21.5</td>
<td>General: Compatibility with reledmac v. 2.26.6.</td>
</tr>
</tbody>
</table>
v2.21.6.
General: Fix page number in footnote and endnotes when using
sameparallelpagenumber option ....................................... 1

v2.21.7.
General: Compatibility with reledmac 2.26.7 ............................. 1

v2.21.8.
General: Fix bug with critical footnotes line number when alternating between
\numberlinefalse and \numberlinetrue ....................................... 1
Fix bug with marginal line number in parallel typesetting when alternating between
\numberlinefalse and \numberlinetrue ....................................... 1
Workaround with a bug of \texttt{B\LaTeX}2018/04/01 patch level 3 to generate the .sty file from
a .ins file ........................................................................ 1

v2.21.9.
General: Fix (little) bug when using calc package ............................ 1

v2.22.0.
General: Compatibility with \texttt{doinsidesthistlinehook} macro .................. 1

v2.22.1.
General: Really reset line number annotation at each line .................... 1

v2.22.2.
General: Fix bug with \texttt{\onlyside} and \texttt{onlysideX} ......................... 1

v2.22.3.
General: When using \texttt{\parindentX} or \texttt{Xparindent} and stanza, use the normal
indentation in the footnote, not the verse indentation ......................... 1

v2.22.4.
\texttt{\theledlanguageR}: Provide compatibility with the latest version of polyglossia (v1.45). 91

v2.23.0.
\texttt{\theledlanguageR}: Restore compatibility with versions of polyglossia prior to v1.45 . 91
General: Add \texttt{Leftpagehook} and \texttt{Rightpagehook}. ...................... 1
Add hanging symbol feature to \texttt{astanza}. ................................ 1

v2.23.1.
General: Fix bug when using polyglossia v1.45 or later ....................... 1

v2.24.0.
General: Add antilabe feature .................................................. 1
Add startstanzaindentsrepetition feature .................................... 1

v2.24.1.
General: Fix incompatibility between lineation by \texttt{\pstart} and (a)stanza ................ 1
Makes \texttt{\numberstanzatrue} working with \texttt{astanza} .................... 1

v2.24.2.
General: Fix some bugs with lineation by page, when a numbered section starts at the
very beginning of page .................................................................. 1

v2.24.3.
General: Fix bug with hanging verse in parallel typesetting ..................... 1

v2.25.0.
General: Add compatibility with \texttt{\linenumannotationothersidetrue} of reledmac. 1

v2.25.1.
General: \texttt{\ifsu:blines@R} migrated to reledmac .............................. 1

v2.25.2.
General: Fix incompatibility with babel v3.61 ................................... 1

v2.25.3.
General: Fix incompatibility with polyglossia v1.55 ............................. 1