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 .................................. 7

2 Options 7
   2.1 Synchronization's options .......................... 7
   2.2 Other options ..................................... 8

3 General 8

*This file (reledpar.dtx) has version number v2.25.0, last revised 2020/11/29.
†maieul at maieul dot net
‡herries dot press at earthlink dot net
## Contents

4 Parallel columns 9

4.1 Basic use 9
4.2 Setting 10
   4.2.1 Column’s width 10
   4.2.2 Column’s separator 10
   4.2.3 Column’s positions 10
   4.2.4 Mixing two columns and one column texts 11
   4.2.5 Schemas of column typesetting 11
   4.2.6 Continuous numbering between single and double column texts 11
4.3 Note about \AtEveryPstart\* and \AtEveryPend\* 12

5 Keeping translation in the outside/inside column 12

6 Facing pages 12

6.1 Basic usage 12
6.2 Setting 13
   6.2.1 Text width 13
   6.2.2 Way of synchronizing 13
   6.2.3 Page number 15
   6.2.4 Page breaking 15
   6.2.5 Right page before \Pages\ 15
   6.2.6 Notes about \mainmatter\ 15
6.3 Critical and familiar footnotes 15
   6.3.1 Notes height setting 16
   6.3.2 About the numbering of familiar footnotes 16
   6.3.3 Using perpage package 16
   6.3.4 Notes for one side only 17
   6.3.5 Familiar notes called on the right side, but to be printed on the left side 17
6.4 Critical notes called on left side but printed on right side 17
6.5 Using line flag 18

7 Left and right texts 18

7.1 Environments 18
7.2 Numbering text lines and paragraphs 18
7.3 First line number and line number increment 19
7.4 Lineation system 20
7.5 Line number style 20
7.6 Line number margin 20
7.7 Line flag 21
7.8 Chunks 21
7.9 \AtEveryPstart\ and \AtEveryPstartCall 22
7.10 Language setting 22
7.11 Executing code at each line 22
7.12 Executing code at each page 22
Contents

8 Verse
9 Side notes
10 Parallel ledgroups
   10.1 General
   10.2 Parallel ledgroups and setspace package
11 Sectioning commands
12 Notes about page number
I Implementation overview
II Preliminaries
   II.1 Package's meta-data
   II.2 Package's requirement
   II.3 Package's options
   II.4 Package's options
      II.4.1 Synchronization's options
      II.4.2 Position options
      II.4.3 Other options
   II.5 Determining side and category of parallel processing
   II.6 Text's width
   II.7 Messages
   II.8 Naming macros
III Sectioning commands
IV Line counting
   IV.1 Setting lineation reset
   IV.2 Setting line number margin
   IV.3 Setting lineation start and step
   IV.4 Setting line flag
   IV.5 Setting line number style
   IV.6 Print marginal line number
   IV.7 Line-number counters and lists
      IV.7.1 Correspond to those in reledmac for regular or left text
      IV.7.2 Specific to reledpar
   IV.8 Reading the line-list file
   IV.9 Commands within the line-list file
   IV.10 Writing to the line-list file
### Contents

- **V Marking text for notes**
  - V.1 Specific hooks and commands for notes ........................................ 55
    - V.1.1 Notes to be printed on one side only ..................................... 55
    - V.1.2 Tools specific to familiar footnotes ...................................... 55
    - V.1.3 Get correct footnote number ................................................. 57
  - V.2 Create hooks ................................................................................. 58
  - V.3 Init standards series (A,B,C,D,E,Z) .............................................. 58
  - V.4 Tools specific to \LaTeX’s classical footnotes ................................ 58

- **VI Pstart numbers dumping and restoration** ........................................ 58

- **VII Parallel environments** ...................................................................... 59

- **VIII Paragraph decomposition and reassembly** ........................................ 62
  - VIII.1 Boxes, counters, \pstart and \pend ............................................. 62
  - VIII.2 Processing one line ..................................................................... 69
  - VIII.3 Line and page number computation .......................................... 75
  - VIII.4 Line number printing .................................................................. 78
  - VIII.5 Pstart number printing in side .................................................. 81
  - VIII.6 Add insertions to the vertical list .............................................. 83
  - VIII.7 Penalties ..................................................................................... 84
  - VIII.8 Printing leftover notes ................................................................. 85

- **IX Footnotes** .......................................................................................... 85
  - IX.1 Footnotes output specific to \Pages ............................................. 85
  - IX.2 Critical footnote printed in right side .......................................... 88

- **X Cross referencing** ................................................................................ 91

- **XI Sidenotes** .......................................................................................... 91

- **XII Verse** .................................................................................................. 94

- **XIII Fixing babel and polyglossia** ........................................................... 96

- **XIV Counts and boxes for parallel texts** .................................................. 99

- **XV Checking text to be processed** ........................................................... 100

- **XVI Parallel columns** ............................................................................... 102

- **XVII Parallel pages** .................................................................................. 113
  - XVII.1 Specific counters ......................................................................... 113
  - XVII.2 Main macro .................................................................................. 113
  - XVII.3 Ensure all notes are printed at the end of parallel pages .............. 120
  - XVII.4 Struts ........................................................................................... 121
  - XVII.5 Page clearing ................................................................................. 121
  - XVII.6 Lines managing ........................................................................... 122
1 Introduction

1.1 Aim of this package

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

The package has to try and coerce \LaTeX{} into paths it was not designed for. Use of the package, therefore, may produce some surprising results. In this case, please report them to the author via GitHub’s issues: https://github.com/maieul/ledmac/issues/

This manual contains a general description of how to use \texttt{reledpar} starting in section 3, the complete source code for the package, with extensive documentation (in sections I through XXIII), and an index to the source code. As \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 \texttt{nomaxlines} 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.13 to understand better those options.

\texttt{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.

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

\texttt{nomaxlines} allows facing pages to have different numbers of lines.

\texttt{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{ledgroup} environment with \texttt{reledpar}[1].

\texttt{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{reledpar}. This is useful when alternating between normal and parallel typesetting.[2]

\texttt{continuousnumberingwithcolumns} allow you to alternate between normal typesetting and parallel column typesetting without resetting the line number (see 4.2.6 p.[3]).

\texttt{sidenotesmarginpage} makes the sidenotes be placed on the margin of the page and not on the margin of the columns.

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

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

\texttt{movecolumnpositiononrightpage} 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 another one on the outer column.

3 General

A file may mix \textit{numbered} and \textit{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 \texttt{reledmac’s} note commands with it: this is appropriate for introductions and other material added by the editor around the edited text.

The \texttt{reledpar} package lets you typeset two \textit{numbered} texts in parallel[4]. 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.

\texttt{reledmac} essentially puts each chunk of numbered text (the text within a \texttt{\pstart} \ldots \texttt{\pend}) into a box and then following the \texttt{\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

---

[1] This option can either be used on \texttt{reledmac} or \texttt{reledpar}.
[2] This option can either be used on \texttt{reledmac} or \texttt{reledpar}.
[3] You can use, anyway, \texttt{\numberlinefalse} to disable printing of line numbers.
line are recalled. The line, with any notes, is then output for printing, possibly with a line number attached. Effectively, all the text is typeset and then afterwards all the notes are typeset.

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

\maxchunks{\langle\textit{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 \reledpar error message along the lines: “Too many \pstart without printing. Some text will be lost.” then you will have to either increase \maxchunks or use the parallel printing commands (\Columns or \Pages) more frequently.

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

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

\section{Parallel columns}

\subsection{Basic use}

\begin{pairs}
\begin{Leftside}\reledmac\numbered\text\commands\end{Leftside}
\begin{Rightside}\reledmac\numbered\text\commands\end{Rightside}
\end{pairs}
\Columns
\begin{pairs}
\begin{Leftside}\reledmac\numbered\text\commands\end{Leftside}
...\end{pairs}
\Columns
\AtBeginPairs Keep in mind that the \Columns must be outside of the pairs environment. You can use the macro \AtBeginPairs to insert a code at the beginning of each pairs environments. That could be useful to add the \sloppy macro to prevent overfull hboxes in two columns.
\AtBeginPairs{\sloppy}

There is no required pagebreak before or after the columns.

4.2 Setting

4.2.1 Column's width
\Lcolwidth \Rcolwidth

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

4.2.2 Column's separator
\columnrulewidth \columnseparator

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

4.2.3 Column's positions
\columnsposition

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.
4.2 Setting

\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 \widthliketwocolumns{true}. To reset this feature, use \widthliketwocolumns{false}. You can also use widthliketwocolumns as a global option when loading reledmac or reledpar.

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. 21).

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

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

\begin{verbatim}
\AtEveryPstart*{%
  \if Economicalcolumns
    \vspace{0.125\baselineskip}
  \else
    \vspace{0.25\baselineskip}
  \fi
}
\end{verbatim}

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

\begin{verbatim}
\begin{pages}
\begin{Leftside} reledmac numbered text commands \end{Leftside}
\begin{Rightside} reledmac numbered text commands \end{Rightside}
\end{pages}
\end{verbatim}
6.2 Setting

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

\Pages

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

6.2 Setting

6.2.1 Text width

\Lcolwidth \Rcolwidth

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 \LaTeX to be run several times. Therefore, it is advised to use in conjunction utilities such as latexmk to ensure that synchronization is complete.

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

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

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

The aim of the default setting is twofold:

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

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

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

\footnote{There is a French version of this article on \url{http://geekographie.maieul.net/185}}
• 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 \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 \texttt{12 p. 25}.

6.2.4 Page breaking

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

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

6.2.5 Right page before \Pages

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

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

\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 \texttt{12 p. 25}.

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

6.2.6 Notes about \mainmatter

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

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

6.3 Critical and familiar footnotes

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

Since \texttt{e\ledpar} v1.13.0, long notes in facing pages can flow from left to right pages, and \textit{vice-versa}.

However, the \texttt{reledmac} default setting for the maximum allotted size to notes is greater than \texttt{textheight}. That makes impossible for long notes to flow across pages.\footnote{The same applies to \LaTeX{} normal notes. Read \url{http://tex.stackexchange.com/a/228283/7712} for technical informations.}

We have not changed this default setting, because we do not want to break compatibility with older version of \texttt{reledmac} and we want to be as close as possible to default \LaTeX{}'s feature.

So, you MUST change the default setting via \texttt{\Xmaxhnotes} (for critical notes) and \texttt{\maxhnotesX} (for familiar notes). Both commands are explained in \texttt{reledmac} handbook \texttt{(7.12.6 p. 56)}. As an advisable setting:

\begin{verbatim}
\AtBeginDocument{%
  \Xmaxhnotes{0.6\texttheight}
  \maxhnotesX{0.6\texttheight}
}
\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 of run of typesetting the footnote numbering will not run down the columns. Instead, it will read both column lines completely across the page, and number footnotes from left to right.

6.3.3 Using \texttt{perpage} package

It follows from what has been said in the preceding paragraph that if you use the \texttt{\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}
\end{verbatim}
6.4 Critical notes called on leftside but printed on right side

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

6.3.4 Notes for one side only
\Xonlyside \onlysideX

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

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

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

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

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

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

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

6.4 Critical notes called on leftside but printed on right side
\edtextlater \edtextnow

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

The relationship between the \edtextlater and \edtextnow commands is determined by the order of calling: the first \edtextnow corresponds to the first \edtextlater, the second \edtextnow corresponds to the second \edtextlater etc. \edtextnow is a parameterless macro, so it gobbles the following space. If you want to keep it, add {} or backslash followed by a space.

6.5 Using line flag

\Xlineflag \Xendlineflag

Use \Xlineflag[⟨s⟩] to add right line flag (7.7 p. 21) 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}
7.3 First line number and line number increment

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

The lineation commands which finish by a R apply for right text. The lineation commands which are starred apply for both left and right texts. The lineation command which does not finish by a R and who are not starred apply for the left side. However, these commands apply to right side when they are called inside a left environment. However, such features should not be used any more. The recommended
practice is to add all setting commands to the preamble. 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 the right side. \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. 22] 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 \linenumberstyle* is the equivalent of reledmac \linenumberstyle for right text. \linenumberstyleR \sublinenumberstyleR \linenumberstyle* \sublinenumberstyle* is the equivalent of reledmac \sublinenumberstyle right text. The starred version are for both side.

7.6 Line number margin

\linenummarginR{⟨margin⟩} sets the line margin for right side. \linenummarginR*{⟨margin⟩} sets for both side. ⟨margin⟩ 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 your 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.
7.7 Line flag

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 \linenumberLevenifblanktrue to enable this on the left side and \linenumberRevenifblanktrue 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{⟨flag⟩}. 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 \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
Numbering, via `\begin{numbering}` and `\end{numbering}`, 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 `reledmac`'s handbook (5.2.2 p. 19).

7.9 \AtEveryPstart and \AtEveryPstartCall

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

Consequently:

- The argument of \AtEveryPstart (see 5.2.4 p. 20) 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` and `\dolineRhook` are the equivalent to the `reledmac` `\dolinehook`, respectively for left and right side.

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

About these two hooks, read `reledmac`'s handbook (5.6 p. 28).

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.

\astanza

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

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

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

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

\sethangingsymbol

Like in reledmac, you could use the \sethangingsymbol command to insert a character in each hanging line. If you use it, you must run \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 \setcounter{stanzaln}{0} 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: \sidenote, \leftnote, \rightnote, \outernote, \innernote.
The \sidenotemargin defines the margin of the sidenote for either left or right side, depending on the current environment.

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

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

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

10 Parallel ledgroups

10.1 General

You can also make parallel ledgroups (see the documentation of reledmac about ledgroups, [10] p. 63). 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}
\bignumbering
\pstart
\begin{ledgroup}
  ledgroup content
\end{ledgroup}
\pend
\pstart
\begin{ledgroup}
  ledgroup content
\end{ledgroup}
\pend
\endnumbering
\end{Leftside}
\begin{Rightside}
\bignumbering
\pstart
\begin{ledgroup}
  ledgroup content
\end{ledgroup}
\pend
\pstart
\begin{ledgroup}
  ledgroup content
\end{ledgroup}
\pend
\endnumbering
\end{Rightside}
\end{pages}
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.

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).

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 `sameparapagenumber` option (6.2.3 p. 15) or `prevpgnotnumbered` option (6.2.5 p. 15), please read the following paragraph if you want to manipulate page numbers manually.

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

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

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

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

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

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

\texttt{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 \LaTeXe. The package also requires the \texttt{reledmac} package, however we do not load it automatically, because we prefer users to know it.

\begin{verbatim}
\%<code>
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{reledpar}[2020/11/29 v2.25.0 reledmac extension for parallel texts]%
\%
\end{verbatim}

II.2 Package’s requirement

Few commands use \texttt{xspace} command.

\begin{verbatim}
\%<code>
\RequirePackage{xspace}%
\%
\end{verbatim}

II.3 Package’s options

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

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

II.4.1 Synchronization’s options

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

\newif\ifshiftedpstarts
\DeclareOptionX{shiftedpstarts}{\shiftedpstartstrue\apptocmd{\par\this\sync\option}{shifted}{}}

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

\newif\ifadvancedshiftedpstarts
\DeclareOptionX{advancedshiftedpstarts}{\advancedshiftedpstartstrue\shiftedpstartstrue\apptocmd{\par\this\sync\option}{advancedshifted}{}}

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

\newif\ifnomaxlines
\DeclareOptionX{nomaxlines}{\nomaxlinestrue\apptocmd{\par\this\sync\option}{nomax}{}}

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

\newif\ifnosyncpstarts
\DeclareOptionX{nosyncpstarts}{\nosyncpstartstrue\apptocmd{\par\this\sync\option}{nosync}{}}

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

II.4.2 Position options

The \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.

\newif\ifmovecolumnspositiononrightpage
\DeclareOptionX{movecolumnspositiononrightpage}{\movecolumnspositiononrightpagetrue}
%

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

\newif\ifsidenotesmarginpage
\DeclareOptionX{sidenotesmarginpage}{\sidenotesmarginpagetrue}
%

II.4.3 Other options

The parledgroup can be called either on reledmac or reledpar.

\newif\ifparledgroup
\DeclareOptionX{parledgroup}{\parledgrouptrue}
%

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

\newif\ifwidthliketwocolumns
\DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}
\DeclareOptionX{continuousnumberingwithcolumns}{\continuousnumberingwithcolumnstrue}
%

Options related to page numbering. The booleans are defined in reledmac.
II.5 Determining side and category of parallel processing

As noted above, much of the code is a duplication of the original \texttt{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 \texttt{Leftside} and \texttt{Rightside} environments set things up appropriately.

\texttt{\ifl@dpairing} \texttt{\ifl@dpaging} \texttt{\ifledRcol}

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

II.6 Text’s width

\texttt{\Lcolwidth} \texttt{\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 \texttt{pages} environment and in the \texttt{Pages} command.

\texttt{\newdimen\Lcolwidth}
\texttt{\Lcolwidth=0.45\textwidth}
\texttt{\newdimen\Rcolwidth}
\texttt{\Rcolwidth=0.45\textwidth}
\texttt{\newdimen\Lcolwidth@pages}
\texttt{\newdimen\Rcolwidth@pages}

II.7 Messages

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

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

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

\led@err@TooManyPstarts { \newcommand*{\led@err@TooManyPstarts}{{ %
\reledpar@error{Too many \string\pstart\space without printing.\n\begin{itemize}
\item Some text will be lost\end{itemize}{\@ehc}}} %

\led@err@polyglossiaTooOld { \newcommand{\led@err@polyglossiaTooOld}{{ %
\reledpar@error{You use the polyglosia package.\n\MessageBreak However, reledpar requires version 2020/04/08 v1.49 or \n\MessageBreak later of the polyglosia package.}{\@ehc}} %

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

\providebool{syntax@}
\led@err@LeftOnRightPage { \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}} %
}}

\led@err@Leftside@PreviousNotPrinted { \newcommand*{\led@err@Leftside@PreviousNotPrinted}{{ %
\reledpar@error{You call a new Leftside environment while the previous\none has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}

\newcommand*{\led@err@Rightside@PreviousNotPrinted}{{ %
\reledpar@error{You call a new Rightside environment while the previous\none has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}

%
II.7 Messages

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

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

\newcommand*{\led@err@Pages@WithoutRightside}{% \reledpar@error{\string\Pages space called without any preceding \texttt{\%} `Rightside' environment (or environment without content)}{\@ehc}}
\newcommand*{\led@err@Pages@WithoutLeftside}{% \reledpar@error{\string\Pages space called without any preceding \texttt{\%} `Leftside' environment (or environment without content)}{\@ehc}}
\newcommand*{\led@err@Columns@WithoutRightside}{% \reledpar@error{\string\Columns space called without any preceding \texttt{\%} `Rightside' environment (or environment without content)}{\@ehc}}
\newcommand*{\led@err@Columns@WithoutLeftside}{% \reledpar@error{\string\Columns space called without any preceding \texttt{\%} `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 right side, despite your configuring it to be for the left side only}{\@ehc}}
\newcommand{\led@error@note@called@onleftside}[1]{%\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}%
}\%
%
\newcommand{\led@error@edtext@later@now}{%\reledpar@error{You call \the@edtext@later\space \string@edtextlater\space commands, but \the@edtext@now\space \string@edtextnow\space commands \@ehc}%(did I follow your intent correctly here?)%
}\%
%
\newcommand*{\led@warn@ChangeSyncOption}[1]{%\reledpar@warning{You have changed synchronization’s options since the last run. We have not read line-list file #1. Please run LaTeX again.}%
}\%
%
\newcommand{\led@warn@setting@in@rightside}[1]{%\reledpar@warning{You use #1 inside rightside environment.\MessageBreak Such behavior is deprecated.\MessageBreak Use instead #1R or #1* in your preamble.}%
}\%
%
\newcommand{\led@error@missing@numbering}[1]{%\reledpar@error{Missing \string@pstart\space ...\space\string@pend\space inside ‘#1’ environment}{\@ehc}%
}\%
II.8 Naming macros

The \LaTeX 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.

\begin{verbatim}
\newnamebox \setnamebox \unhnamebox \unvnamebox \namebox
A set of macros for creating and using "named" boxes; the macros are called after the regular box macros, but including the string "name".
\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}

\begin{verbatim}
\ifcsboxvoid
The \texttt{\ifcsboxvoid} command comes on top of the etoolbox package. The first argument is a box name, the second is the \texttt{(true)} value, the third the \texttt{(false)} value. If the first argument refers to a non-existing box, or if it refers to a box that is void, the \texttt{(true)} value is returned. otherwise, \texttt{(false)} is returned.
\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}}
\end{verbatim}

\begin{verbatim}
\newnamecount \usenamecount
Macros for creating and using "named" counts.
\providecommand*{\newnamecount}[1]{\expandafter\newcount\csname #1\endcsname}
\providecommand*{\usenamecount}[1]{\csname #1\endcsname}
\end{verbatim}


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

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

195 \newcount\section@numR
196 \section@numR=200
197 %

\ifpstratedL \ifpstratedL is set FALSE at the start of left side numbering, and similarly for
\ifpstratedR \ifpstratedR. \ifpstratedL is defined in reledmac.

198 \pst@rtedLfalse
199 \newif\ifpstratedR
200 \%

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

202 \newcommand*{\beginnumberingR}{% 
203 \ifnumberingR
204 \led@err@NumberingStarted
205 \endnumberingR
206 \fi
207 \global\section@numR \z@
208 \global\pstratedRfalse
209 \global\numberingRtrue
210 \global\afterendnumberingRfalse%
211 \global\advancesection\section@numR \z
212 \global\absline@numR \z
213 \gdef\normal@page@breakR{}
214 \gdef\@oprev0pbR{}
215 \gdef\@oprev0nopbR{}
216 \global\line@numR \z
217 \global\@lockR \z
218 \global\@sublockR \z
219 \global\sublines@Rfalse
220 \global\let\next@page@numR\relax
221 \global\let\this@section@next@page@numR\relax%
222 \global\let\sub@change\relax
223 \global\last@page@numR=-10000%
224 \global\stopmsdata@inserted@true%
225 \global\let@msdata@add@msd@cR@dataR
226 \global\csundef{@msdata@add@msd@cR@dataR}%
227 \ifcontinuousnumberingwithcolumns%
This is the left text version of the regular `\endnumbering` and must follow the last text for a left text numbered section. It sets `\ifpstedL` 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.
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 \texttt{numbering\_R}.

\begin{align*}
\def\initnumbering@sectcountR{} & \def\save@section@number{} & \def\set@section@number{} \\
\setcounter{chapterR}{\old@chapter} & \setcounter{sectionR}{\old@section} & \setcounter{subsectionR}{\old@subsection} & \setcounter{subsubsectionR}{\old@subsubsection}]
\end{align*}
\newcommand{\set@sectcountR}{% 
\let\c@chapter\c@chapterR% 
\let\c@section\c@sectionR% 
\let\c@subsection\c@subsectionR% 
\let\c@subsubsection\c@subsubsectionR% 
}%
\pausenumberingR
\resumenumberingR
\ifresumenumberingR@start
\pausenumberingR@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@numRR%
\fi%
\endnumberingR% 
\global\numberingRtrue%
\global\afterendnumberingRfalse%
}%
\newif\ifresumenumberingR@start%
\newcommand*{\resumenumberingR}{% 
\ifnumberingR 
\global\pst@rtedRtrue 
\global\advance\section@numR \@ne 
\global\resumenumberingR@starttrue%
\led@mess@SectionContinued{\the\section@numR R}% 
\line@list@stuffR{\jobname\extensionchars\the\section@numR R}% 
\l@dend@stuff
\begingroup
\initnumbering@sectcountR% 
\else
\led@err@NumberingShouldHaveStarted 
\endnumberingR 
\beginnumberingR 
\fi%
}%
\memorydumpL
\memorydumpR
\memorydump
These are the right text equivalents of \pausenumbering and \resumenumbering.
\newcommand{\pausenumberingR}{% 
\newcommand*{\pausenumberingR}{% 
\ifx\this@section@next@page@numR\relax% 
\global\pausenumbering@page@numR=0% 
\else%
\global\pausenumbering@page@numR=\this@section@next@page@numRR%
\fi%
\endnumberingR% 
\global\numberingRtrue%
\global\afterendnumberingRfalse%
}%
\newcommand*{\resumenumberingR}{% 
\ifnumberingR 
\global\pst@rtedRtrue 
\global\advance\section@numR \@ne 
\global\resumenumberingR@starttrue%
\led@mess@SectionContinued{\the\section@numR R}% 
\line@list@stuffR{\jobname\extensionchars\the\section@numR R}% 
\l@dend@stuff
\begingroup
\initnumbering@sectcountR% 
\else
\led@err@NumberingShouldHaveStarted 
\endnumberingR 
\beginnumberingR 
\fi%
}%
\memorydumpL
\memorydumpR
\memorydump
is a shorthand for \pausenumbering \resumenumbering. This will clear
the memorised stuff for the previous chunks while keeping the numbering going.
IV Line counting

IV.1 Setting lineation reset

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

\lineationR \lineationR{\textit{word}} is the macro used to select the lineation system for right texts. Its argument is a string: either page, pstart or section.
IV.2 Setting line number margin

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.

By default put right text numbers at the right.

\newcount\line@marginR
\line@marginR=\@ne
\linenummarginColumns* \linenummarginColumns \linenummarginColumnsR

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

\newcount\line@margin@columnsR %
\newcommand\linenummarginColumns[
]{% %
}\linenummarginColumns[
]{% %
} %
\WithSuffix\newcommand\linenummarginColumns*[n]{% %
}\linenummarginColumns*[n]{% %
} %
\newcommand\linenummarginColumnsR{l}{% %
}\linenummarginColumnsR{l}{% %
} %
\linenumOnlyPagesForColumns %
\linenumOnlyPagesForColumnsR %
\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 \texttt{reledmac} which right text lines should be printed with line numbers. $\texttt{firstlinenumR}$ is the number of the first line in each section that gets a number; $\texttt{linenumincrementR}$ is the difference between successive numbered lines. The initial values of these counters produce labels on lines 5, 10, 15, etc. $\texttt{linenumincrementR}$ 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 $\texttt{firstlinenumR}$ and $\texttt{linenumincrementR}$, but for sub-line numbers. $\texttt{sublinenumincrementR}$ 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 \texttt{reledmac}. The starred versions are specific to \texttt{eledpar}.

\begin{verbatim}
\newcommand\firstlinenum*[1]{%
\setcounter{firstlinenumR}{#1}%
\setcounter{firstlinenum}{#1}%
}
\newcommand\linenumincrement*[1]{%
\setcounter{linenumincrementR}{#1}%
\setcounter{linenumincrement}{#1}%
}
\newcommand\firstsublinenum*[1]{%
\setcounter{firstsublinenumR}{#1}%
\setcounter{firstsublinenum}{#1}%
}
\newcommand\sublinenumincrement*[1]{%
\setcounter{sublinenumincrementR}{#1}%
\setcounter{sublinenumincrement}{#1}%
}
\end{verbatim}
IV. Line counting

\firstlinenumR \linenumincrementR \firstsublinenumR \sublinenumincrementR

And the “R” suffixed version.

\newcommand\firstlinenumR[1]{\setcounter{firstlinenumR}{#1}}
\newcommand\linenumincrementR[1]{\setcounter{linenumincrementR}{#1}}
\newcommand\firstsublinenumR[1]{\setcounter{subfirstlinenumR}{#1}}
\newcommand\sublinenumincrementR[1]{\setcounter{sublinenumincrementR}{#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 \linenumrepR{\langle\textlangle\rangle} typesets the right line number \langle\textlangle\rangle, 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\langle\textlangle\rangle}{\@nameuse{@#1}{\langle\textlangle\rangle}}}
\newcommand*{\sublinenumberstyleR}[1]{\def\sublinenumrepR\langle\textlangle\rangle}{\@nameuse{@#1}{\langle\textlangle\rangle}}}

\linenumberstyle* \sublinenumberstyle* And for both side.
IV.6 Print marginal line number

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

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

\newcommand*{\leftlinenum}{{}}
   \l@dlinenum
   \kern\linenumsep
\newcommand*{\rightlinenum}{{}}
   \kern\linenumsep
   \l@dlinenum
   \ifsublines
   \ifnum\subline>\z@
   \unskip\fullstop\sublinenumrep{\subline}
   \fi
\fi
\linenumrep{\line}\@lineflag
\ifsublines\sublinenumrep{\subline}\fi
\unskip
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.

The boolean \texttt{ifsublines@R} is set to true when we start subline on rightside.
Now we can define the list macros that will be created from the line-list file. They are directly analogous to the left text ones. The full list of action codes and their meanings is given in the reledmac manual.

Here are the commands to create these lists:

```latex
\list@create{\line@listR}
\list@create{\insertlines@listR}
\list@create{\actionlines@listR}
\list@create{\actions@listR}
\list@create{\annot@listR}
```

The right text page numbers.

```latex
\page@numR
\this@section@page@numR
```

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.

```latex
\linesinpar@listL
\linesinpar@listR
\maxlinesinpar@list
```

Clear the right lines for \read@linelist

```latex
\list@clearing@regR
\list@clear{\line@listR}
\list@clear{\insertlines@listR}
\list@clear{\actionlines@listR}
\list@clear{\actions@listR}
\list@clear{\annot@listR}
```

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.

```latex
\par@sync@option
\newcommand{\par@sync@option}{1}{% 
  \IfStrEq{#1}{\par@this@sync@option}{}% 
}
```
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.

\read@linelist \read@linelist{\textit{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.

\@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\@dchset@num\relax \else 
 \advance\absline@numR \m@ne 
 \cs@def{\@dchset@num@R\the\absline@numR}{}%To remember this line have been marked by a \setlinenum 
 \set@line@action 
 \let\@dchset@num\relax 
 \advance\absline@numR \m@ne 
 \advance\line@numR \m@ne% % do we need this? 
\fi 
\reset@current@annot 
\advance\absline@numR \m@ne 
\ifx\next@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 
\fi 
\fi 
\fi 
\fi 
\fi 
\fi
IV.9 Commands within the line-list file

\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
\ifsublines@R
\ifnum\sub@lockR<\tw@
\advance\subline@numR \@ne
\fi
\else
\ifnum\@lockR<\tw@
\advance\line@numR \@ne \subline@numR \z@
\fi
\fi

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

% \@set
The \@set{⟨num⟩} macro sets the current visible line number to the value specified as its argument. This is used to implement \setline. It is defined in reledmac.

% \l@d@set
The \l@d@set{⟨num⟩} macro sets the line number for the next \pstart... to the value

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

% \newcount\last@page@numR
\last@page@numR=-10000
% \newcount\this@section@last@page@numR
\this@section@last@page@numR=-10000
%
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@lockonR \lock@on adds an entry to the action-code list to turn line number locking on. The current setting of the sub-lineation flag tells us whether this applies to line numbers or sub-line numbers. It is defined in reledmac, however the code specific to right side is defined here, in \do@lockonR.

\lock@off \do@lockoff \do@lockoffR \skip@lockoff \lock@off adds an entry to the action-code list to turn line number locking off. It is defined in reledmac, however the code specific to right side is defined here, in \do@lockoffR.
IV.9 Commands within the line-list file

\newcommand{\do@lockoffR}{%
\xright@appenditem{\the\absline@numR} to \actionlines@listR
\ifsuitlinesR
\xright@appenditem{\-1006} to \actions@listR
\ifnum\sub@lockR=\tw@
\sub@lockR \thr@@
\else
\sub@lockR \z@
\fi
\else
\xright@appenditem{\-1004} to \actions@listR
\ifnum\@lockR=\tw@
\@lockR \thr@@
\else
\@lockR \z@
\fi
\fi}
%
\n@num

\n@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 \n@ref commands, corresponding to uses of \edtext within the first argument of another instance of \edtext.

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

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

\newcommand*{\n@refregR}[2]{{%
\global\advance\edtext@level by %
\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 executes its argument, so that nested \@ref commands are just skipped this time. Some other macros need to be temporarily redefined to suppress their action.

\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@listR.

\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.

\expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\@edtext@level\endcsname}%
%
And now, call \@ref@reg@parseargR, which can be also called by \@ref@later \@ref@reg@parseR{#2}%
%
Decrease edtext level counter.
IV.9 \textit{Commands within the line-list file}

\newcommand{\@ref@reg@parseR}[1]{
\begin{verbatim}
% 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@edtextR@\the@edtext@level\endcsname}}%
% \end{macrocode}
\ifcsundef{sw@list@edtextR@\the@edtext@level}{\create@this@edtext@level}{}%
\letcs{\@tmp}{sw@list@edtext@tmp@\the@edtext@level}%
\letcs{\@tmpp}{sw@list@edtext@\the@edtext@level}%
\xright@appenditem{\expandonce\@tmpp}to\@tmp%
\global\cslet{sw@list@edtextR@\the@edtext@level}{\@tmp}%
\fi%
\fi

% \end{verbatim}
\end{verbatim}

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

\renewcommand*{\@pend}[1]{
\begin{verbatim}
% \begin{verbatim}
\ifbypstart{\@pend}{[1]{%
\unless\ifinstanza%
\global\line@num=0%
\fi%
\fi%
\xright@appenditem{#1}to\linesinpar@listL%
}
\end{verbatim}
\end{verbatim}

\renewcommand*{\@pendR}{[1]{%
\ifbypstartR{\@pend}{[1]{%
\unless\ifinstanzaR%
\global\line@numR=0%
\fi%
\fi%
\xright@appenditem{#1}to\linesinpar@listR%
}
\end{verbatim}

\end{verbatim}
\@pstart and \@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{afterlines@pstart@#1L}{#3}}}{\csnumgdef{minpage@pstart@#1}{#2}}\%\}%\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{afterlines@pstart@#1R}{#3}}}{\csnumgdef{afterlines@pstart@#1R}{#3}}\%\}%\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.
IV.10 Writing to the line-list file

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. reledmac and reledpar can fill the next@line@list@stuffR hook between a \endnumberingR (associated with numbered file n) and a \beginnumberingR (associated with numbered file n + 1). It allows adding content to the numbered file n + 1R and not nR.

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

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

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

\starts@sub \endsub \starts@sub and \endsub turn sub-lineation on and off, by writing appropriate instruc-
tions to the line-list file. There are both defined in reledmac.

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

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

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

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

\skipnumbering

V Marking text for notes

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

\critext \edtext \set@line The \set@line macro is called by \edtext to put the line-reference field and font
specifier for the current block of text into \l@d@nums. It is defined in reledmac.
V.1 Specific hooks and commands for notes

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

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

V.1.1 Notes to be printed on one side only

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

\unless\ifnofamiliar@%
\csgdef{onlysideX@#1}{}%
\newnamebox{footins#1@kept}%
\fi%
\unless\ifnocritical@%
\csgdef{Xonlyside@#1}{}%
\newnamebox{#1footins@kept}%
\fi%
%
V.1.2 Tools specific to familiar footnotes

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

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

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

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

\begin{verbatim}
\newcounter{footnote@typeset}
\end{verbatim}

**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@mk\endcsname
\end{verbatim}

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

\begin{verbatim}
\expandafter\newcommand\csname footnote@nomk\endcsname[1]{
\toggletrue{nom@}
\csuse{footnote@}{##1}
\togglefalse{nom@}
}
\end{verbatim}

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

\begin{verbatim}
\toggletrue{nom@}
\csuse{footnote@}{##1}
\togglefalse{nom@}
\end{verbatim}

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}.

\begin{verbatim}
\letcs{@tmp}{footnote@mk}
\gl@p@tmp@to@tmpa
\global\cslet{footnote@mk}{@tmp}
\end{verbatim}

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

\begin{verbatim}
\expandafter\newcommand\csname footnote@mk\endcsname{
\letcs{@tmp}{footnote@mk}
\numdef{@tmpa}{\csuse{c@footnote@}}
\global\xright@appenditem{@tmpa}{@tmp}
\global\cslet{footnote@mk}{@tmp}
}
\end{verbatim}

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

\begin{verbatim}
\letcs{@tmp}{footnote@mk}
\gl@p@tmp@to@tmpa
\global\cslet{footnote@mk}{@tmp}
\end{verbatim}
Set the footnotecounter with it. For the sake of security, we make a backup of the previous value.

\letcs{\old@footnote}{c@footnote#1}\
\setcounter{footnote#1}{\@tmpa}\

Define the footnote mark and print it

\protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}\
\csuse{@footnotemark#1}\

Restore previous footnote counter and finally add space.

\setcounter{footnote#1}{\old@footnote}\
\xspace\

End of tools specific to familiar notes.

\fi

End of newseries@par.

V.1.3 Get correct footnote number

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

\newcommand{\save@familiarfootnote@number}{\%
  \unless\ifnofamiliar0\%
    \def\do##1{\csxdef{saved@footnote##1}{\the\csname c@footnote##1\endcsname}}\%
  \dolistloop{\@series}\
  \fi\%
  \xdef\saved@footnote{\the@c@footnote}\
}\%
\newcommand{\get@familiarfootnote@number}{\%
  \unless\ifnofamiliar0\%
    \def\do##1{\setcounter{footnote##1@typeset}{\csuse{saved@footnote##1}}}
  \dolistloop{\@series}\
  \fi\%
  \setcounter{footnote@typeset}{\saved@footnote}\
}\%
V.2 Create hooks

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

```
\unless\fnocritical\%
  \newhookcommand@series{Xonlyside}\%
\fi\%
\unless\fnnofamiliar\%
  \newhookcommand@series{onlysideX}\%
\fi\%
```

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

`\init@series@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. 55).

```
\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”.

```
\newcommand{\pstartpc}{\pstart}
\newcommand{\endpstartpc}{\endpstart}
\newcommand{\startpc}{\start}
\newcommand{\endpc}{\end}
\newcommand{\onlyside}{\onlyside}
\newcommand{\onlysideX}{\onlysideX}
```

```
\unless\fnocritical\%
  \newhookcommand@series{Xonlyside}\%
\fi\%
\unless\fnnofamiliar\%
  \newhookcommand@series{onlysideX}\%
\fi\%
\unless\fnctext\%
  \newhookcommand@series{text}\%
\fi\%
```
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).

\hspace{1cm}

\def\dump@pstartL@pc{%  
  \xright@appenditem{\the\c@pstartL}{\to}\list@pstartL@pc%  
  \global\cslet{numberpstart@L\the\l@dnumpstartsL}{\ifnumberpstart}%  
  \global\cslet{numberline@L\the\l@dnumpstartsL}{\ifnumberline}%  
  \global\cslet{linenumannotationotherside@L\the\l@dnumpstartsL}{\iflinenumannotationotherside}%  
}%

\def\dump@pstartR@pc{%  
  \xright@appenditem{\the\c@pstartR}{\to}\list@pstartR@pc%  
  \global\cslet{numberpstart@R\the\l@dnumpstartsR}{\ifnumberpstart}%  
  \global\cslet{numberline@R\the\l@dnumpstartsR}{\ifnumberline}%  
  \global\cslet{linenumannotationotherside@R\the\l@dnumpstartsR}{\iflinenumannotationotherside}%  
}%

And so, the commands to restore them.

\def\restore@pstartL@pc{%  
  \ifx\list@pstartL@pc\empty\else%  
    \g@p\list@pstartL@pc\to\@temp%  
    \global\c@pstartL=\@temp%  
  \fi%  
}%

\def\restore@pstartR@pc{%  
  \ifx\list@pstartR@pc\empty\else%  
    \g@p\list@pstartR@pc\to\@temp%  
    \global\c@pstartR=\@temp%  
  \fi%  
}%

VII Parallel environments

The initial set up for parallel processing is deceptively simple.

pairs pages
chapterinpages The \texttt{pairs} environment is for parallel columns and the \texttt{pages} environment for parallel pages.

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

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

The \texttt{pages} environment additionally sets the “column” widths to the \texttt{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%
  \global\Lcolwidth@pages=\Lcolwidth%Does not use \texttt{\setlength}, in case of calc package redefinition (cf. https://github.com/latex3/latex2e/issues/58)
  \global\Rcolwidth@pages=\Rcolwidth%
% \AtBeginPairs
\end{verbatim}
\end{verbatim}
Leftside Within the pairs and pages environments the left and right hand texts are within Leftside and Rightside environments, respectively. The Leftside environment is simple, indicating that right text is not within its purview and using some particular macros.

\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{\global\instanzaLtrue\old@startstanza[\#1][\#2]}\%
\expandafter\ifvoid\csname l@dLcolrawbox1\endcsname\%
\led@error@missing@numbering{Leftside}\%
\fi\%
\Leftsidehookend}
%

\Leftsidehook \Leftsidehookend \Rightsidehook \Rightsidehookend

Hooks into the start and end of the Leftside and Rightside environments. These are initially empty.
\newcommand*{\Leftsidehook}{\%
\newcommand*{\Leftsidehookend}{\%
\newcommand*{\Rightsidehook}{\%
\newcommand*{\Rightsidehookend}{\%
%
\Rightpagehook \Leftpagehook

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.
\newenvironment{Rightside}{\%}
\expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else\%
\led@err@Rightside@PreviousNotPrinted\%
\fi\%
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 `/f_lags/` 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.


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

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

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

\pstartR  

\newcounter{pstartL}
\renewcommand{\the\pstartL}{{\bfseries\@arabic\c@pstartL}. }
\newcounter{pstartR}
\renewcommand{\the\pstartR}{{\bfseries\@arabic\c@pstartR}. }

\newcommandx*{\pstartL}[2][1,2,usedefault]{%  
\if@nobreak%
\let\@oldnobreak\@nobreaktrue%
\else%
\let\@oldnobreak\@nobreakfalse%
\fi%
\@nobreaktrue%
\ifluatex%
\xdef\l@luatextextdir@L{\the\textdir}%
\xdef\l@luatexpardir@L{\the\pardir}%
\xdef\l@luatexbodydir@L{\the\bodydir}%
\fi%
\ifnumbering \else%
\led@err@PstartNotNumbered%
\beginnumbering%
\fi%
\ifnumberedpar%
\led@err@PstartInPstart%
\fi%
\fi%
%
If this is the first \pstart in a numbered section, clear any inserts and set \ifpst@rtedL
to FALSE.

\ifpst@rtedL\else%
\list@clear{\inserts@list}%
\global\let\next@insert=\empty%
\global\pst@rtedLtrue%
\fi%
\begingroup\everypar{}}%
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.

Dump the optional arguments
VIII.1 Boxes, counters, \pstart and \pend

Let \by@autopar to FALSE.
\global\by@autoparfalse
%
Gobble following space (automatically done if there is no optional argument)
\ignorespaces%
}
%
The same for right side.
\newcommandx*\pstartR[2][1,2,usedefault]{% 
\if@nobreak%
\let\@oldnobreak\@nobreaktrue%
\else%
\let\@oldnobreak\@nobreakfalse%
\fi%
\@nobreaktrue%
\ifluatex%
\xdef\luatextextdirR{\the\textdir}%
\xdef\luatexpardirR{\the\pardir}%
\xdef\luatexbodydirR{\the\bodydir}%
\fi%
\ifnumberingR \else%
\le@err@PstartNotNumbered%
\beginnumberingR%
\fi%
\ifndefnumberedR \else%
\le@err@PstartInPstart%
\pendR%
\fi%
\ifndefnumberedparR
\le@err@PstartInPstart%
\pendR%
\fi%
\ifndefnumberedR\else%
\list@clear\inserts@listR%
Paragraph decomposition and reassembly

\global\let\next@insertR=\empty%
\global\pstartRtrue%
\fi
\begingroup\everypar{}
\global\ad\pstartR\one%
\ifnum\l@dnumpstartsR=\l@dc@maxchunks%
\led@err@TooManyPstarts%
\global\l@dnumpstartsR=\l@dc@maxchunks%
\fi
\global\setnamebox{l@dRcolrawbox\the\l@dnumpstartsR}=%vbox\bgroup%
\l@dzeropenalties%
\ifboolexpr{\bool{autopar} and bool{by@autopar}}{}
\ifnumberpstart\ifsidepstartnum\else\thepstartR\fi\fi\@at@start@every@pstart%
\hsize=Rcolwidth%
\numberedpar@true%
\iflabelpstart%
\protected@edef\@currentlabel{\p@pstartR\thepstartR}%
\fi%
\providetoggle{before@pstartR@\the\l@dnumpstartsR @par}%
\ifboolexpr{\test{#1} and test{#2}}%
\ifat@every@pstart@star@%
\global\togglefalse{before@pstartR0\the\l@dnumpstartsR @par}%
\else%
\global\toggletrue{before@pstartR0\the\l@dnumpstartsR @par}%
\fi%
\csgdef{before@pstartR0\the\l@dnumpstartsR}{\at@every@pstart}%
\global\togglefalse{\noindent#1}%
\csgdef{before@pstartR0\the\l@dnumpstartsR}{\noindent#2}%
\csgdef{before@pstartR0\the\l@dnumpstartsR}{#2}%
\pendL \pend must be used to end a numbered paragraph. Again we need a version that knows about left parallel texts.

We immediately call \endgraf to end the paragraph; this ensures that there will be no large interline penalties to prevent us from slicing the paragraph into pieces.

End the group that was begun in the \pstart.

Dump content of the optional argument.
The version of `\pend` needed for right texts.

\newcommandx{\pendR}[2][1,2,usedefault]{%
  \ifnumberingR \else%
     \led@err@PendNotNumbered%
  \fi%
  \ifnumberedparR \else%
     \led@err@PendNoPstart%
  \fi%
  \at@end@every@pend%
  \endgraf\global\num@linesR=\prevgraf\egroup%
  \global\par@lineR=0%
  \get@firsthalf@antilabe@wd%
  \endgroup%
  \ignorespaces%
  \@oldnobreak%
  \dump@pstartR@pc%
  \ifnumberpstart%
     \addtocounter{pstartR}{1}%
  \fi%
  \parledgroup@beforenotesR@save{R}
  \providetoggle{after@pendR@	he\l@dnumpstartsR @par}%
  \ifboolexpr{\test {\ifstrempty{#1}}\and\test {\ifstrempty{#2}}}{%
    \ifat@every@pend@star%
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 ...
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.
VIII.2 Processing one line

\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}%
}

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

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

\newcommand{\addapparatusL}{%
  \if@firstlineofpage%
These high-level commands just redefine the low-level commands. They have to be used by user, without \makeatletter.

\newcommand*{\do@lineLhook}{\gdef\do@lineLhook{#1}}
\newcommand*{\do@lineRhook}{\gdef\do@lineRhook{#1}}
\newcommand*{\do@insidelineLhook}{\gdef\do@insidelineLhook{#1}}
\newcommand*{\do@insidelineRhook}{\gdef\do@insidelineRhook{#1}}

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

\newcommand*{\do@lineLhook}{\{} \newcommand*{\do@lineRhook}{\{} \newcommand*{\do@insidelineLhook}{\{} \newcommand*{\do@insidelineRhook}{\{}
\newcommand*{\do@lineLhook}{\{ \newcommand*{\do@lineRhook}{\{ \newcommand*{\do@insidelineLhook}{\{ \newcommand*{\do@insidelineRhook}{\{

The \do@lineR macro is called to do all the processing for a single line of right text.
VIII Paragraph decomposition and reassembly

\ifvbox\namebox{10dRcolrawbox\the\l0dpscR} \\
{vbadness=10000% \\
\spittopskip=\z@ \\
do1lineRhook% \\
10emptyd0ta% \\
global\setbox\one0lineR=\vsplit\namebox{10dRcolrawbox\the\l0dpscR} to\baselineskip} \\
\IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\parledgroup@notes@startR}{}% \\
unvbox\one0lineR \global\setbox\one0lineR=\lastbox% \\
@writepageofparR% \\
getline@numR% \\
@ifnum\@lockR>\@ne% \\
inserthangingsymbolRtrue% \\
else% \\
inserthangingsymbolRfalse% \\
fi% \\
setbox\10drightbox% \\
hb\@xt\ Rcolwidth{% \\
@if0\@dhidenumerator% \\
global\10dhidenumeratorfalse% \\
f0\@lockesR% \\
else% \\
affixline@numR% \\
f0% \\
xifinlist{\the\l0dpscR}{\eled@sectionsR@@}{} \\
{\unless\ifshiftedpstarts% \\
\add@apparatusR% \\
\fi% \\
}% \\
\% \\
\{print@lineR% \\
}\% \\
\add@penaltiesR% \\
global\advance\@donetotallinesR\@ne% \\
\else% \\
@iflinenumberRevenifblank% \\
\setbox\10drightbox \hb\@xt\ Rcolwidth{% \\
 new0lineR% \\
10emptyd0ta% \\
getline@numR% \\
affixline@numR% \\
10dld0ta% \\
hspace*{\Rcolwidth}% \\
1edrlfill10drd0ta% \\
}% \\
\else% \\
\setbox\10drightbox \hb\@xt\ Rcolwidth{% \\
 ifcontinuousnumberingwithcolumns% \\
 new0lineR% \\
\hspace*{\Rcolwidth}% \\
1edrlfill10drd0ta% \\
}% \\
\else% \\
}\% \\
\advance\@donetotallinesR\@ne% \\
\advance\@donereallinesR\@ne% \\
\else% \\
\iflinenumberRevenifblank% \\
\setbox\10drightbox \hb\@xt\ Rcolwidth{% \\
 new0lineR% \\
10emptyd0ta% \\
getline@numR% \\
affixline@numR% \\
10dld0ta% \\
hspace*{\Rcolwidth}% \\
1edrlfill10drd0ta% \\
}% \\
\else% \\
\setbox\10drightbox \hb\@xt\ Rcolwidth{% \\
 ifcontinuousnumberingwithcolumns% \\
 new0lineR% \\
\hspace*{\Rcolwidth}% \\
1edrlfill10drd0ta% \\
}% \\
\else% \\
}
VIII.3 Line and page number computation

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

\newcommand{\add@apparatusR}{%
  \if@firstlineofpageR%
    \set@Xtxtbeforenotes%
    \set@txtbeforenotesX%
    \global\@firstlineofpageRfalse%
  \fi%
  \ifndefstring{\ms@data@position}{msdata-regular}{%
    \insert@msdata%
    \add@insertsR%
    \add@Xgroupbyline%
  }{%
    \add@insertsR%
    \add@Xgroupbyline%
    \insert@msdata%
  }%
  \affixside@noteR%
}%

\newline
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.
\ifsublines\R\fi
\ifnum\sub@lock<\tw@\global\advance\subline@num \@ne\fi
\else\fi
\ifnum\@lock<\tw@\global\advance\line@num \@ne\global\subline@num \z@\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi

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

\newcommand*{\do@ballastR}{\global\ballast@count=\z@\begingroup \advance\absline@num \@ne\ifnum\next@actionlineR=\absline@num \@ne\ifnum\next@actionR>-1001\global\advance\ballast@count by -\c@ballast\fi\fi\endgroup}
VIII.3  Line and page number computation

The \doactionsR 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 \doactions@nextR for this.

```latex
\newif\if@doskipversenumberR
\newcommand*{\doactions@fixedcodeR}{%
  \ifcase@1\@dtempcnta%
    \or% 1001 = starting sublineation
      \global\sublines@Rtrue
    \or% 1002 = ending sublineation
      \global\sublines@Rfalse
    \or% 1003 = starting locking number
      \global\@lockR=\one
    \or% 1004 = ending locking number
      \ifnum\@lockR=\tw@
        \global\@lockR=\thr@@
      \else
        \global\@lockR=\z@
      \fi
    \or% 1005 = starting locking subnumber
      \global\sub@lockR=\one
    \or% 1006 = ending locking subnumber
      \ifnum\sub@lockR=\tw@
        \global\sub@lockR=\thr@@
      \else
        \global\sub@lockR=\z@
      \fi
    \or% 1007 = skipping numbering
      \@doskipnumbertrue
    \or% 1008 = skipping numbering in stanza
      \@doskipversenumbertrue
    \or% 1009 = hiding number
      \@dhidenumbertrue
    \or% 1010 = inserting msdata
      \add@msdata
    \else%
      \if@dowarn@BadAction
    \fi%
  \else%
    \fi%
} \newcommand*{\doactionsR}{%
  \global\let\doactions@nextR=\relax
  \@dtempcntb=\absline@numR
  \ifnum\@dtempcntb<\next@actionlineR\else
    \ifnum\@dtempcntb<\next@actionlineR\else
      \ifboolexpr{\%}
        \booltrue\resumenumberingR\start{%
```

```latex
\global\let\doactions@nextR=\relax \
\@dtempcntb=\absline@numR \\n@ifnum\@dtempcntb<\next@actionlineR\else \\n@ifnum\@dtempcntb<\next@actionlineR\else \\n@if\%\then\%\relax \\n\ifboolexpr{\%}
  \booltrue\resumenumberingR\start{%
```
VIII.4 Line number printing

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

\newcommand*{\l@dcalcnunm}[3]{%  
  \ifnum #1 > #2\relax  
  \@l@dtempcnta = #1\relax  
  \advance\@l@dtempcnta by -#2\relax  
  \divide\@l@dtempcnta by #3\relax  
}
\multiply\@dtempcnta by #3\relax
\advance\@dtempcnta by #2\relax
\else
\@dtempcnta=#2\relax
\fi}
\newcommand*{\ch@cksub@l@ckR}{%
\ifcase\sub@lockR
\or
\ifnum\sublock@disp=\@ne
\@dtempcntb \z@ \@dtempcnta \@ne
\fi
\or
\ifnum\sublock@disp=\tw@
\else
\@dtempcntb \z@ \@dtempcnta \@ne
\fi
\or
\ifnum\sublock@disp=\z@
\@dtempcntb \z@ \@dtempcnta \@ne
\fi
\fi}
\newcommand*{\ch@ck@l@ckR}{%
\ifcase\@lockR
\or
\ifnum\lock@disp=\@ne
\@dtempcntb \z@ \@dtempcnta \@ne
\fi
\or
\ifnum\lock@disp=\tw@
\else
\@dtempcntb \z@ \@dtempcnta \@ne
\fi
\or
\ifnum\lock@disp=\z@
\@dtempcntb \z@ \@dtempcnta \@ne
\fi
\fi}
\newcommand*{\f@x@l@cksR}{%
\ifcase\@lockR
\or
\global\@lockR \tw@
\or
\global\@lockR \z@
\fi
\ifcase\sub@lockR
\or
\global\sub@lockR \tw@
\or
\global\sub@lockR \z@
\fi
\ifcase\sub@lockR
\or
\global\sub@lockR \tw@
\or
\global\sub@lockR \z@
\fi
\or \or
\global\sub@lockR \z@
\fi

\let\linenumberlistR\empty%
\newcommand*{\affixline@numR}{% 
\ifledgroupnotesR\else\ifnumberline
\ifl@dskipnumber
\global\l@dskipnumberfalse
\else
\global\l@dskipnumberfalse
\fi}
\else 
\ifs@ublinesR
\@l@dtempcntb=\subline@numR
\@dcalcnum{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR}
\ch@cksub@l@ckR
\else 
\@l@dtempcntb=\line@numR
\if\linenumberlistR\empty%
\l@dcalcnum{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}
\else
\@l@dtempcntb=\line@numR
\edef\rem@inderR{\line@numR}
\edef\sc@n@list{\def\noexpand\rem@inderR{##1,\number\@l@dtempcntb,##2|\def\noexpand\rem@inderR{##2}}}
\sc@n@list\expandafter\sc@n@list\rem@inderR|
\ifx\rem@inderR\empty\advance\@l@dtempcntb\@ne\fi
\fi
\ch@ck@l@ckR
\fi
\ifnum\@l@dtempcntb=\@l@dtempcntb
\if\dskipversenumberR\else
\if@twocolumn
\if@firstcolumn
\gdef\l@dld@ta{\llap{{\leftlinenumR}}}
\iflinenumannotationothersideR
\gdef\l@drd@ta{\rlap{{\rightlinenumannotationR}}}
\else
\gdef\l@drd@ta{\rlap{{\rightlinenumR}}}
\iflinenumannotationothersideR
\gdef\l@dld@ta{\llap{{\leftlinenumannotationR}}}
\fi
\else
\ifboolexpr{bool {l@dprintingcolumns} and test \ifnumgreater{\line@margin@columnsR}{\m@ne}}
\@l@dtempcntb=\line@margin@columnsR
\@l@dtempcntb=\line@marginR
\fi
\fi
\fi
\fi
\fi
\global\l@dskipnumberfalse
\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.
VIII.6 Add insertions to the vertical list

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

\list@create{\inserts@listR}

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

\newcommand*{\add@insertsR}{%
VIII.7 Penalties

\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 \(-10000\).

\newcommand*{\add@penaltiesR}{\@l@dtempcnta=\ballast@count
  \ifnum\num@linesR>\@ne
    \global\advance\par@lineR \@one
    \ifnum\par@lineR=\@one
      \advance\@l@dtempcnta by \clubpenalty
    \fi
    \@l@dtempcntb=\par@lineR \advance\@l@dtempcntb \@one
    \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}{}

\%  

VIII.8 Printing leftover notes

\flush@notesR

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

\newcommand*{\flush@notesR}{%}
\iftoggle{notfirstrun@\jobname.extensionchars\section@numR R}{%\@xloop%}{}%  
\ifx\inserts@listR\empty \else%  
\@xloop\if\inserts@listR\empty \else%  
\gl@p\inserts@listR@to@\insertR%  
\@insertR%  
\global@set\insertR=\undefined%  
\repeat%}{%}
\%  
\%

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 \print@notesX@forpages, which replaces \print@Xnotes 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.

\footnote{See \url{http://tex.stackexchange.com/a/230332/7712}}
• To solve this problem, we change the magnification factor associated with notes before going to the next page. If we start a page where no notes are supposed to be printed, the magnification counter is set to 0. The dimension associated to footnote is set to \maxdimen, and so we can keep all the notes we want, without any break inside. We also set the note skip to 0pt. Before starting a new page where these notes are supposed to be printed, we reset these counter and skip to their default values. (About these counter, dimension and skip, read The TeXbook p. 122-125).

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

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

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

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

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

\ifcempty{Xonlyside@#1}{%
  \csuse{#1footstart}{#1}%
  \csuse{#1footgroup}{#1}%
}%
%
Second case: notes are for one side only. First test if we are in a page where they must be printed.

\ifboolexpr{%
  ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page}})
  or%
  (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}}))}%
%
If we are in a page where notes must be printed, print the notes.

{%
  \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@%
\global\skip\csuse{#1footins}=\z@%
\global\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.

```latex
\{%
\restore@Xnotes@settings{#1}%
}%
```

End of \print@Xnotes@forpages.

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

```latex
\newcommand{\restore@Xnotes@settings}[1]{%
\unless\ifnocritical@
\global\count\csuse{#1footins} = \csuse{default@#1footins}%
\global\skip\csuse{#1footins} = \csuse{Xbeforenotes@#1}%
\bgroup%
\csuse{Xnotefontsize@#1}%
\global\dimen\csuse{#1footins} = \csuse{Xmaxhnotes@#1}%
\egroup%
\global\setnamebox{#1footins@kept} = \box\namebox{#1footins}%
\fi%
}%
```

And now, the same for familiar footnotes.

```latex
\newcommand{\print@notesX@forpages}[1]{%
\ifcsempty{onlysideX@#1}{%
\csuse{footstart#1}{#1}%
\csuse{footgroup#1}{#1}%
}{%
\ifboolexpr{%
((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page}})%
or%
(test {\ifcsstring{onlysideX@#1}{R}} and test{\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}[1]{% 
  \unless\ifnofamiliar@ % 
  \global\count\csuse{footins#1}=\csuse{default@footins#1}% 
  \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}% 
  \bgroup% 
  \csuse{Xnotefontsize@#1}% 
  \global\dimen\csuse{footins#1}=\csuse{maxhnotesX@#1}% 
  \egroup% 
  \global\setnamebox{footins#1@kept}=\box\namebox{footins#1}% 
  \fi %
}}} %

\insert@notes@for@onlyside %insert@notes@for@onlyside is everytime \Pages go to the next side. It just reinsert the notes note printed on the previous side because of Xonlyside or \onlysideX setting.

\def\do##1{% 
  \unless\ifnocritical@% 
  \ifvoid\csuse{##1footins@kept}\else% 
    \expandafter\insert\csname ##1footins\endcsname% 
  \bgroup% 
  \unvnamebox{##1footins@kept}% 
  \egroup% 
  \fi% 
  \restore@notesX@settings{##1}% 
} %
\dolistloop{@series}%

IX.2 Critical footnote printed in right side

\edtext@later and \edtext@now 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 \edtext@later and the other for \edtext@now.
IX.2 Critical footnote printed in right side

\newcount\edtext@now%\newcount\edtext@later%
\newcommand{\edtextlater}[2]{% #1 lemma, #2 critical notes
    \global\advance\edtext@later by \@ne%
    \global\advance@\edtext@level by \@ne%
    % As we are in a pseudo-\cs{edtext}, we now need to get the \cs{sameword} data stored in the auxiliary file for this \cs{edtextlater}.
    \ifcsvoid{sw@list@edtext@\the\edtext@level}{}
    \expandafter\gl@p\csname sw@list@edtext@\the\edtext@level\endcsname\to\sw@inthisedtext
\endgraf}

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

\csxdef{edtext@later@\the\edtext@later}{%
    \edtextnow will insert a empty \edtext.
    \noexpand\edtext{%
    \}%
    \}%
    \}%

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

\noexpand\xxref{start:edtext:later:{\the\edtext@later}}{end:edtext:later:{\the\edtext@later}}%
\noexpand\linenum{|||\edfont@info}%
\unexpanded{\def\sw@inthisedtext}{\expandonce{\sw@inthisedtext}}%
\endgraf}

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.

\ifledRcol%
\unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@true}}%
\}%
\noexpand\setcounter{pstartR}{\noexpand\xpstartref{start:edtext:\n\text{later}:\the\edtext@later}}% 
\else% \unexp\mathtt{appto@beforeinsertofthisedtext{\ledRcol@false}}% \unexp\mathtt{appto@beforeinsertofthisedtext}% \% \noexpand\setcounter{pstartL}{\noexpand\xpstartref{start:edtext:\n\text{later}:\the\edtext@later}}% 
\fi% 
\fi% 
\fi% 
And the footnote command of this \textit{\texttt{\textbackslash edtext later}}.

\unexp\mathtt{#2}% 
\unexp\mathtt{}}% 
\unexp\mathtt{}}% 
\unexp\mathtt{%}

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 \textit{\texttt{\textbackslash edtext\texttt{\textbackslash later}}} and, if required, about the use of a \textit{\texttt{\textbackslash lemma}}

\edlabel{start:edtext:later:\the\edtext@later}% \flag@start@later% 
\bgroup% \def\lemma##1{% \if\\additional\\color{\red}{\\text{\texttt{\textbackslash lemma}}}% \\texttt{\textbackslash \texttt{\textbackslash write\texttt{\textbackslash linenum\texttt{\textbackslash out\texttt{\textbackslash string\texttt{\textbackslash \texttt{\textbackslash \texttt{\textbackslash lemma}}}}}}}} \else% \\texttt{\textbackslash \texttt{\textbackslash write\texttt{\textbackslash linenum\texttt{\textbackslash out\texttt{\textbackslash string\texttt{\textbackslash \texttt{\textbackslash \texttt{\textbackslash lemma}}}}}}}} \fi% 
}%
\renewcommand\dolistloop{\@series}#2%
\egroup%
\showlemma{#1}%
\edlabel{end:edtext:later:\the\edtext@later}% 
\flag@end@later% 
%
We decrease the counter increased at the beginning.

\advance\@edtext@level by -1

\edtextnow just calls the command defined as is, reading the \textit{\texttt{\textbackslash edtext\texttt{\textbackslash later}}} list.
\newcommand{\edtextnow}[0]{\%\global\advance\edtext@now by \@ne\csuse{edtext@later\the\edtext@now}\%}\%\%

### X Cross referencing

\labelref@list

Set up a new list, \labelref@list, to hold the page, line and sub-line numbers for each label in right text.

\list@create{\labelref@list}

\edlabel This command is defined only one time in reledmac, including features for reledpar.

\@lab

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

It is defined on reledmac.

#### XI Sidenotes

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

```latex
\newdimen\morespace@leftnote@rightcolumn
\newdimen\morespace@rightnote@leftcolumn

\newcommand{\get@sidenote@morespace@columns}{%
  \if@sidenotesmarginpage%
    \morespace@leftnote@rightcolumn=%
    \morespace@rightnote@leftcolumn=%.%
\global\morespace@leftnote@rightcolumn=%
\morespace@rightnote@leftcolumn=%

\newcommand{\get@sidenote@morespace@columns}{%}
\if@sidenotesmarginpage%

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

\newdimen\morespace@leftnote@rightcolumn
\newdimen\morespace@rightnote@leftcolumn

\newcommand{\get@sidenote@morespace@columns}{%}
\if@sidenotesmarginpage%

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

\global\morespace@leftnote@rightcolumn=%
\morespace@rightnote@leftcolumn=%

% \if@sidenotesmarginpage %

\global\morespace@leftnote@rightcolumn=%
\morespace@rightnote@leftcolumn=%
% \if@sidenotesmarginpage %

If `\if@sidenotesmarginpage` is False
\texttt{\textbackslash{affixside@note}} \texttt{The right text version of \textbackslash{affixside@note}.}

\texttt{\textbackslash{newcommand}*{\textbackslash{affixside@note}R}{\%}
\texttt{\prepare@edindex@fornote{\the\page@numR}|\the\line@numR|\the\subline@numR}|\the\page@numR|\the\line@numR|\the\subline@numR|}}
\texttt{\def\sidenotecontent@{}%}
\texttt{\numgdef{\itemcount@}{0}%%}
\texttt{\def\do##1{%}
\texttt{\ifnumequal{\itemcount@}{0}{%}
\texttt{\appto\sidenotecontent@{##1}}%% Not print not separator before the 1st note}
\texttt{\appto\sidenotecontent@{\这些notesep #1}%%}
\texttt{\numgdef{\itemcount@}{\itemcount@+1}%%}
\texttt{\dolistloop{\l@dcsnotetext}%%}
\texttt{\ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%}
\texttt{\ifx\@templ@d\@templ@n \else%}
\texttt{\if@twocolumn%}
\texttt{\if@firstcolumn%}
\texttt{\setl@dlp@rbox{##1}{\sidenotecontent@}%%}
\texttt{\else%}
\texttt{\setl@drp@rbox{\sidenotecontent@}%%}
\texttt{\fi%}
\texttt{\else%}
\texttt{\@l@dtempcntb=\sidenote@marginR%}
\texttt{\ifnum\@l@dtempcntb>\@ne%}
\texttt{\ifnumodd{\@l@dtempcntb}%%}
\texttt{\setl@drp@rbox{\sidenotecontent@}%%}
\texttt{\else%}
\texttt{\setl@dlp@rbox{\sidenotecontent@}%%}
\texttt{\fi%}
\texttt{\fi%}
\texttt{\@l@dtempcntb=\sidenote@marginR%}
\texttt{\ifnum\@l@dtempcntb>\@one%}
\texttt{\ifnumodd{\@l@dtempcntb}%%}
\texttt{\setl@dlp@rbox{\sidenotecontent@}%%}
\texttt{\else%}
\texttt{\setl@drp@rbox{\sidenotecontent@}%%}
\texttt{\fi%}
\texttt{\fi%}
\texttt{\ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%}
XII Verse

Like in \reledmac, the insertion of hangsymbol 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.

\newif\ifinserthangingsymbolR
\newcommand{\inserthangingsymbolL}{%\ifinserthangingsymbolL%\ifinastanzaL%\@hangingsymbol%\fi%\ifinastanzaL%\@hangingsymbol%\fi%\fi%}%\newcommand{\inserthangingsymbolR}{%\ifinserthangingsymbolR%\ifinastanzaR%\@hangingsymbol%\fi%\ifinastanzaR%\@hangingsymbol%\fi%\fi%}%

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

\newenvironment{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@}=\z@% \let\stanza@hang\relax \let\endlock\relax \else \rightskip\z@ plus 1fil\relax \fi 
\ifnum\usenamecount{szp@0@}=\z@% \let\sza@penalty\relax \fi 
\def&{% \endlock\mbox{}% \sza@penalty \global\advance\stanza@count\@ne \@astanza@line} 
\def\&{\@stopastanza} 
\ifboolexpr{\not\ifdefvoid{\at@every@stanza} \and\ifstrempty{#1} \and\ifstrempty{#2}}% \pstart[#1][#2]\at@start@every@stanza} \@astanza@line 
\let\par\relax\ignorespaces%No paragraph in verses 
\% \global\undef\parindent@beforestanza% \@insertstanzanumber% 
\% 
\@stopastanza This command is called by \& in astanza environment. It allows optional arguments.
XIII Fixing babel and polyglossia

With parallel texts there is the possibility that the two sides might use different languages via babel. On the other hand, nor babel nor polyglossia might not be called at all (even though it might be already built into the format).
With the normal sequential text each line is initially typeset in the current language environment, and then it is output at which time its attachments are typeset (in the same language environment. In the parallel case lines are typeset in their current language but an attachment might be typeset outside the language environment of its line if the left and right side languages are different. To counter this, we have to make sure that the correct language is used at the proper times.

\if\dusedbabel\fi\if\@dusedbabelfalse\fi\label{dusedbabel}
\if\@dusedbabelfalse\fi\label{dusedbabeltrue}

A flag for checking if babel has been used as a package.

\@checklang
\@bbl@set@language

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

\patchcmd{\bbl@set@language}{\select@language{\languagename}}{\edef\languagename{#1}\select@language{\languagename}}{}{}%

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

\selectlanguage\@duselanguage
\@theledlanguageL
\@theledlanguageR
\selectlanguage

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

\newcommand*{\@duselanguage}{[1]}
\gdef\@theledlanguageL{}
\gdef\@theledlanguageR{}
%

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

\AtBeginDocument{%
\@ifundefined{xpg@main@language}{%
\@ifundefined{bbl@main@language}{%
Either babel has not been used or it has been used with no specified language.
Here we deal with the case where babel has been used. `\selectlanguage` has to be redefined to use our version of `\bbl@set@language` and to store the left or right language.

\begin{verbatim}
\@usedbabelfalse
\{%
% #1\selectlanguage
\end{verbatim}

\begin{verbatim}
\@usedbabeltrue
\let\@oldselectlanguage\selectlanguage
\let\@oldbbl@set@language\bbl@set@language
\renewcommand{\selectlanguage}[1]{
\@oldselectlanguage{#1}%
\ifledRcol \gdef\theledlanguageR{#1}%
\else \gdef\theledlanguageL{#1}%
\fi}
\end{verbatim}

\begin{verbatim}
\@uselanguage simply calls the original \selectlanguage so that \theledlanguageL and \theledlanguageR are unaltered.
\end{verbatim}

\begin{verbatim}
\renewcommand*{\@uselanguage}[1]{
\expandafter\@oldselectlanguage\expandafter{#1}}
\end{verbatim}

\begin{verbatim}
Lastly, initialise the left and right languages to the current babel one.
\end{verbatim}

\begin{verbatim}
\gdef\theledlanguageL{\bbl@main@language}
\gdef\theledlanguageR{\bbl@main@language}
\end{verbatim}

\begin{verbatim}
If use polyglossia
\end{verbatim}

\begin{verbatim}
{%
@ifpackagelater{polyglossia}{2020/04/08}{\@errpolyglossiaTooOld}
%
\let\old@otherlanguage\otherlanguage%
\renewcommand{\otherlanguage}[2][]{%
\xpg@set@group@aux%
\selectlanguage[#1]{#2}%
\ifledRcol \gdef\theledlanguageR{#2}%
\else \gdef\theledlanguageL{#2}%
\fi}%
\renewcommand{\@uselanguage}[1]{%
\csuse{no\languagename@numbers}\
@select@language[#1]%
}%
\gdef\theledlanguageL{\xpg@main@language}%
\gdef\theledlanguageR{\xpg@main@language}%
%
%
That is it.
\end{verbatim}
XIV Counts and boxes for parallel texts

In sequential text, each chunk (that enclosed by \pstart ... \pend) is put into a box called \rawtext 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@dnumpstartsL
\newcount\l@dnumpstartsR

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

\newcount\l@dpscL
\newcount\l@dpscR

A couple of scratch counts to count left and right \pstart, respectively.

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

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

\newcommand*{\l@setupmaxlinecounts}{%}

To be able to synchronise left and right texts we need to know the maximum number of text lines there are in each pair of chunks. \l@setupmaxlinecounts creates \maxchunks new counts called \l@maxlinesinpar1, etc., and \l@zeromaxlinecounts zeroes all of them.
% 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@}
%

XV  Checking text to be processed

\if@pstarts
\newif@ifopstarts
\newcommand*{\check@pstarts}{%\if@opstarts
\newif@ifopstarts
\newcommand*{\check@pstarts}{%\ifnum\l@dnumpstartsL>\l@dpscL
\l@opstartstrue\else\l@opstartsfalse\fi\fi}
}\l@opstartstrue
\else
\ifnum\l@dnumpstartsR>\l@dpscR
\l@opstartstrue
\l@opstartsfalse\fi\fi
}
%
\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}{%
\ifaraw@textfalse
\ifvbox\namebox{l@dLcolrawbox\the\l@dpscL} \araw@texttrue
\else
\ifvbox\namebox{l@dRcolrawbox\the\l@dpscR} \araw@texttrue
\fi
\fi
}

\@writelinesinparL \@writelinesinparR
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@outR{\string@pendR\the\@donereallinesR}}% 
\next% 
\global\@donereallinesR \z@}

\@writepageofparL \@writepageofparR
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}}% 
}
XVI Parallel columns

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

\newsavebox\@eledsectionL
\newsavebox\@eledsectionR

\Columns The \Columns command results in the previous Left and Right texts being typeset in matching columns. There should be equal numbers of chunks in the left and right texts.

\newcommand*{\Columns}{%
  \iflsdpairing%
    \led@err@Columns@InsideEnv%
  \fi%
  \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{l@dLcolrawbox1}}}{%}
    \ifcsboxvoid{l@dRcolrawbox1}{%
      \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%
Check if there are chunks to be processed, and process them two by two (left and right pairs).

\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

\checkraw@text
\{\loop\ifraw@text
% Grab the next pair of left and right text lines and output them, swapping languages if they differ, adding section title if needed.
\l@duselanguage{\theledlanguageL}%
\do@lineL
\xifinlist{the\pstarts@typeset@L}{\eledsections@@}%
\{%\n\ifdefstring{\eledsectmark}{L}%
\{"csuse{eledsectmark\the\pstarts@typeset@L}%
\}%\n\global\csundef{eledsectmark\the\pstarts@typeset@L}%
\savebox{\eledsectionL}{\parbox[t][t]{\Lcolwidth}{\vbox{}
\}\print@eledsectionL}}%\vbox{--> prevent alignment troubles with RTL language
\}%\%
\}%\%
\l@duselanguage{\theledlanguageR}%
\do@lineR
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.
\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).

\get@intercolumns@width \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.
\newdimen\intercolumns@width
\newcommand\get@intercolumns@width{%
  \begin{macrocode}
  \global\intercolumns@width=\z@%
  \ifdefstring{\columns@position}{C}{%
    First case, the columns are centered.
    {%
      First sub-case, the width between columns is automatically calculated.
      \ifboolexpr{test {\ifdimless{\beforecolumnseparator}{\z@}}
        and test {\ifdimless{\aftercolumnseparator}{\z@}}}%
          \global\intercolumns@width=\dimexpr\columnrulewidth +%
          (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth)*1/2%
          \relax%
        \}%
    %
    Second sub-case, the width between column is fully determined by user setting.
    \ifboolexpr{test {\ifdimgreater{\beforecolumnseparator}{\z@}}
        and test {\ifdimgreater{\aftercolumnseparator}{\z@}}}%
      \global\intercolumns@width=\dimexpr\columnrulewidth + \beforecolumnseparator + \aftercolumnseparator
      \relax%
    }%
    %
    Third sub-case, the width before column separator is determined by user setting, but the
    width after column separator is automatically calculated.
    \ifboolexpr{test {\ifdimgreater{\beforecolumnseparator}{\z@}}
        and test {\ifdimless{\aftercolumnseparator}{\z@}}}%
      \global\intercolumns@width=\dimexpr\columnrulewidth + \beforecolumnseparator +
    }%
  \end{macrocode}
}
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 + \aftercolumnseparator + (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \aftercolumnseparator) * 1/3\relax\}
\else\}
  \%
\fi

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 {\ifdimless{\aftercolumnseparator}{\z@}}}\
  \global\intercolumns@width=\dimexpr\columnrulewidth + (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth)*2/3\relax\}
\else\}
  \%
\fi

The total of the width before and after column separator
\relax\%
\%

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

\ifboolexpr{\test {\ifdimgreater{\beforecolumnseparator}{\z@}}}
XVI Parallel columns

and test \ifdimgreater{\aftercolumnseparator}{0}\%
\%
\%
\global\intercolumns@width=\dimexpr\columnrulewidth + \beforecolumnseparator + \aftercolumnseparator \relax\%
\%
\%

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

\ifboolexpr{%
\test {\ifdimgreater{\beforecolumnseparator}{0}}% and \test {\ifdimless{\aftercolumnseparator}{0}}%
\}%,
\%
\global\intercolumns@width=\dimexpr\columnrulewidth + \beforecolumnseparator +
(\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \beforecolumnseparator) * 1/2\%
\relax%
\}
\{%
%

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}{0}}% and \test {\ifdimgreater{\aftercolumnseparator}{0}}%
\}%,
\%
\global\intercolumns@width=\dimexpr\columnrulewidth + \aftercolumnseparator +
(\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \aftercolumnseparator) * 1/2\%
\relax%
\}
\{%
%

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

\}%
\%

End of \get@intercolumns@width.
\printleftcolumn and \printrightcolumn print the line number of the left or right column respectively. Normally, it is defined by the content of the \texttt{Leftside} or \texttt{Rightside} environments. But if \texttt{movecolumnspositiononrightpage} is set to \texttt{TRUE}, it also depends on the page number.

\newcommand{\printleftcolumn}{\%
  \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@leftbox%
  \ifhbox\@eledsectionL%
    \usebox\@eledsectionL%
  \fi%
  \%
  \unhbox\@l@rightbox%
  \ifhbox\@eledsectionR%
    \usebox\@eledsectionR%
  \fi%
  \%
\}%

\newcommand{\printrightcolumn}{\%
  \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\@r@rightbox%
  \ifhbox\@eledsectionR%
    \usebox\@eledsectionR%
  \fi%
  \%
  \unhbox\@r@leftbox%
  \ifhbox\@eledsectionL%
    \usebox\@eledsectionL%
\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}{}\n    \xifinlist{\next@abslineR}{\l@prev@pbR}{\@pbtrue}{}\n    \xifinlist{\next@absline}{\l@prev@nopb}{\@nopbtrue}{}\n    \xifinlist{\next@abslineR}{\l@prev@nopbR}{\@nopbtrue}{}\n  }{}
  \IfStrEq{\led@pb@setting}{after}{
    \xifinlist{\the\absline@num}{\l@prev@pb}{\@pbtrue}{}\n    \xifinlist{\the\absline@numR}{\l@prev@pbR}{\@pbtrue}{}\n    \xifinlist{\the\absline@num}{\l@prev@nopb}{\@nopbtrue}{}\n    \xifinlist{\the\absline@numR}{\l@prev@nopbR}{\@nopbtrue}{}\n  }{}
  \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
  \if@pb\pagebreak[4]\fi
}%

\column_separator \column_rulewidth

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 \column_rulewidth (initially 0pt so the rule is invisible).

\newcommand*{\column_separator}{\smash{\rule[-0.2\baselineskip]{\column_rulewidth}{1.05\baselineskip}}}\newdimen\column_rulewidth\column_rulewidth=\z@\%}

\columns_position \columns

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

\newcommand*{\columns}{[1]{}\xdef{\columns@position}{[1]}\%\xdef{\columns@position}{R}\%\%}
\beforecolumnseparator and \aftercolumnseparator lengths are defined to -1pt. If user changes them to a positive length, the lengths are used to define blank spaces before / after the column separator, instead of \hfill.

\newlength{\beforecolumnseparator} \setlength{\beforecolumnseparator}{-2pt} \newlength{\aftercolumnseparator} \setlength{\aftercolumnseparator}{-2pt}

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

\newcommand{\setwidthliketwocolumns}{% \get@intercolumns@width\hsize=\dimexpr \Lcolwidth + \intercolumns@width + \Rcolwidth \relax}\newcommand{\setpositionliketwocolumns@L}{% \renewcommand{\ledrlfill}{\hfill}}\newcommand{\setnotespositionliketwocolumns@L}{% \newdimen\temp% \temp=\hsize% \tempa=\Lcolwidth% \advance\tempa\Rcolwidth% \advance\temp-\tempa% \divide\temp by 2% \leftskip=\temp% \rightskip=-\temp%}
The \Columns@print@before@pstart and \Columns@print@after@pend print the content of the optional argument of \texttt{\pstart / \pend}. If this content is not empty, it also print the separator.
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.

\begin{verbatim}
\newcount\numpagelinesL
\newcount\numpagelinesR
\newcount\l@dminpagelines
\end{verbatim}

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.

\begin{verbatim}
\newcommand*{\Pages}[1][]{\usedefault}{% 
  \ifl@dpairing% \led@err@Pages@InsideEnv% \fi% \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{l@dLcolrawbox1}}}{}% \fi% \ifcsboxvoid{l@dRcolrawbox1}% \ifcsboxvoid{l@dLcolrawbox1}%
\end{verbatim}
Get onto an empty even (left) page, then initialise counters, etc.

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

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

Check if there are chunks to be processed.
Loop over the number of chunks, incrementing the chunk counts (\l@dpscL and \l@dpscR are chunk (box) counts).

\global\advance\l@dpscL \one% \global\advance\l@dpscR \one%

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

\getlinesfromparlistL% \getlinesfromparlistR% \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}{\usenamecount{l@dmaxlinesinpar\the\l@dpscL}}% \check@pstarts% \repeat%

Zero the counts again, ready for the next bit.

\global\l@dpscL=\z@% \global\l@dpscR=\z@

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

\getlinesfrompagelistL% \getlinesfrompagelistR% \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}{\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.

\check@pstarts% \if@pstarts%

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

\global\advance\l@dpscL \one% \global\advance\l@dpscR \one% \global\advance\pstarts@typeset@L \one% \restore@pstartL@pc% \restore@pstartR@pc%

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

\ifshiftedpstarts
  \ifdim\ht\l@dleftbox>0pt
    \parledgroup@correction@notespacing{L}
    \hb@xt0.0\hsize{\ledstrutL\unhbox\l@dleftbox}
  \else\fi
  \xifinlist{\the\pstarts@typeset@L}{\eled@sections@@}
  \{
    \add@apparatusL
  \}
\fi
\unless\ifadvancedshiftedpstarts\% 
\dimen0=\pagetotal\%
\advance\dimen0 by \baselineskip\%
\global\pagetotal=\dimen0\%
\else\%
\ifnomaxlines\%
\numdef{\@tmp}\{\the\l@dpscL+1}\%
\ifsdef{minpage@pstart@\@tmp}{\%
\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}{%
\dimen0=\pagetotal\%
\advance\dimen0 by \baselineskip\%
\global\pagetotal=\dimen0\%
}\}%
}\%
\}%
\fi\%
\fi\%
\else\%
\parledgroup@correction@notespacing{L}
\hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
\fi\%
\fi\%
\else\%
\parledgroup@correction@notespacing{L}
\hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
\fi\%
\fi\%

Perhaps we have to move to the next (left) box. Check if we have got all we can onto the page. If not, repeat for the next line. Check if we have to print the optional argument of the last pend. Check if the page is full. Check if the verse is split in two subsequent pages. Check there is any forced page breaks. Reset the verse skipnumber boolean

\get@nextboxL\%
\global\l@dskipversenumberfalse\%
\ifprint@last@after@pendL\%
\csuse{after@pendL\the\l@dpscL}\%
\global\csundef{after@pendL\the\l@dpscL}\%
\fi\%
\checkpageL\%
\checkverseL\%
\checkpbL\%
\repeat\%

That (left) page has been filled. Output the number of real lines on the page — if the page break is because the page has been filled with lines, use the actual number, otherwise the page has been ended early in order to synchronise with the facing page so use an impossibly large number.

\if\l@dpagefull\%
\@writelinesonpageL{\the\numpagelinesL}\%
\else\%
\@writelinesonpageL{1000}\%

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.

\numpagelines L \z@%
\parledgroup@correction@notespacing@init%
\clear@dleftpage }

Now do the same for the right text.

\checkpage R%
\Rightpagehook%
\l@dusetlanguage{\theledlanguage R}%
\{ 
\loop\ifl@dsamepage%
\set@sectcount R%
\ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{%}
\csuse{before@pstart R\the\l@dpsc R}%
\global\csundef{before@pstart R\the\l@dpsc R}%
\do@line R%
\xifinlist{\the\l@dpsc R}{\eled@sections R@@}{%}
{\print@eledsection R}%
{%}
\advance\numpagelines R \@ne%
\ifshiftedpstarts%
\ifdim\ht\l@drightbox > 0 pt%
\parledgroup@correction@notespacing{R}%
\hb@xt@ \hsize{\ledstrut\unhbox\l@drightbox}%
\else%
\xifinlist{\the\l@dpsc R}{\eled@sections R@@}{%}
{\add@apparatus R}%
{%}
\unless\ifadvancedshiftedpstarts%
\dimen0 = \pagetotal%
\advance\dimen0 by \baselineskip%
\global\pagetotal = \dimen0%
\else%
\ifnomaxlines%
\numdef{\@tmp}{\the\l@dpsc R + 1}%
\ifsdef{\minpage@pstart 0 \@tmp}{}
\ifnumless{\the\c@page}{\csuse{minpage@pstart 0 \@tmp}}%
{\dimen0 = \pagetotal%}
\advance\dimen0 by \baselineskip%
\global\pagetotal = \dimen0%
{%}
{%}
{%}
The page is full, so move onto the next (left, odd) page and repeat left text processing.

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.

We have now output the text from all the chunks.

Make sure that there are no inserts hanging around.
Zero counts ready for the next set of left/right text chunks. The boolean tests for stanza are switched to false.

\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 \edtext@later and \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}}%

\finish@Pages@notes

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 \Pages typesetting, and that there are no more long notes left for the next pages.

\newcommand{\finish@Pages@notes}{% 
  \def\do##1{% 
    \ifnocritical@% 
      \global\newnamebox{##1footins}% 
    \fi
    \ifnofamiliar@% 
      \global\newnamebox{footins##1}% 
    \fi
  }% 
}

First, declare footnote box if there was no previous declared. E.g. if familiar or critical notes were disabled by \reledmac’s options.
We must also restore the settings for the footnotes in case of using onlyside option.

\restore@Xnotes@settings{##1}\%
\restore@notesX@settings{##1}\%

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

\ifvoid\csuse{##1footins}\%
\ifvoid\csuse{footins##1}\else\%
\newpage\null\%
\listbreak\%
\fi\%
\else\%
\newpage\null\%
\listbreak\%
\fi\%
\dolistloop{\@series}\%
\%

\XVII.4 Struts
\ledstrutL
\ledstrutR

Struts inserted into left and right text lines.

\newcommand*{\ledstrutL}{}
\newcommand*{\ledstrutR}{}

\%

\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.

\ifodd\c@page\hbox{}#1\clearpage\fi\%
\else\%
\clearpage\%
\fi\%
\ifdim\pagetotal<\topskip% on an empty page
\else\%
\clearpage\%
\Pages@mainmatter\%
\fi\%
\fiodd\c@page%
\ifprevpgnotnumbered%
\addtocounter{par@page}{-1}%
\clearl@dleftpage and \clearl@drightpage get us onto an odd and even page, respectively, checking that we end up on the subsequent page. Both commands use \newpage and not \clearpage. Because \clearpage prints all footnotes before the next page, even if it has to add new empty pages, while \newpage does not. And as we want notes started in the left page continue in the right page and vice-versa, we must use \newpage and not \clearpage.

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

\newcommand*{\getlinesfromparlistL}{%}
\ifx\linesinpar@listL\empty%
\gdef\@cs@linesinparL{0}%
\else
\gl@p\linesinpar@listL\to\@cs@linesinparL
\fi%
\newcommand*{\getlinesfromparlistR}{%}

\section{Lines managing}
\getlinesfromparlistL gets the next entry from the \linesinpar@listL and puts it into \@cs@linesinparL; if the list is empty, it sets \@cs@linesinparL to 0. Similarly for \getlinesfromparlistR.
XVII.6 Lines managing

\ifx\linesinpar@listR\empty
  \gdef\@cs@linesinparR{0}\%\else
  \gl@p\linesinpar@listR\to\@cs@linesinparR\fi

\getlinesfrompagelistL\getlinesfrompagelistR
This macro 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}
XVII.7 Page break managing

The checkpageL tests if the space and lines already taken on the page by text and footnotes is less than the constraints. If so, then \ifl@dsamepage is set FALSE and \ifl@dsamepage is set TRUE. If the page is spatially full then \ifl@dpagefull is set TRUE and \ifl@dpagefull is set FALSE. If it is not spatially full but the maximum number of lines have been output then both \ifl@dpagefull and \ifl@dpagefull are set FALSE.
\check@goal
\ifdim\pagetotal<\ledthegoal
  \ifnum\numpagelinesR<\l@dminpagelines
  \else
    \ifnomaxlines%
      \l@dsamepagefalse%
      \l@dpagewhitefalse%
    \else
      \fi
    \fi%
    \fi
  \else
    \l@dsamepagefalse
    \l@dpagewhitetrue
  \fi%
\else
  \l@dsamepagefalse
  \l@dpagewhitefalse
\fi%
\ifprint@last@after@pendR%
  \l@dpagewhitefalse%
  \l@dsamepagefalse%
  \print@last@after@pendRfalse%
\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@dpagewhitefalse\l@dsamepagefalse}{}
    \xifinlist{\the\absline@num}{\l@prev@nopb}{\l@dpagewhitetrue\l@dsamepagetrue}{}
  }{}
  \IfStrEq{\led@pb@setting}{before}{
    \numdef{\next@absline}{\the\absline@num+1}
    \xifinlist{\next@absline}{\l@prev@nopb}{\l@dpagewhitefalse\l@dsamepagefalse}{}
    \xifinlist{\next@absline}{\l@prev@nopb}{\l@dpagewhitefalse\l@dsamepagetrue}{}
  }{}
}
\newcommand{\checkpbR}{
  \IfStrEq{\led@pb@setting}{after}{
    \xifinlist{\the\absline@num}{\l@prev@pbR}{\l@dpagewhitefalse\l@dsamepagefalse}{}
    \xifinlist{\the\absline@num}{\l@prev@nopbR}{\l@dpagewhitefalse\l@dsamepagetrue}{}
  }{}
  \IfStrEq{\led@pb@setting}{before}{
    \xifinlist{\the\absline@num}{\l@prev@pbR}{\l@dpagewhitefalse\l@dsamepagefalse}{}
    \xifinlist{\the\absline@num}{\l@prev@nopbR}{\l@dpagewhitefalse\l@dsamepagetrue}{}
  }{}
}
\checkverseL and \checkverseR are called after each line is printed. They prevent page break inside line of verse.

\checkverseL
\checkverseR

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

The current box is not empty; do nothing.

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

But not when sufficient lines for this page have been generated (except when we don’t do any synchronization whatsoever). output.
If we use the nomaxlines option, we will start at new page, but we take count of the lines to be typeset for the actual right chunk on the right page, before starting new chunk on the left page.

Sufficient lines have been output.
XVII.8  Getting boxes content

\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.

\writtenlinesLfalse
\ifbypstart@%
\global\line@num=0%
\resetprevline@%
\fi
%
% Add the content of the optional argument of the previous \protect\cs{pend }.
% \begin{macrocode}
\csuse{after@pendL@	he\l@dpscL}%
\global\csundef{after@pendL@	he\l@dpscL}%
%
Check the number of lines

\l@dcalc@maxoftwo{\the\usenamecount{l@dmaxlinesinpar\the\l@dpscL}}%
{\the\@donetotallinesL}
{\usenamecount{l@dmaxlinesinpar\the\l@dpscL}}%
\global\@donetotallinesL \z@
%
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@notespacing@final{L}
\else
%
\print@last@after@pendLtrue%
\fi
\fi
XVIII  The mainmatter option of \Pages

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

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

\chapter% \pretocmd{\chapter}{% 
% \ifl@dprintingpages% 
% \vbox{}% 
% }% 
% {}% 
% 
% \eledsectnotoc \eledsectnotoc just saves its content \@eledsectnotoc, which will be tested where sectioning commands will be printed.

% \newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}} 
% \eledsectnotoc{R} 
% 
% \eledsectmark \eledsectmark just saves its content \@eledsectmark, which will be tested where sectioning commands will be printed.

% \newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}} 
% \eledsectmark{L} 
% 
% \eledsection@correcting@skip Because the vertical correction needed after inserting a title in parallel depends whether we are in parallel columns or parallel pages, we stock its length in \eledsection@correcting@skip.

% \newskip\eledsection@correcting@skip 
% 
% \eled@sectioningR@out We save the sectioning commands of the right side in the \eled@sectioningR@out file.

% \newwrite\eled@sectioningR@out 
% 
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 \l@prev@pbR macro is a etoolbox’s list, which contains the lines in which page breaks occur (before or after). The \l@prev@nopbR macro is a etoolbox list, which contains the lines in which NO page breaks occur (before or after).

\def\l@prev@pbR{}
\def\l@prev@nopbR{}

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

\newcommand{\l@ledpbR}{\write\linenum@outR{\string\l@led@pbR}}
\newcommand{\l@ledpbnumR}[1]{\write\linenum@outR{\string\l@led@pbnumR{#1}}}
\newcommand{\l@lednopbR}{\write\linenum@outR{\string\l@led@nopbR}}
\newcommand{\l@lednopbnumR}[1]{\write\linenum@outR{\string\l@led@nopbnumR{#1}}}

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

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

XXI Parallel ledgroup

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

\newmarks{\l@parledgroup@}
\newmarks{\l@parledgroup@series}
\newmarks{\l@parledgroup@type}

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

\newcommand{\l@parledgroup@notes@startL}{}
\ifnum\usenamecount{\l@dmaxlinesinpar}\the\l@dpscL>0\%\fi\IfStrEq{\splitfirstmarks}\l@parledgroup@type{\footnoteX}{\csuse{bhooknoteX}\splitfirstmarks}\l@parledgroup@series{}}{}%
\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.

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

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

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

\dimdef{\parledgroup@notespacing@correction}{0pt}
\newcommand{\parledgroup@notespacing@set@correction}{\begin{tabular}{@{}c@{}}\@getfirstseries\csuse{Xnotefontsize@\@firstseries}\end{tabular}\@parledgroupnotespacing\dimgdef{\temp@spacing}{\baselineskip}}\dimgdef{\parledgroup@notespacing@correction}{\baselineskip-\temp@spacing}

\parledgroup@correction@notespacing@init sets the value of accumulated corrections of note spacing to 0 pt. It is called at the beginning of each pages AND at the end of each ledgroup.

\newcommand{\parledgroup@correction@notespacing@init}{\dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}\dimdef{\parledgroup@notespacing@correction@modulo}{0pt}}

\parledgroup@correction@notespacing@final adds the total space deleted because of correction for notes, in a parallel ledgroup. It also adds the space needed by the other side spaces between note rules and notes. It is called after the print of each pstart/pend.

\newcommand{\parledgroup@correction@notespacing@final}[1]{\ifparlegroup\vspace{\parledgroup@notespacing@correction@accumulated}\parledgroup@correction@notespacing@init\fi\ifstrequal{#1}{L}{\numdef{\@checking}{\the\l@dpscL-1}}\else{\numdef{\@checking}{\the\l@dpscR-1}}\fi\numdef{\@checking}{\the\l@dpscR-1}}
\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.
The macro \texttt{parledgroup@beforenotes@save} dumps the space before notes of the current parallel ledgroup in a macro named with the current pstart number.

\newcommand{\parledgroup@beforenotes@save}[1]{\ifparledgroup\csdimdef{}{\texttt{parledgroup@beforenotes#1}}\csdimdef{}{\texttt{parledgroup@beforenotes#1}}\fi}

\section{XXII Compatibility with \texttt{eledmac}}

Here, we define some command for the \texttt{eledmac-compat} option.

\ifeledmaccompat\fi

\section{XXIII The End}

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

A.1 Migration to \eledpar 1.4.3

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

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

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

See the following two examples:

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

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

A.2 Migration from \eledpar to \reledpar

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

A.2.1 Deprecated options

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

A.2.2 \renewcommand replaced with command

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

\begin{verbatim}
Deprecated \renewcommand   Replaced with
\goalfraction             \setgoalfraction
\parledgroupnotespacing  \setparledgroupnotespacing
\Rlineflag                \setRlineflag
\end{verbatim}
A.2.3 Commands the names of which have changed

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

<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 reledpar 2.2.0

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

A.4 Migration to reledpar 2.3.0

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

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

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

A.5 Migration to reledpar 2.4.0

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

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

A.6 Migration to reledpar 2.5.0

If you use either \stanza or a stanza environment, please read A.12 p. 411.

A.7 Migration to reledpar 2.6.0

\printlinenumR was deleted. Use \Xlineflag instead.

A.8 Migration to reledpar 2.6.1

If you use perpage package to control footnote numbering, please read the handbook on 6.3.3 p. 16.
A.9 Migration to \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 \texttt{\doinsideLhook} and \texttt{\doinsideRhook} has been moved to a place where \texttt{\markboth} can work.

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

Please contact us if that causes problems.

A.11 Migration to \reledpar v. 2.17.2

When changing the \texttt{\Lcolwidth} and \texttt{\Rcolwidth} length within an \texttt{\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 \texttt{\setstanzaindents\{2,\ldots\}}, the indent was, in reality, typeset as equal to 4. This happened:

- only in parallel typesetting;
- only with the \texttt{\stanza} macro, not with the \texttt{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 \texttt{\setstanzaindents}.

References


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

Symbols

\@adv
\@astanza\@line
\@cs\@linesin\@par\@L
\@cs\@linesin\@par\@R
\@cs\@lineson\@page\@L
\@cs\@lineson\@page\@R
\@donetotallines\@L
\@donetotallines\@R
\@eledsection\@L
\@eledsection\@R
\@lab
\@lop\@L
\@lop\@R
\@morespace\@left\@note\@right\@column
\@morespace\@right\@note\@left\@column
\@nl
\@nl\@reg\@R
\@pstart\@true
\@pstart\@false
\@pstart\@R
\@pendsync\@option
\@par\@sync\@option
\@par\@this\@sync\@option
\@par\@led\@group\@note\@spacing
\@pend
\@pend\@R
\@stopastanza
\@writelinesin\@par\@L
\@writelinesin\@par\@R
\@writelineson\@page\@L
\@writelineson\@page\@R
\@writepageof\@par\@L
\@writepageof\@par\@R
COMMAND\(+
COMMAND\@@line\@flag
COMMAND\@adv
COMMAND\@cs\@linesin\@par\@L
COMMAND\@cs\@linesin\@par\@R
COMMAND\@cs\@lineson\@page\@L
COMMAND\@cs\@lineson\@page\@R
COMMAND\@eledsection\@L
COMMAND\@eledsection\@R
COMMAND\@eledsect\@mark
COMMAND\@ref\@reg\@R
COMMAND\@ref\@reg\@parse\@R
COMMAND\@pstarts\@false
COMMAND\@pstart\@true
COMMAND\@set
COMMAND\@stopastanza
COMMAND\@this\@sync\@option
COMMAND\@oped\@g\@\@note\@spacing
COMMAND\@oped\@g\@\@left\@note\@right\@column
COMMAND\@oped\@g\@\@right\@note\@left\@column
COMMAND\@oped\@g\@\@left\@note\@right\@column
COMMAND\@oped\@g\@\@right\@note\@left\@column
COMMAND\@oped\@g\@\@left\@note\@right\@column
COMMAND\@oped\@g\@\@right\@note\@left\@column
COMMAND\@oped\@g\@\@left\@note\@right\@column
COMMAND\@oped\@g\@\@right\@note\@left\@column
COMMAND\@oped\@g\@\@left\@note\@right\@column
COMMAND\@oped\@g\@\@right\@note\@left\@column
COMMAND\@oped\@g\@\@left\@note\@right\@column
COMMAND\@oped\@g\@\@right\@note\@left\@column
COMMAND\@oped\@g\@\@left\@note\@right\@column
COMMAND\@oped\@g\@\@right\@note\@left\@column
<table>
<thead>
<tr>
<th>Command</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>cleartol@devenpage</td>
<td>121, 131</td>
</tr>
<tr>
<td>clearpage</td>
<td>122, 164</td>
</tr>
<tr>
<td>clear@evenpage</td>
<td>121</td>
</tr>
<tr>
<td>columnrulewidth</td>
<td>10, 110</td>
</tr>
<tr>
<td>columns</td>
<td>168</td>
</tr>
<tr>
<td>columns@position</td>
<td>110</td>
</tr>
<tr>
<td>columnseparator</td>
<td>10</td>
</tr>
<tr>
<td>columnsposition</td>
<td>10, 162</td>
</tr>
<tr>
<td>continuousnumberingwithcolumns</td>
<td>169</td>
</tr>
<tr>
<td>correct@Xfootins@box</td>
<td>164</td>
</tr>
<tr>
<td>correct@footinSX@box</td>
<td>164</td>
</tr>
<tr>
<td>critext</td>
<td>163</td>
</tr>
<tr>
<td>csname</td>
<td>59</td>
</tr>
<tr>
<td>displaywidowpenalty</td>
<td>84</td>
</tr>
<tr>
<td>do@actions</td>
<td>76</td>
</tr>
<tr>
<td>do@actions@fixedcode</td>
<td>160</td>
</tr>
<tr>
<td>do@actions@nextR</td>
<td>77</td>
</tr>
<tr>
<td>do@ballast</td>
<td>77, 160</td>
</tr>
<tr>
<td>do@ballastR</td>
<td>84</td>
</tr>
<tr>
<td>do@insidelineLhook</td>
<td>76</td>
</tr>
<tr>
<td>do@insidelineRhook</td>
<td>162</td>
</tr>
<tr>
<td>do@line</td>
<td>69</td>
</tr>
<tr>
<td>do@line(L/R)</td>
<td>73</td>
</tr>
<tr>
<td>do@lineL</td>
<td>70, 84, 160, 161</td>
</tr>
<tr>
<td>do@lineLhook</td>
<td>160</td>
</tr>
<tr>
<td>do@lineR</td>
<td>73, 160–162</td>
</tr>
<tr>
<td>do@lineRhook</td>
<td>160</td>
</tr>
<tr>
<td>do@lockoff</td>
<td>165</td>
</tr>
<tr>
<td>do@lockoffR</td>
<td>48</td>
</tr>
<tr>
<td>do@lockon</td>
<td>165</td>
</tr>
<tr>
<td>do@lockonR</td>
<td>48</td>
</tr>
<tr>
<td>doinsidelineLhook</td>
<td>22, 140, 163, 167</td>
</tr>
<tr>
<td>doinsidelineRhook</td>
<td>22, 140, 163, 167</td>
</tr>
<tr>
<td>doinsidelinehook</td>
<td>22</td>
</tr>
<tr>
<td>doinsidethislinehook</td>
<td>170</td>
</tr>
<tr>
<td>dolineLhook</td>
<td>22, 163</td>
</tr>
<tr>
<td>dolineRhook</td>
<td>22, 163</td>
</tr>
<tr>
<td>dolinehook</td>
<td>22</td>
</tr>
<tr>
<td>edindex</td>
<td>163</td>
</tr>
<tr>
<td>edlabel</td>
<td>161, 163</td>
</tr>
<tr>
<td>edlineref</td>
<td>168</td>
</tr>
<tr>
<td>edtext</td>
<td>17, 49, 54, 89, 163, 167, 168</td>
</tr>
<tr>
<td>edtext@later</td>
<td>89, 90, 120</td>
</tr>
<tr>
<td>edtext@now</td>
<td>120</td>
</tr>
<tr>
<td>edtextlater</td>
<td>17, 18, 88–90, 167–169</td>
</tr>
<tr>
<td>edtextnow</td>
<td>17, 18, 88–90, 167, 168</td>
</tr>
<tr>
<td>eled@sectioningR@out</td>
<td>132</td>
</tr>
</tbody>
</table>
### Index

<table>
<thead>
<tr>
<th>Command</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>eledchapter</code></td>
<td>164</td>
</tr>
<tr>
<td><code>eledsection</code></td>
<td>163–165, 169</td>
</tr>
<tr>
<td><code>eledsection@correcting@skip</code></td>
<td>132</td>
</tr>
<tr>
<td><code>eledsectmark</code></td>
<td>25, 132</td>
</tr>
<tr>
<td><code>eledsectnotoc</code></td>
<td>25, 132</td>
</tr>
<tr>
<td><code>eledxxx</code></td>
<td>163</td>
</tr>
<tr>
<td><code>end</code></td>
<td>23</td>
</tr>
<tr>
<td><code>endgraf</code></td>
<td>67</td>
</tr>
<tr>
<td><code>endlock</code></td>
<td>54, 165</td>
</tr>
<tr>
<td><code>endnumbering</code></td>
<td>11, 18, 22, 35, 164, 168, 169</td>
</tr>
<tr>
<td><code>endnumberingR</code></td>
<td>53</td>
</tr>
<tr>
<td><code>endsub</code></td>
<td>54, 165</td>
</tr>
<tr>
<td><code>endnumbering</code></td>
<td>18</td>
</tr>
<tr>
<td><code>expandafter</code></td>
<td>56</td>
</tr>
<tr>
<td><code>extensionchars</code></td>
<td>34</td>
</tr>
<tr>
<td><code>firstlinenum</code></td>
<td>19, 162, 165, 166</td>
</tr>
<tr>
<td><code>firstsublinenum</code></td>
<td>162, 165, 166</td>
</tr>
<tr>
<td><code>fix@page</code></td>
<td>47, 165</td>
</tr>
<tr>
<td><code>flag@end</code></td>
<td>54, 160, 163</td>
</tr>
<tr>
<td><code>flag@start</code></td>
<td>54, 163</td>
</tr>
<tr>
<td><code>flush@notesR</code></td>
<td>85</td>
</tr>
<tr>
<td><code>footnote</code></td>
<td>58</td>
</tr>
<tr>
<td><code>footnoteX</code></td>
<td>55, 56</td>
</tr>
<tr>
<td><code>footnoteX@reading</code></td>
<td>166</td>
</tr>
<tr>
<td><code>footnoteXmk</code></td>
<td>17</td>
</tr>
<tr>
<td><code>footnotexnomk</code></td>
<td>17, 56</td>
</tr>
<tr>
<td><code>frontmatter</code></td>
<td>15, 25</td>
</tr>
<tr>
<td><code>get@intercolumns@width</code></td>
<td>105, 108</td>
</tr>
<tr>
<td><code>get@nextboxL</code></td>
<td>161</td>
</tr>
<tr>
<td><code>get@nextboxR</code></td>
<td>161</td>
</tr>
<tr>
<td><code>get@sidенote@more@space@columns</code></td>
<td>92, 93</td>
</tr>
<tr>
<td><code>getline@numL</code></td>
<td>75</td>
</tr>
<tr>
<td><code>getline@numR</code></td>
<td>75</td>
</tr>
<tr>
<td><code>getlinefrompagelistL</code></td>
<td>123</td>
</tr>
<tr>
<td><code>getlinefrompagelistR</code></td>
<td>123</td>
</tr>
<tr>
<td><code>getlinefromparlistL</code></td>
<td>122</td>
</tr>
<tr>
<td><code>getlinefromparlistR</code></td>
<td>122</td>
</tr>
<tr>
<td><code>glip</code></td>
<td>56</td>
</tr>
<tr>
<td><code>goalfractio</code></td>
<td>138</td>
</tr>
<tr>
<td><code>hangingsymbol</code></td>
<td>138, 160</td>
</tr>
<tr>
<td><code>hfill</code></td>
<td>105, 111</td>
</tr>
<tr>
<td><code>hidenumbering</code></td>
<td>19, 165</td>
</tr>
<tr>
<td><code>@getnextbox</code></td>
<td>127</td>
</tr>
<tr>
<td><code>@sidенотesc@margin@page</code></td>
<td>92</td>
</tr>
<tr>
<td><code>@pages@mainmatter</code></td>
<td>131</td>
</tr>
<tr>
<td><code>@bypage@</code></td>
<td>165</td>
</tr>
<tr>
<td><code>@bypstart@R</code></td>
<td>165</td>
</tr>
<tr>
<td><code>@csboxvoid</code></td>
<td>33</td>
</tr>
<tr>
<td><code>@dim</code></td>
<td>105</td>
</tr>
<tr>
<td><code>@ins@erthangingsymbol</code></td>
<td>94</td>
</tr>
</tbody>
</table>
Index

<table>
<thead>
<tr>
<th>Command</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>@drsnote</td>
<td>162</td>
</tr>
<tr>
<td>@dsetupmaxlinecounts</td>
<td>99</td>
</tr>
<tr>
<td>@duselanguage</td>
<td>97, 98, 160</td>
</tr>
<tr>
<td>@dzeromaxlinecounts</td>
<td>99</td>
</tr>
<tr>
<td>@prev@nopbR</td>
<td>133</td>
</tr>
<tr>
<td>@prev@phnumR</td>
<td>133</td>
</tr>
<tr>
<td>@pbR</td>
<td>133</td>
</tr>
<tr>
<td>@phnumR</td>
<td>133</td>
</tr>
<tr>
<td>@innernote</td>
<td>23</td>
</tr>
<tr>
<td>@leftnote</td>
<td>23</td>
</tr>
<tr>
<td>@linenote</td>
<td>23</td>
</tr>
<tr>
<td>@nopbR</td>
<td>110, 125</td>
</tr>
<tr>
<td>@phR</td>
<td>133</td>
</tr>
<tr>
<td>@phnumR</td>
<td>133</td>
</tr>
<tr>
<td>@rightnote</td>
<td>23</td>
</tr>
<tr>
<td>@rightnote</td>
<td>23</td>
</tr>
<tr>
<td>@strutL</td>
<td>160</td>
</tr>
<tr>
<td>@strutR</td>
<td>160, 166</td>
</tr>
<tr>
<td>@theoal</td>
<td>126</td>
</tr>
<tr>
<td>@theoal</td>
<td>126</td>
</tr>
<tr>
<td>@linenote</td>
<td>43, 160</td>
</tr>
<tr>
<td>@linenumnumR</td>
<td>44</td>
</tr>
<tr>
<td>@linenumber</td>
<td>89, 90</td>
</tr>
<tr>
<td>@linet</td>
<td>56</td>
</tr>
<tr>
<td>@list@R</td>
<td>50</td>
</tr>
<tr>
<td>@list@stuff</td>
<td>46, 53</td>
</tr>
<tr>
<td>@margin</td>
<td>39</td>
</tr>
<tr>
<td>@marginR</td>
<td>39, 160</td>
</tr>
<tr>
<td>@numR</td>
<td>44</td>
</tr>
<tr>
<td>@lineation</td>
<td>20, 164, 169</td>
</tr>
<tr>
<td>@lineation*</td>
<td>20, 39, 163</td>
</tr>
<tr>
<td>@lineationR</td>
<td>20, 38, 164, 167</td>
</tr>
<tr>
<td>@linenumber</td>
<td>89</td>
</tr>
<tr>
<td>@linenum@out</td>
<td>91</td>
</tr>
<tr>
<td>@linenum@outR</td>
<td>52</td>
</tr>
<tr>
<td>@linenumOnlyPagesForColumns</td>
<td>21, 40, 169</td>
</tr>
<tr>
<td>@linenumOnlyPagesForColumnsR</td>
<td>21, 40, 169</td>
</tr>
<tr>
<td>@linenumnotationothersidettrue</td>
<td>170</td>
</tr>
<tr>
<td>@linenumnumberLevenifblanktrue</td>
<td>21, 167</td>
</tr>
</tbody>
</table>
Index

\texttt{COMMAND\linenumberRevenifblank} ........................................ 167
\texttt{COMMAND\linenumberRevenifblanktrue} ................................... 21
\texttt{COMMAND\linenumberannotation} ............................................ 169
\texttt{COMMAND\linenumberlist} .................................................... 20, 140, 167
\texttt{COMMAND\linenumberlistR} ................................................. 20, 140, 167
\texttt{COMMAND\linenumberstyle} ................................................... 20, 139, 168
\texttt{COMMAND\linenumberstyle*} .............................................. 139
\texttt{COMMAND\linenumberstyleR} .............................................. 20, 139
\texttt{COMMAND\linenumberincrement} .......................................... 19, 162, 165, 166
\texttt{COMMAND\linenummargin} .................................................... 20, 21, 39, 160, 165–167
\texttt{COMMAND\linenummargin*} ................................................ 20, 39, 166
\texttt{COMMAND\linenummarginColumns} ....................................... 20, 40
\texttt{COMMAND\linenummarginColumns*} ..................................... 20
\texttt{COMMAND\linenummarginColumnsR} ..................................... 20
\texttt{COMMAND\linenummarginR} .................................................. 20, 21, 39, 166
\texttt{COMMAND\linenumrepl} ..................................................... 42, 160
\texttt{COMMAND\linesinpar@listL} ................................................. 51, 122
\texttt{COMMAND\linesonpage@listL} .............................................. 52, 123
\texttt{COMMAND\lock@off} .......................................................... 48
\texttt{COMMAND\lock@on} ............................................................. 48
\texttt{COMMAND\mainmatter} ....................................................... 3, 15, 25, 166
\texttt{COMMAND\makeatletter} ...................................................... 73
\texttt{COMMAND\markboth} .......................................................... 140, 167
\texttt{COMMAND\maxchunks} ......................................................... 9, 23, 99, 100
\texttt{COMMAND\maxdimen} ......................................................... 86
\texttt{COMMAND\maxhnotesX} ....................................................... 16
\texttt{COMMAND\memorydump} ...................................................... 18, 37
\texttt{COMMAND\movecolumnspositiononrightpage} ............................ 109
\texttt{COMMAND\msdata} ............................................................ 168
\texttt{COMMAND\num@num} .......................................................... 164
\texttt{COMMAND\new@lineL} ......................................................... 53
\texttt{COMMAND\new@lineR} .......................................................... 54
\texttt{COMMAND\newhookcommand@series} .................................... 58
\texttt{COMMAND\newf} ............................................................... 164
\texttt{COMMAND\newpage} ........................................................... 121, 122, 164
\texttt{COMMAND\newseries} ......................................................... 58
\texttt{COMMAND\newseries@} ....................................................... 55
\texttt{COMMAND\newseries@par} .................................................. 55, 57, 58
\texttt{COMMAND\next@line@list@stuffR} ........................................ 53, 169
\texttt{COMMAND\noeledxxx} ........................................................ 163
\texttt{COMMAND\nomark@} ........................................................... 56
\texttt{COMMAND\nomaxlines} ....................................................... 52
\texttt{COMMAND\notesXwidthliketwocolumns} .................................. 11, 163
\texttt{COMMAND\num@lines} ........................................................ 84
\texttt{COMMAND\num@lines(R)} ..................................................... 62
\texttt{COMMAND\numberingR} ....................................................... 36
\texttt{COMMAND\numberlinefalse} .................................................. 8, 169
\texttt{COMMAND\numberlinetrue} ................................................... 169
\texttt{COMMAND\numberonlyfirstinline} ......................................... 161
\texttt{COMMAND\numberpstartfalse} ............................................ 19, 169
# Index

<table>
<thead>
<tr>
<th>Package</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>memoir</td>
<td>140</td>
</tr>
<tr>
<td>musixtex</td>
<td>162</td>
</tr>
<tr>
<td>perpage</td>
<td>3, 16, 139, 166</td>
</tr>
<tr>
<td>polyglossia</td>
<td>22, 96–98, 167, 170</td>
</tr>
<tr>
<td>setspace</td>
<td>4, 25</td>
</tr>
<tr>
<td>syntonly</td>
<td>166</td>
</tr>
<tr>
<td>keyval</td>
<td>26</td>
</tr>
<tr>
<td>keyfr</td>
<td>167</td>
</tr>
</tbody>
</table>

## A

<table>
<thead>
<tr>
<th>Command</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\absline@numR</td>
<td>1</td>
</tr>
<tr>
<td>\actionlines@listR</td>
<td>1</td>
</tr>
<tr>
<td>\actions@listR</td>
<td>1</td>
</tr>
<tr>
<td>\add@apparatusL</td>
<td>1</td>
</tr>
<tr>
<td>\add@apparatusR</td>
<td>1</td>
</tr>
<tr>
<td>\add@inserts@nextR</td>
<td>1</td>
</tr>
<tr>
<td>\add@insertsR</td>
<td>1</td>
</tr>
<tr>
<td>\add@penaltiesL</td>
<td>1</td>
</tr>
<tr>
<td>\add@penaltiesR</td>
<td>1</td>
</tr>
<tr>
<td>\advanceline</td>
<td>1</td>
</tr>
<tr>
<td>\affixline@numR</td>
<td>1</td>
</tr>
<tr>
<td>\affixpstart@numL</td>
<td>1</td>
</tr>
<tr>
<td>\affixpstart@numR</td>
<td>1</td>
</tr>
<tr>
<td>\affixside@noteR</td>
<td>1</td>
</tr>
<tr>
<td>\aftercolumnseparator</td>
<td>1, 10</td>
</tr>
<tr>
<td>\annot@listR</td>
<td>1</td>
</tr>
<tr>
<td>\araw@textfalse</td>
<td>1</td>
</tr>
<tr>
<td>\araw@texttrue</td>
<td>1</td>
</tr>
<tr>
<td>\astanza (environment)</td>
<td>23</td>
</tr>
<tr>
<td>\AtBeginPairs</td>
<td>1, 10</td>
</tr>
<tr>
<td>\AtEveryPstartCall</td>
<td>1</td>
</tr>
<tr>
<td>\autopar</td>
<td>1</td>
</tr>
</tbody>
</table>

## B

<table>
<thead>
<tr>
<th>Command</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\bbil@set@language</td>
<td>1</td>
</tr>
<tr>
<td>\beforecolumnseparator</td>
<td>1, 10</td>
</tr>
<tr>
<td>\beginnumbering</td>
<td>18</td>
</tr>
<tr>
<td>\beginnumberingR</td>
<td>1</td>
</tr>
</tbody>
</table>

## C

<table>
<thead>
<tr>
<th>Command</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\c@firstlinenumR</td>
<td>1</td>
</tr>
<tr>
<td>\c@firstsublinenumR</td>
<td>1</td>
</tr>
<tr>
<td>\c@linenumincrementR</td>
<td>1</td>
</tr>
<tr>
<td>\c@sublinenumincrementR</td>
<td>1</td>
</tr>
<tr>
<td>\ch@ck@ck@ckR</td>
<td>1</td>
</tr>
<tr>
<td>\ch@cks@cks@ckR</td>
<td>1</td>
</tr>
<tr>
<td>\chapter</td>
<td>1</td>
</tr>
</tbody>
</table>
Index

\chapterinpages ................................................. 1
\check@goal .................................................... 1
\check@postarts ............................................... 1
\check@pageL .................................................. 1
\check@pageR .................................................. 1
\check@pbcolumns ............................................. 1
\check@pbl ..................................................... 1
\check@pbR ..................................................... 1
\check@verse@text ............................................ 1
\check@verseL ................................................ 1
\check@verseR ................................................ 1
\clear@odleftpage .......................................... 1
\clear@odrightpage ......................................... 1
\clear@oevenpage ........................................... 1
\clear@oevenpage ........................................... 1
\column@rulewidth .......................................... 1.10
\Columns ..................................................... 1.9
\columns@position ........................................... 1
\Columns@print@after@pend ................................ 1
\Columns@print@before@start ................................ 1
\columnsep ................................................... 1
\columns@position ........................................... 1.10
\correct@footinsX@box ...................................... 1
\correct@Xfootins@box ...................................... 1
\count@Lline ................................................ 1
\count@Rline ................................................ 1
\critext ....................................................... 1

D
\do@actions@fixedcodeR ........................................ 1
\do@actions@nextR ........................................... 1
\do@actionsR ................................................ 1
\do@ballastR ................................................ 1
\do@insideline@hook ........................................ 1
\do@insidelineRhook ......................................... 1
\do@lineL ..................................................... 1
\do@lineR ..................................................... 1
\do@lineRhook ................................................ 1
\do@lockoff .................................................. 1
\do@lockoffR ................................................ 1
\do@lockon .................................................... 1
\do@lockonR .................................................. 1
\do@insideline@hook ........................................ 1.22
\do@insidelineRhook ......................................... 1.22
\dolineRhook ................................................ 1.22
\dolineRhook ................................................ 1.22
\dump@postartL@pc .......................................... 1
\dump@postartR@pc .......................................... 1
\lock@off .................................................. 1
\maxchunks ................................................ 1, 9
\maxlinesinpar@list ........................................ 1
\memorydump ................................................ 18
\memorydumpL ................................................ 1
\memorydumpR ................................................ 1
\n@num ..................................................... 1
\namebox ................................................... 1
\new@lineL .................................................. 1
\new@lineR .................................................. 1
\newnamebox ................................................ 1
\newnamecount ............................................. 1
\newseries@par ........................................... 1
\next@line@list@stuffR .................................... 1
\notesXwidthliketwocolumns ................................ 11
\num@linesR ................................................. 1
\numberpstartfalse ....................................... 19
\numberpstarttrue .......................................... 19
\numpagelinesL ............................................ 1
\numpagelinesR ............................................ 1
\one@lineR .................................................. 1
\onlysideX .................................................. 17
\optionadvancedshiftedpstarts ............................ 13, 14, 27
\optionauxdir ............................................. 167
\optioncontinuousnumberingwithcolumns .................. 11, 167–169
\optionmainmatter .......................................... 131
\optionmovecolumnspositiononrightpage .................. 12, 168
\optionnomaxlines ......................................... 13, 14, 27
\optionnopbinverse ....................................... 168
\optionnosyncpstarts ...................................... 14, 27, 127
\optionprevpnotnumbered .................................. 168
\optionprevpgstyle ........................................ 168
\optionsameparallelpagernumber .......................... 165, 169
\optionshiftedpstarts ..................................... 7, 14, 27, 161
\optionshiftedverses ...................................... 160, 161
\optionsidenotesmarginpage ................................ 24, 28, 169
\optionwidthliketwocolumns ................................ 11, 169

\page@action ................................................ 1
\page@numR .................................................. 1
\Pages ...................................................... 1, 12
pages (environment) ....................................... 12
\Pages@mainmatter .......................................... 1
pairs (environment) ................................................................. 9
\par@lineR ................................................................. 1
\parledgroup@ ................................................................. 1
\parledgroup@beforenotes@save ................................................ 1
\parledgroup@beforenotesL ................................................ 1
\parledgroup@beforenotesR ................................................ 1
\parledgroup@correction@notespacing ........................................ 1
\parledgroup@correction@notespacing@final ................................ 1
\parledgroup@correction@notespacing@init ................................ 1
\parledgroup@notes@startL ................................................ 1
\parledgroup@notes@startR ................................................ 1
\parledgroup@notespacing@correction ....................................... 1
\parledgroup@notespacing@set@correction ................................... 1
\parledgroupseries@ .......................................................... 1
\parledgroup@typo ........................................................... 1
\pausenumbering@page@numR .................................................. 1
\pausenumberingR ............................................................. 1
\pend ................................................................. 21
\pendL ................................................................. 1
\pendR ................................................................. 1
\prev@nopbR ............................................................... 1
\prev@pbR ............................................................... 1
\prevpgstyle ............................................................... 1
\print@columnseparator ..................................................... 1
\print@eledsectionL ....................................................... 1
\print@eledsectionR ....................................................... 1
\print@leftcolumn ........................................................ 1
\print@lineL ............................................................. 1
\print@lineR ............................................................. 1
\print@notesX@forpages .................................................... 1
\print@rightcolumn ........................................................ 1
\print@Xnotes@forpages .................................................... 1
\pstart ................................................................. 21
\pstartL ................................................................. 1
\pstartR ................................................................. 1

R ....................................................................................... 1
\Rcolwidth ................................................................. 1, 10, 13
\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
\rightlinenumbering@annotationR ........................................... 1
\rightlinenumR ............................................................. 1
\rightpagehook ............................................................. 1
\rightpstartnumL ........................................................... 1

Index

\this@section@last@page@numR ........................................... 1
\this@section@page@numR ........................................... 1

U
\unhnamebox ......................................................... 1
\unvnamebox ......................................................... 1
\usenamecount ....................................................... 1

W
\widthliketwocolumns ............................................. 11

X
\Xendlineflag ....................................................... 18
\Xlineflag ........................................................... 18
\Xnoteswidthliketwocolumns ...................................... 11
\Xonlyside ........................................................... 17
Change History

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

v0.2.0.
\Columns: Added \dchecklang and \duselanguage to \Columns ............... 103
\Pages: Added \duselanguage to \Pages ..................................... 116
General: Added section of babel related code ............................. 96
Fix babel problems ............................................................... 1

v0.3.0.
\Pages: Added \ledstrutL to \Pages .......................................... 116
Added \ledstrutR to \Pages ................................................... 118
\Rightsidehookend: Added \Leftsidehook, \Leftsidehookend, \Rightsidehook
and \Rightsidehookend ....................................................... 61
\affixline@numR: Changed \affixline@numR to match new eledmac ......... 78
\do@actions@nextR: Used \do@actions@fixedcode in \do@actionsR .......... 77
\do@lineL: Added \do@lineLhook to \do@lineL ............................. 70
Simplified \do@lineL by using macros for some common code .......... 70
\do@lineR: Changed \do@lineR similarly to \do@lineL ................... 73
\flag@end: Removed extraneous spaces from \flag@end .................. 54
\ifledRcol: Moved \ifl@dpairing to eledmac ................................ 29
\ifp@stedR: Moved \ifp@stedL to eledmac .................................. 34
\ld@linenumR: Simplified \leftlinenumR and \rightlinenumR by introducing
\ld@linenumR ................................................................. 43
\ld@numpstartsR: Moved \ld@numpstartsL to eledmac ..................... 99
\ledstrutR: Added \ledstrutL and \ledstrutR .............................. 121
\sublinenumrepR: Added \linenumrepR and \sublinenumrepR ............ 42
General: Added \do@lineLhook and \do@lineRhook ......................... 73
Added hooks into Leftside environment ................................. 61
Reorganize for ledarab ..................................................... 1

v0.3.a.
\line@marginR: Do not just set \line@marginR in \linenummargin .......... 39
General: Minor \linenummargin fix .......................................... 1

v0.3.b.
\Pages: Added \d@minpagelines calculation for succeeding page pairs ... 119
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 use \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
Change History

v0.9.0.
- \ifledRcol: Moved \ifledRcol and \ifnumberingR to eledmac
- General: Possibility to number \pstart
- Possibility to number the pstart with the commands \numberpstarttrue

v0.9.1.
- General: The numbering of the pstarts restarts on each \beginnumbering

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

v0.9.3.
- General: \thepstartL and \thepstartR use now \bfseries and not \bf, which is deprecated and makes conflicts with memoir class

v0.10.0.
- General: \edlabel commands on the right side are now correctly indicated
- \edlabel commands which start a paragraph are now put in the right place

v0.11.0.
- \Columns: Line numbering by pstart
- \affixline@numR: Changed \affixline@numR to allow to disable line numbering (like in eledmac 0.15)
- \get@nextboxR: Change \get@nextboxL and \get@nextboxR to allow to disable line numbering (like in eledmac 0.15)
- \psstart number can be printed in side
- \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)
- Lineation can be by pstart (like in eledmac 0.15)
- New management of hangingsymbol insertion, preventing undesirable insertions

v0.12.0.
- General: New management of hangingsymbol insertion, preventing undesirable insertions

v1.0.0.
- General: Compatibility with eledmac. Change name to eledpar
- Debug in lineation by pstarts

v1.0.1.
- General: Correction on \numberonlyfirstinline with lineation by pstart or by page

v1.1.0.
- \psstartR: Add \labelpstarttrue (from eledmac)
- General: The shiftedverses options becomes shiftedpstarts

v1.1.1.
- \psstartR: Correct \psstartR bug introduced by 1.1

v1.1.2.
- \affixside@noteR: Remove spurious space between line number and line content

v1.2.0.
- General: Support for \ledsection commands in parallel texts

v1.2.1.
- \set@sectcountR: For the right section, the counter is defined only once

v1.3.0.
- \edtext: Manage RTL language

v1.3.2.
- General: Debug with some classes
<table>
<thead>
<tr>
<th>Version</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.3.3.</td>
<td>General: Debugging the left notes of the right column.</td>
</tr>
<tr>
<td>v1.3.4.</td>
<td>General: Allow use of commands in sidenotes, as introduced by eledmac 1.0</td>
</tr>
<tr>
<td>v1.4.0.</td>
<td>General: Added \do@insidelineLhook and \do@insidelineRhook</td>
</tr>
<tr>
<td>v1.4.1.</td>
<td>General: Enable the use of stanzaindentsrepetition within astanza environment</td>
</tr>
<tr>
<td>v1.4.3.</td>
<td>\inserthangingsymbolR: Hanging verse is no longer automatically flush right.</td>
</tr>
<tr>
<td></td>
<td>\pendL: Spurious spaces in \pendL</td>
</tr>
<tr>
<td></td>
<td>\pendR: Spurious spaces in \pstartR</td>
</tr>
<tr>
<td></td>
<td>\pstartR: Spurious spaces in \pstartL and \pstartR</td>
</tr>
<tr>
<td></td>
<td>General: Corrects a false hanging verse when a verse is exactly the length of a line.</td>
</tr>
<tr>
<td>v1.5.0.</td>
<td>\sublinenumincrement*: Add starred version of \firstlinenum,</td>
</tr>
<tr>
<td></td>
<td>\linenumincrement, \firstsublinenum, \sublinenumincrement to change both Left and Rightside</td>
</tr>
<tr>
<td></td>
<td>General: Add, as in eledmac, features to manage page breaks.</td>
</tr>
<tr>
<td>v1.6.0.</td>
<td>General: Add tool and documentation for parallel ledgroups</td>
</tr>
<tr>
<td>v1.7.0.</td>
<td>General: Add, as in eledmac, features to make crossrefs with pstart numbers.</td>
</tr>
<tr>
<td>v1.8.0.</td>
<td>\Columns: Modify \Columns to enable to add section's title.</td>
</tr>
<tr>
<td></td>
<td>Suppress \ld@checklang from \Columns</td>
</tr>
<tr>
<td></td>
<td>\Pages: Modify \Pages to enable to add section's title.</td>
</tr>
<tr>
<td></td>
<td>\ld@checklang: Suppress \ld@checklang which did not work and was not logical, because both columns could have the same language but not the main language of the document.</td>
</tr>
<tr>
<td></td>
<td>\pendL: As in eledmac, \pendL can have an optional argument.</td>
</tr>
<tr>
<td></td>
<td>\pendR: As in eledmac, \pendR can have an optional argument.</td>
</tr>
<tr>
<td></td>
<td>\print@columnseparator: Move some code of \Columns to \print@columnseparator.</td>
</tr>
<tr>
<td></td>
<td>\pstartR: As in eledmac, \pstartL and \pstartR can have an optional argument.</td>
</tr>
<tr>
<td></td>
<td>\sidenotemarginR: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar.</td>
</tr>
<tr>
<td></td>
<td>Add \sidenotemargin*</td>
</tr>
<tr>
<td></td>
<td>\theledlanguageR: Correct left/right language setting with polyglossia</td>
</tr>
<tr>
<td></td>
<td>General: \beginnumbering is defined only on eledmac, not on eledpar.</td>
</tr>
<tr>
<td></td>
<td>\ld@l@note, \ld@r@note, \ld@c@note defined only one time, in eledmac.</td>
</tr>
<tr>
<td></td>
<td>Add \beforecolumnseparator and \aftercolumnseparator</td>
</tr>
<tr>
<td></td>
<td>Add \columnsposition</td>
</tr>
<tr>
<td></td>
<td>Add, as in eledmac, new system of sectioning commands.</td>
</tr>
<tr>
<td></td>
<td>Add, as in eledmac, option to insert something after \pends / verses.</td>
</tr>
<tr>
<td></td>
<td>Add, as in eledmac, option to insert something between \pstarts / verse.</td>
</tr>
<tr>
<td></td>
<td>Change \do@lineR and \do@lineL to allow new sectioning commands.</td>
</tr>
<tr>
<td></td>
<td>Compatibility with musixtex.</td>
</tr>
<tr>
<td></td>
<td>Debug eledmac sectioning command after using \resumenumbering.</td>
</tr>
<tr>
<td></td>
<td>New sectioning commands, as in eledmac.</td>
</tr>
</tbody>
</table>
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.

v1.8.1.
- \do@lineL: Fix a bug with critical notes at the beginning of a page, (maybe added by v1.8.0 (?)).
- \do@lineR: Fix a bug with critical notes at the beginning of a page, added by v1.8.0 (?).

v1.8.2.
- flag@end: flag@start and flag@end are now defined only one time for eledmac and eledpar.
- lineation*: Add lineation*.
- reledperror: Errors specific to eledpar send to eledpar handbook.
- General: Debug \eledsection with some paper sizes.
- Debug left and side note (bugs added by 1.8.0).

v1.8.3.
- Pages: Debug blank pages when using optional argument in the last \pend.
- pausenumbering@pa@numR: Debug resumenumberingR.
- General: Add \noeledxxx, as in eledmac.

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

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

v1.10.0.
- \xpages: Debug wrong pages splitting when no optional argument is used in last \pend (bug was added in v1.8.3).
- Debug wrong parallel pages synchronization when an \edtext falls across two pages.
- General: Compatibility with \AtEveryPstart and \AtEveryPend.
- Restore critical notes in \eledsection in parallel columns (this bug was added in 1.8.2).

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.

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

v1.12.0.
- Columns: Add \l@dprintingcolumnstrue.
- \xpages: Add \l@dprintingpagestrue.
- edlabel: edlabel and \edindex works now with hyperref when using eledpar.
- edlabel is now defined only one time for both eledmac and eledpar.
- print@eledsectionL: Compatibility with LuaLaTeX RTL languages.
- General: Compatibility with LuaLaTeX RTL languages.
General: Compatibility with LuaLaTeX RTL languages. ........................................ 1

v1.12.1.
\printEledsection: Fix bug with LuaLaTeX RTL \eledsection. .......................... 72

v1.13.0.
\Pages: Prevent false overfull hboxes when using \Pages outside of pages environment. .............................. 114
When using shiftedpstarts option, a \eledleftbox with a null height will advance the \pagetotal in any case. .............................. 113
\clearEledrightpage: Use \newpage instead of \cleardoublepage. ......................... 122
\ifleddRcol: Remove false boolean settings which are not needed. .................. 29
General: Enable the use of optional argument of & in astanza environment. .......... 95
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) ...................... 113
\correct@footins\X@box: Call \correct@footins\X@box and \correct@Xfootins\X@box directly in \print@notes\X@forpages and \print@Xnotes\X@forpages. ......................................................... 85
Correct \correct@footins\X@box and \correct@Xfootins\X@box ......................... 85

v1.14.0.
General: Fix a bug with line number position when using \eledsection and similar commands for RTL texts with LuaLaTeX. ........................................ 1
The \newif are not followed by boolean values set to false, because it is the \LaTeX default setting. ................................................................. 1

v1.15.0.
\do@actions@nextR: Add actions 1008 and 1009 ................................................. 77
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting when a verse is hanging ................................................. 94
\lineationR: As \lineation, \lineationR automatically set the \pstartinfootnote. ......................... 38
\n@num: \n@num defined only one time for both Eledmac and Eledpar. .................. 49
\skipnumbering: \skipnumbering defined only one time for both Eledmac and Eledpar ................................................. 54
General: Add \AtEveryPstartCall ................................................................. 1
Add sameparallelpagenumber option. ......................................................... 15
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 ................................................. 55
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 .................................................. 19
v2.0.0.
  \@adv: \@adv defined only in reledmac. .................................... 47
  \@lab: \@lab defined only in eledmac. ..................................... 91
  \@ref@regR: \@ref defined only in reledmac, code specific to right side moved in \@ref@regR. ............................................................ 49
  \@set: \@set defined only in reledmac. ..................................... 47
  \advanceline: \advanceline defined only in reledmac. ....................... 54
  \bb1@set@language: Patch \bb1@set@language instead of redefining it .......... 97
  \@do@lockonR: \@do@lockon defined only in reledmac. ......................... 48
  \endlock: \@startlock and \@endlock defined only in reledmac. .............. 54
  \endsub: \@startsub and \@endsub defined only in reledmac. ................ 54
  \fix@page: \fix@page is defined only once in reledmac ...................... 47
  \l@d@set: \l@d@set defined only in reledmac. .................................. 48
  \line@marginR: \line@marginR is defined only once time in reledmac ....... 39
  \page@action: \page@action defined only in reledmac. ....................... 48
  \read@linelist: \read@linelist is defined only once time in \reledmac ....... 46
  \set@line: \set@line defined only in reledmac. ................................ 54
  \set@line@action: \set@line@action defined only in reledmac ................ 48
  \set@line: \set@line defined only in reledmac. ................................ 54
  \set@linenum: \set@linenum defined only in reledmac. ....................... 54
  \skip@lockoff: \do@lockoff defined only in reledmac. ......................... 48
  \sub@action: \sub@action defined only in reledmac. ......................... 48
  \sub@linenumincrement*: \first@linenum, \linenumincrement, \first@sub@linenum, \sub@linenumincrement are now defined only in reledmac. 41
  \@n@l is now defined only in reledmac. ..................................... 46
  \if@by@page@0 and \if@by@p@start@0 defined in eledmac. .................... 38
  Fix some bugs with sameparallelpagenumber option. ........................ 1
  Many code refactored and moved to reledmac. ................................ 1
  Package's name becomes reledpar. ......................................... 1
  Totally new implementation of sameparallelpagenumber option. ............. 1
  chapterinpagedelete: Deleting the old system of managing parallel chapter, keep only the new one with \patchcmd. .................. 60
v2.1.0.
  General: Fix a bug when using \eled@section and related on right pages when page width is short. ........................................... 1
  Fix a bug when using \pagenumbering with memoir (bug added in v2.0.0). ...... 1
  Fix a bug with \set@par@ed@group@not@es@pa@cing with the shiftedpstarts option. 1
  Fix incompatibility between optional argument of \pstart and \numberpstart@true 1
  Options to custom empty right page before \Pages. .......................... 1
Change History

v2.2.0.
General: astanza environment can take an optional argument, which will be the
optional argument of \pstart started by this environment. ................. 1
New tools to number stanza ............................................. 1

v2.2.1.
General: Fix a bug with optional argument of last left \pend .................. 1

v2.3.0.
\Pages: Fix a bug when calling \Columns after a \Pages (bug added in v1.13.0). ... 114
General: Change some internal codes in order to provide compatibility with \TeX\ release
of october 2015 .................................................................... 1
Fix a bug with title number in parallel columns ............................... 1
New line setting command suffixed by R to set only the right side. ....... 1

v2.4.0.
\ledstrutR: Deleted \ledstrutL and \ledstrutR .................................. 121
Fix a bug with dotted letter ..................................................... 121
General: New way of (not) synchronizing the parallel pages. ............ 1
Option to switch to \mainmatter when calling \Pages ......................... 1

v2.5.0.
General: Disable empty lines as paragraph in astanza. ..................... 1
Fix a bug introduced in v1.15.0 which made hanging indentation in verse not work
anymore. .............................................................................. 1
New commands \linenummarginR and \linenummargin* ...................... 1

v2.5.1.
General: Fix spurious space when using optional argument of astanza environment
(introduced in v2.5.0). ......................................................... 1

v2.5.2.
General: Fix a bug introduced in v2.5.0 with \linenummargin, \firstlinenum,
\linenumincrement, \firstsublinenum, \sublinenumincrement ............. 1

v2.6.0.
\l@dm@ke@l@e@b@l@e@sR: \@Rlineflag is not stored directly after the line number, but as a
fifth argument of \the@labelX. Can be retrieved by \xflagref. ............... 91
General: \Xlineflag and \Xendlineflag added ................................. 1
\printlinesR deleted ............................................................. 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 \syntaxonly
command of the syntaxonly 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 relemac. ... 55
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

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

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

v2.10.0.
\do@actions@nextR: Add action 1010 .............................................. 77
General: Add new tools to make apparatuses of manuscripts .................................. 1

v2.11.0.
\correct@footinsX@box: Clarification in the handbook about the use of \Xonlyside and onlysideX and error message if misused. ......................... 85
New implementation of \Xonlyside and onlysideX hooks, prevent trouble with vertical spacing. ............................................................. 85
General: Compatibility with reledmac’s auxdir option. ............................... 1

v2.12.0.
General: Add \edtextlater and \edtextnow ........................................... 1, 88
Fix a bug with \AtBeginPairs ............................................................. 1
Fix a bug with \lineationR ............................................................. 1

v2.13.0.
General: Compatibility with the new features of reledmac 2.15.0 .................. 1

v2.13.1.
General: Simplification of the code concerning the sectioning command ............ 1

v2.14.0.
General: Add \linenumberlistR, equivalent to \linenumberlist for the right side. . 1
Error message when left / right side are not defined ...................................... 1
More explicit error message when the stanza indentation is not defined. ............. 1
New commands to have specific line margins for text running in parallel columns ... 1
When indexing texts in sidenotes with \edtext, referring to the line number where the sidenote is called. ........................................... 1

v2.14.1.
General: Fix a bug when using \markboth on \doinsidelineLhook and \doinsidelineRhook ............................................................. 1
Fix a bug with vertical space before sectioning command in optional argument of \pstart (bug added in v2.13.1). ........................................ 1
v2.14.2.
  General: Fix spurious space with \edtextlater. .............................. 1
v2.14.3.
  General: Fix a bug in \edtextnow. ............................................ 1
v2.14.4.
  General: Take into account \linenumberstyle when using \edlineref. .... 1
v2.15.0.
  General: Add movecolumnpositiononrightpage option. ....................... 1
  Fix a bug about space before sectioning commands after a pairs environment (bug added in v. 2.14.4). ........................................ 1
  Fix a bug with \msdata when using multiple \beginnumbering...\endnumbering 1
  Fix a bug with page break when using \columns (bug added in v. 2.14.1). .... 1
  New macro \sidetotemarginr, fix default value to right. .................. 1
v2.16.0.
  General: Compatibility with \setmsdataposition .............................. 1
v2.16.1.
  General: Restore sidenotes on left side, deleted by mistake in v2.16.0 .... 1
v2.16.2.
  General: Change log message when numbered files still don’t exist, in order to improve compatibility with latexmk. ............................ 1
v2.16.3.
  General: Fix a bug when setting \aftercolumnseparator. ...................... 1
v2.17.0.
  General: Change code for compatibility with reledmac 2.18.0 ................ 1
v2.17.1.
  General: Fix a bug with prevpgstyle option when not using prevpgnotnumbered. .... 1
v2.17.2.
  \Pages: Fix a bug when changing \Lcolwidtth and \Rcolwidtth in pages environment. .................................................. 114
v2.17.3.
  General: Fix a bug when using multiple sidenotes on the same line, on right side. .... 1
v2.17.4.
  General: Fix bugs with sublines in parallel typesetting. ..................... 1
v2.18.0.
  General: Compatibility with v2.20.0 of reledmac. .......................... 1
v2.19.0.
  General: Compatibility with v2.22.0 of reledmac. .......................... 1
v2.19.1.
  General: Fix a bug with \edtext in sectioning command with shiftedpstarts mode. .... 1
v2.19.2.
  General: \continuousnumberingwithcolumns also works for the pstart number. .... 1
v2.19.3.
  General: Do not print footnotes at the first run. ........................... 1
  Fix a bug with nopbinverse when using consecutive \Pages. ................... 1
v2.19.4.
  General: Optimisation of the code in order to not print footnotes at the first run. .... 1
v2.19.5.
  General: Fix a bug added in astanza in v. 2.18.0 ............................ 1
v2.20.0.
\next\line\thing\@stuff: Add \next\line\thing\@stuff internal hook. ........ 53
General: Add \linenum\only\Pages\for\Columns and
\linenum\only\Pages\for\Columns to tell to print the line numbers associated with
a column only on some specific pages (for example, only on right pages). ........ 1
Add sidenotes\margin\page option to make the sidenotes refer to page margins and
not column margins. .......................................................... 1
Fix a bug with \continuous\numbering\with\columns\when\using\multiple\series\of
continued numbering text. Now, the line number are correctly reset at each
\begin\numbering. ................................................................. 1
Fix a bug with \continuous\numbering\with\columns\and\inner/outer\notes\and\line
numbering. ................................................................. 1
Fix a bug with \continuous\numbering\with\columns\when\a \begin\numbering is
called after another \begin\numbering...\end\numbering. ........ 1
Fix a bug with \width\like\two\columns\when\manually\setting
\before\column\separator\and\after\column\separator. ........ 1
Fix compatibility between \continuous\numbering\with\columns\and
\numberp\startfalse. ....................................................... 1
Fix incompatibility between \line\ation\{\page\}\and
\continuous\numbering\with\columns\option ................................ 1
v2.20.1.
General: Fix bug with \continuous\numbering\with\columns\when\left\column\and
right\column\have\not\the\same\size. ........................................ 1
v2.20.2.
General: Fix bugs with \continuous\numbering\with\columns\when\going\to\a\new\page 1
v2.21.0.
General: Add compatibility with new features of reledmac 2.26.0 .............. 1
v2.21.1.
General: Fix bug with \linenumber\annotation ................................ 1
v2.21.2.
General: Fix bug with \xp\start\and\ed\text\later ................................ 1
v2.21.3.
General: Fix bug when using \numberp\startfalse and alternating between manual
\p\start\and\automatic\p\start produced by \autopar .................. 1
v2.21.4.
General: Fix bug with using \eledsection and related when alternate normal
typesetting and parallel typesetting ...................................... 1
v2.21.5.
General: Compatibility with reledmac v. 2.26.6 ................................ 1
v2.21.6.
General: Fix page number in footnote and endnotes when using
\sameparallel\pagenumber\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{EPiX2018/04/01 patch level 3} to generate the \texttt{.sty} file from a \texttt{.ins} file ................................. 1
v2.21.9.
  General: Fix (little) bug when using \texttt{calc} package. ......................... 1
v2.22.0.
  General: Compatibility with \texttt{\textbackslash doinsidethislinehook} macro. ........ 1
v2.22.1.
  General: Really reset line number annotation at each line ......................... 1
v2.22.2.
  General: Fix bug with \texttt{\textbackslash onlyside} and \texttt{\textbackslash 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{\textbackslash theledlanguageR}: Provide compatibility with the latest version of polyglossia (v1.45). 98
v2.23.0.
  \texttt{\textbackslash theledlanguageR}: Restore compatibility with versions of polyglossia prior to v1.45. . 98
  General: Add \texttt{\textbackslash Leftpagehook} and \texttt{\textbackslash 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 startstanza\textbackslash indent\textbackslash repetition feature. ......................... 1
v2.24.1.
  General: Fix incompatibility between lineation by \texttt{\pstart} and (a)stanza. .......... 1
  Makes \texttt{\numberstantzatrue} 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{\textbackslash linenumannotationothersidetrue} of \texttt{reledmac}. . 1