

Parallel typesetting for critical editions: the `reledpar` package*

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

Abstract

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

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

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

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

Contents

1 Introduction	5
1.1 Aim of this package	5
1.2 Historical overview	6
2 Options	6
2.1 Synchronization’s options	6
2.2 Other options	6
3 General	7

*This file (`reledpar.dtx`) has version number v2.20.0, last revised 2017/08/17.

[†]maieul at maieul dot net

[‡]herries dot press at earthlink dot net

4 Parallel columns	8
4.1 Basic use	8
4.2 Setting	9
4.2.1 Column's width	9
4.2.2 Column's separator	9
4.2.3 Column's positions	9
4.2.4 Mixing two columns and one column texts	10
4.2.5 Schemas of column typesetting	10
4.3 Note about <code>\AtEveryPstart*</code> and <code>\AtEveryPend*</code>	10
5 Keeping translation in the outside/inside column	11
6 Facing pages	11
6.1 Basic usage	11
6.2 Setting	11
6.2.1 Text width	11
6.2.2 Way of synchronizing	11
6.2.3 Page number	13
6.2.4 Page breaking	13
6.2.5 Right page before <code>\Pages</code>	13
6.2.6 Notes about <code>\mainmatter</code>	14
6.3 Critical and familiar footnotes	14
6.3.1 Notes height setting	14
6.3.2 About the numbering of familiar footnotes	14
6.3.3 Using <code>perpage</code> package	15
6.3.4 Notes for one side only	15
6.3.5 Familiar notes called on the right side, but to be printed on the left side	15
6.4 Critical notes called on leftside but printed on right side	16
6.5 Using line flag	16
7 Left and right texts	16
7.1 Environments	16
7.2 Numbering text lines and paragraphs	16
7.3 First line number and line number increment	18
7.4 Lineation system	18
7.5 Line number style	18
7.6 Line number margin	18
7.7 Line flag	19
7.8 Chunks	20
7.9 <code>\AtEveryPstart</code> and <code>\AtEveryPstartCall</code>	20
7.10 Language setting	20
7.11 Executing code at each line	21
8 Verse	21

<i>Contents</i>	3
9 Side notes	22
10 Parallel ledgroups	22
10.1 General	22
10.2 Parallel ledgroups and <code>setspace</code> package	23
11 Sectioning commands	23
12 Notes about page number	23
I Implementation overview	25
II Preliminaries	25
II.1 Package's meta-data	25
II.2 Package's requirement	25
II.3 Package's options	25
II.4 Package's options	26
II.4.1 Synchronization's options	26
II.4.2 Position options	27
II.4.3 Other options	27
II.5 Determining side and category of parallel processing	28
II.6 Text's width	28
II.7 Messages	28
II.8 Naming macros	31
III Sectioning commands	32
IV Line counting	37
IV.1 Setting lineation reset	37
IV.2 Setting line number margin	38
IV.3 Setting lineation start and step	40
IV.4 Setting line flag	41
IV.5 Setting line number style	41
IV.6 Print marginal line number	42
IV.7 Line-number counters and lists	42
IV.7.1 Correspond to those in <code>reledmac</code> for regular or left text	42
IV.7.2 Specific to <code>reledpar</code>	43
IV.8 Reading the line-list file	43
IV.9 Commands within the line-list file	44
IV.10 Writing to the line-list file	50
V Marking text for notes	52
V.1 Specific hooks and commands for notes	52
V.1.1 Notes to be printed on one side only	53
V.1.2 Tools specific to familiar footnotes	53
V.1.3 Get correct footnote number	55
V.2 Create hooks	55

V.3 Init standards series (A,B,C,D,E,Z)	56
V.4 Tools specific to \LaTeX 's classical footnotes	56
VI Pstart numbers dumping and restoration	56
VII Parallel environments	57
VIII Paragraph decomposition and reassembly	60
VIII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code>	60
VIII.2 Processing one line	67
VIII.3 Line and page number computation	72
VIII.4 Line number printing	75
VIII.5 Pstart number printing in side	78
VIII.6 Add insertions to the vertical list	80
VIII.7 Penalties	80
VIII.8 Printing leftover notes	81
IX Footnotes	82
IX.1 Footnotes output specific to <code>\Pages</code>	82
IX.2 Critical footnote printed in right side	85
X Cross referencing	87
XI Side notes	88
XII Verse	90
XIII Fixing babel and polyglossia	93
XIV Counts and boxes for parallel texts	95
XV Checking text to be processed	97
XVI Parallel columns	98
XVII Parallel pages	109
XVII.1 Specific counters	109
XVII.2 Main macro	110
XVII.3 Ensure all notes are printed at the end of parallel pages	116
XVII.4 Struts	117
XVII.5 Page clearing	117
XVII.6 Lines managing	118
XVII.7 Page break managing	120
XVII.8 Getting boxes content	123
XVIII Page numbering	127
XVIII.1 Global options	127
XVIII.2 mainmatter option of <code>\Pages</code>	129

XIX Sections' titles' commands	129
XX Page break/no page break, depending on the specific line	130
XXI Parallel ledgroup	131
XXII Compatibility with eledmac	135
XXIII The End	135
Appendix A Some things to do when changing version	136
Appendix A.1 Migration to eledpar 1.4.3	136
Appendix A.2 Migration from eledpar to reledpar	136
Appendix A.2.1 Deprecated options	136
Appendix A.2.2 \renewcommand replaced with command	136
Appendix A.2.3 Commands the names of which have changed	137
Appendix A.3 Migration to reledpar 2.2.0	137
Appendix A.4 Migration to reledpar 2.3.0	137
Appendix A.5 Migration to reledpar 2.4.0	137
Appendix A.6 Migration to reledpar 2.5.0	137
Appendix A.7 Migration to reledpar 2.6.0	137
Appendix A.8 Migration to reledpar 2.6.1	137
Appendix A.9 Migration to reledpar 2.14.0	138
Appendix A.10 Migration to reledpar 2.14.1	138
Appendix A.11 Migration to v. 2.17.2	138
References	138
Index	138
Change History	159

1 Introduction

1.1 Aim of this package

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

The package has to try and coerce \TeX into paths it was not designed for. Use of the package, therefore, may produce some surprising results. In this case, please reports them to the author via github's issues: <https://github.com/maieul/ledmac/issues/>.

This manual contains a general description of how to use `reledpar` starting in section 3; the complete source code for the package, with extensive documentation (in

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

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

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

1.2 Historical overview

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

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

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

2 Options

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

2.1 Synchronization's options

Please read the paragraph on synchronization's option on 6.2.2 p. 11 to understand better those options.

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

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

nomaxlines allows facing pages to have different numbers of lines.

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

2.2 Other options

parledgroup allows the use of `ledgroup` environment with `reledpar`.¹

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

widthliketwocolumns set the width of the text printed in a single column to be the same as the width of the text printed in two parallel columns with `reledpar`. This is useful when alternating between normal and parallel typesetting.²

continuousnumberingwithcolumns allow you to alternate between normal typesetting and parallel column typesetting without resetting the line number (see 4.2.4 p. 10).

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

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

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

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

3 General

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

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

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

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

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

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

`\maxchunks`

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

```
\maxchunks{5120}
```

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

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

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

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

4 Parallel columns

4.1 Basic use

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

`\Columns`

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

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

`\AtBeginPairs`

Keep in mind that the `\Columns` **must be** outside of the `pairs` environment. You

can use the macro `\AtBeginPairs` to insert a code at the beginning of each pairs environments. That could be useful to add the `\sloppy` macro to prevent overfull hboxes in two columns.

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

There is no required pagebreak before or after the columns.

4.2 Setting

4.2.1 Column's width

`\Lcolwidth` `\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

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

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

`\beforecolumnseparator` `\aftercolumnseparator` By default, the spaces around column separator are the same as the space:

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

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

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

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

4.2.4 Mixing two columns and one column texts

`\widthliketwocolumns`

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

`\Xnoteswidthliketwocolumns`

`\notesXwidthliketwocolumns`

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

If you want to have continuous line numbers between multiple columns and single columns, use the `continuousnumberingwithcolumns` option when loading `reledmac` or `reledpar`. You will need to use `\pausenumbering... \resumenumbering` instead of `\endnumbering...endnumbering` (see 5.2.7 p. 20).

The first time you start **a left column or a single column**, use `\beginnumbering`, then use `\resumenumbering`. When you stop a left or a single column, use `\pausenumbering`, except the last time, where you must use `\endnumbering`.

The first time you start **a right column**, use `\beginnumbering`, then use `\resumenumbering`. When you stop a right column, use `\pausenumbering`, except the last time, where you must use `\endnumbering`.

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`.

4.3 Note about `\AtEveryPstart*` and `\AtEveryPend*`

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

The content of `\AtEveryPstart*` / `\AtEveryPend*` (with star) is added before every `\pstart` / after every `\pend`, but there is no distinction between left and right sides.

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 two times the vertical spacing, vertical spacing will be twice as large as expected. A solution is to add a test inside to make distinction between columns parallel typesetting / page parallel typesetting.

```
\AtEveryPstart*{%
\ifl@dprintingcolumns
  \vspace{0.125\baselineskip}
\else
  \vspace{0.25\baselineskip}
\fi
}
```

5 Keeping translation in the outside/inside column

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

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

6 Facing pages

6.1 Basic usage

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

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

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

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

6.2 Setting

6.2.1 Text width

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

`\Rcolwidth`

6.2.2 Way of synchronizing⁴

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

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

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

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

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

The aim of the default setting is twofold:

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

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

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

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

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

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

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

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

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

6.2.3 Page number

By default, `\Pages` use the standard \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 `sameparallelpagenumbers` to have the same page number for both left and right side. In this case, this setting will apply only for pages typeset by `\Pages`, not for “normal” pages.

Please also read advising in 12 p. 23.

6.2.4 Page breaking

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

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

6.2.5 Right page before `\Pages`

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

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

prevpgstyle=*⟨style⟩* in order to set the style of this page. A common value of *⟨style⟩* is empty. Use **prevpgstyle**=empty will suppress header and footer in this page. Please also read advising in 12 p. 23.

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

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

So, you MUST change the default setting via `\Xmaxhnotes` (for critical notes) and `\maxhnotesX` (for familiar notes). Both commands are explained in `reledmac` handbook (7.13.6 p. 49). As an advisable setting:

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

6.3.2 About the numbering of familiar footnotes

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

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

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

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

6.3.3 Using perpage package

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

So do not set:

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

But set:

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

6.3.4 Notes for one side only

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

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

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

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

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

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

```
\begin{Leftside}
\beginnumbering
\pstart
```

```

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

```

6.4 Critical notes called on leftside but printed on right side

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

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

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

6.5 Using line flag

`\Xlineflag` Use `\Xlineflag[<s>]` to add right line flag (7.7 p. 19) to right critical footnotes and
`\Xendlineflag` `\Xendlineflag[<s>]` to add it to right critical endnotes.

7 Left and right texts

7.1 Environments

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

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

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

`\endnumbering`, like:

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

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

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

`\memorydump`

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

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

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`.

`\numberpstarttrue`
`\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 Left side.
```

`\thepstartL`
`\thepstartR`
`\skipnumbering`

`\hidenumbering`

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

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

7.3 First line number and line number increment

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

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

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

`\firstlinenum*`
`\linenumincrement*`
`\firstsublinenum*`
`\sublinenumincrement*`
`\firstlinenumR`
`\linenumincrementR`
`\firstsublinenumR`
`\sublinenumincrementR`
`\lineationR`
`\lineation*`

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. 21) 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`
`\sublinenumberstyleR`
`\linenumberstyle*`
`\sublinenumberstyle*`

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

7.6 Line number margin

`\linenummarginR`
`\linenummargin*`

`\linenummarginColumns`
`\linenummarginColumnsR`
`\linenummarginColumns*`

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

Suppose you typeset parallel texts both in parallel pages and in parallel columns. In

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

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

`\linenumOnlyPagesForColumns`
`\linenumOnlyPagesForColumnsR`
 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 to get line numbers only on the inner margin, you need to get line number 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 the two cases, you need, for one column, to have line numbers only on one page of a double pages.

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).

7.7 Line flag

`\setRlineflag`
`\setRlineflag{<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.

`\linenumberLevenifblanktrue`
`\linenumberRevenifblanktrue`
 By default, when a blank line is printed on one side, in order to synchronize with the other side, no line number is printed. However, you can decide to print them for blank lines, also. Use `\linenumberLevenifblanktrue` to enable it on the left side, and `\linenumberRevenifblanktrue` to enable it on right side.

7.8 Chunks

`\pstart` `\pend` In a serial (non-parallel) mode, each numbered paragraph, or chunk, is contained between the `\pstart` and `\pend` macros, and the paragraph is output when the `\pend` macro occurs. The situation is somewhat different with parallel typesetting as the left text (contained within `\pstart` and `\pend` groups within the `Leftside` environment) has to be set in parallel with the right text (contained within its own `\pstart` and `\pend` groups within the corresponding `Rightside` environment) the `\pend` macros cannot immediately initiate any typesetting — this has to be controlled by the `\Columns` or `\Pages` macros. Several chunks may be specified within a `Leftside` or `Rightside` environment. A multi-chunk text then looks like:

```
\begin{...side}
  % \beginnumbering
  \pstart first chunk \pend
  \pstart second chunk \pend
  ...
  \pstart last chunk \pend
  % \endnumbering
\end{...side}
```

Numbering, via `\beginnumbering` and `\endnumbering`, may extend across several `Leftside` or `Rightside` environments. Remember, though, that the left/right sides are effectively independent of each other.

`\autopar` The `\autopar` macro can be used, instead of manually inserting `\pstart... \pends`. Please read `reledmac`'s handbook (5.2.2 p. 18).

7.9 `\AtEveryPstart` and `\AtEveryPstartCall`

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

Consequently:

- The argument of `\AtEveryPstart` (see 5.2.4 p. 19) is called before every chunk is printed, except if you used an optional argument for the `\pstart`.
- The argument of `\AtEveryPstartCall` is called before every `\pstart`.

7.10 Language setting

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

language selection for the side. The right side language is also applied to the right side line numbers.

7.11 Executing code at each line

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

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

About these two hooks, read `reledmac`'s handbook (5.5 p. 24).

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

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

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

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

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

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

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

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

`\thestanzaL` When using `\stanzanumtrue` (9.9 p. 54) 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.

`\thestanzaR`

9 Side notes

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

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. 55). To do it you have:

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

See the following example:

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

```

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

```

10.2 Parallel ledgroups and setspace package

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

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

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

11 Sectioning commands

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

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

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

12 Notes about page number

If you use `sameparallepagenumber` option (6.2.3 p. 13 or `prevpgnotnumbered` option (6.2.5 p. 13), please read the following paragraph if you want to manipulate page numbers manually.

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

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

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

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

I Implementation overview

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

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

`reledpar`’s solution to the problem of parallel texts is to put the two texts into separate boxes, and then appropriately extract the pairs of lines from the boxes. This involves duplicating much of the original box code for an extra right text box. The other, smaller, part of the code is concerned with coordinating the line extractions from the boxes.

II Preliminaries

II.1 Package’s meta-data

Announce the name and version of the package, which is targeted for $\LaTeX 2\epsilon$. The package also requires the `reledmac` package, however we do not load it automatically, because we prefer users to know it.

```

1 %<*code>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledpar}[2017/08/17 v2.20.0 reledmac extension for
  parallel texts]%
4
5 %
```

II.2 Package’s requirement

Few commands use `\xspace` command.

```

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

II.3 Package’s options

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

```

8 \RequirePackage{xkeyval}
9 %
```

II.4 Package's options

II.4.1 Synchronization's options

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

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

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

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

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

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

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

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

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

```

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

```

II.4.2 Position options

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

```

39 \newif\ifmovecolumnsonrightpage%
40 \DeclareOptionX{movecolumnsonrightpage}{%
41   \movecolumnsonrightpagetrue%
42 }%
43 %

```

`\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.

```

44 \newif\if@sidenotesmarginpage%
45 \DeclareOptionX{sidenotesmarginpage}{%
46   \@sidenotesmarginpagetrue%
47 }%
48 %

```

II.4.3 Other options

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

```

49 \DeclareOptionX{parledgroup}{\parledgrouptrue}
50 %

```

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

```

51 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
52 \DeclareOptionX{continuousnumberingwithcolumns}{\
continuousnumberingwithcolumnstrue}%
53 %

```

`\ifsameparallelpagenumber` Options related to page numbering
`\ifprevpgnotnumbered`

```

54 \newif\ifsameparallelpagenumber
55 \newif\ifprevpgnotnumbered
56 \DeclareOptionX{sameparallelpagenumber}{\sameparallelpagenumberttrue}
57 \DeclareOptionX{prevpgnotnumbered}{\prevpgnotnumberedtrue}
58 %

```

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

```

59 \DeclareOptionX{prevpgstyle}{\gdef\prevpgstyle{#1}}%
60 %

```

```

61 \ProcessOptionsX%
62 %

```

II.5 Determining side and category of parallel processing

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

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

II.6 Text’s width

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

```

63 \newdimen\Lcolwidth
64 \Lcolwidth=0.45\textwidth
65 \newdimen\Rcolwidth
66 \Rcolwidth=0.45\textwidth
67 \newdimen\Lcolwidth@pages
68 \newdimen\Rcolwidth@pages
69 %

```

II.7 Messages

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

```

\reledpar@error70 \newcommand{\reledpar@error}[2]{\PackageError{reledpar}{#1}{#2}}
71 %

```

```

\reledpar@warning72 \newcommand{\reledpar@warning}[1]{\PackageWarning{reledpar}{#1}}%
73 %

\led@err@TooManyPstarts74 \newcommand*{\led@err@TooManyPstarts}{%
75 \reledpar@error{Too many \string\pstart\space without printing.
76 Some text will be lost}{\@ehc}}
77 %

\led@err@BadLeftRightPstarts78 \newcommand*{\led@err@BadLeftRightPstarts}[2]{%
79 \reledpar@error{The numbers of left (#1) and right (#2)
80 \string\pstart s do not match}{\@ehc}}
81 %

\led@err@LeftOnRightPage82 \providebool{syntax@}
\led@err@RightOnLeftPage83 \newcommand*{\led@err@LeftOnRightPage}{%
84 \notbool{syntax@}%
85 {\reledpar@error{The left page has ended on a right page}{\@ehc}}%
86 {}%
87 }
88 \newcommand*{\led@err@RightOnLeftPage}{%
89 \notbool{syntax@}%
90 {\reledpar@error{The right page has ended on a left page}{\@ehc}}
91 {}%
92 }%
93 %

\led@err@Leftside@PreviousNotPrinted84 \newcommand*{\led@err@Leftside@PreviousNotPrinted}{%
\led@err@Rightside@PreviousNotPrinted85 \reledpar@error{You call a new Leftside environment while the previous
one has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}
86 \newcommand*{\led@err@Rightside@PreviousNotPrinted}{%
87 \reledpar@error{You call a new Rightside environment while the previous
one has not been typeset by \string\Pages\space or \string\Columns}{\@ehc}}
88 %

\led@err@Pages@InsideEnv99 \newcommand*{\led@err@Pages@InsideEnv}{%
\led@err@Columns@InsideEnv100 \reledpar@error{\string\Pages\space must be called *outside* of the `
pages` environment}{\@ehc}}
101 \newcommand*{\led@err@Columns@InsideEnv}{%
102 \reledpar@error{\string\Columns\space must be called *outside* of the `
pairs` environment}{\@ehc}}
103 %

```

```

\led@err@Pages@WithoutEnv04 \newcommand*\led@err@Pages@WithoutEnv}{%
\led@err@Columns@WithoutEnv05 \reledpar@error{\string\Pages\space called without previous `pages`
environment}{\@ehc}}
106 \newcommand*\led@err@Columns@WithoutEnv}{%
107 \reledpar@error{\string\Columns\space called without previous `pairs`
environment}{\@ehc}}
108 %

```

```

\led@err@Pages@WithoutRightside09 \newcommand*\led@err@Pages@WithoutRightside}{%
\led@err@Pages@WithoutLeftside10 \reledpar@error{\string\Pages\space called without any preceding `
\led@err@Columns@WithoutRightside Rightside` environment (or environment without content)}{\@ehc}}%
\led@err@Columns@WithoutLeftside11 \newcommand*\led@err@Pages@WithoutLeftside}{%
112 \reledpar@error{\string\Pages\space called without any preceding `
Leftside` environment (or environment without content)}{\@ehc}}%
113 \newcommand*\led@err@Columns@WithoutRightside}{%
114 \reledpar@error{\string\Columns\space called without any preceding `
Rightside` environment (or environment without content)}{\@ehc}}%
115 \newcommand*\led@err@Columns@WithoutLeftside}{%
116 \reledpar@error{\string\Columns\space called without any preceding `
Leftside` environment (or environment without content)}{\@ehc}}%
117 %

```

```

\led@error@fail@patch@thepage18 \newcommand\led@error@fail@patch@thepage}{%
119 \reledpar@error{Fail to patch \string\thepage\space command.}{\@ehc}%
120 }%
121 %

```

```

\led@error@fail@patch@pagenumbering22 \newcommand\led@error@fail@patch@pagenumbering}{%
123 \reledpar@error{Fail to patch \string\pagenumbering\space command.}{\@ehc
}%
124 }%
125 %

```

```

\led@error@note@called@onrightside26 \newcommand\led@error@note@called@onrightside}[1]{%
\led@error@note@called@onleftside27 \reledpar@error{#1 called on right side, despite your configuring it to
be for the left side only}{\@ehc}%
128 }%
129 \newcommand\led@error@note@called@onleftside}[1]{%
130 \reledpar@error{#1 called on left side, despite your configuring it to be
for the right side only}{\@ehc}%
131 }%
132 %

```

```

error@fail@patch@@mempnum33 \newcommand{\led@error@fail@patch@@mempnum}{%
134 \reledpar@error{Fail to patch \string\@mempnum\space command.}\@ehc}%
135 }%
136 %

or@fail@patch@@outputpage37 \newcommand{\led@error@fail@patch@@outputpage}{%
138 \reledpar@error{Fail to patch \string\@outputpage\space command.}\@ehc}%
139 }%
140 %

ed@error@edtext@later@now41 \newcommand{\led@error@edtext@later@now}{%
142 \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?)
143 }%
144 %

led@warn@ChangeSyncOption45 \newcommand*{\led@warn@ChangeSyncOption}[1]{%
146 \reledpar@warning{You have changed synchronization's options since the
last run. We have not read line-list file #1. Please run LaTeX again.}%
147 }%
148 %

warn@setting@in@rightside49 \newcommand{\led@warn@setting@in@rightside}[1]{%
150 \reledpar@warning{You use #1 inside rightside environment.\MessageBreak%
151 Such behavior is deprecated.\MessageBreak%
152 Use instead #1R or #1* in your preamble.}%
153 }
154 %

d@error@missing@numbering55 \newcommand{\led@error@missing@numbering}[1]{%
156 \reledpar@error{Missing \string\...pstart\string\pend\space inside `#1`
environment}\@ehc}%
157 }%
158 %

```

II.8 Naming macros

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

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

```

159 \providecommand*\newnamebox}[1]{%
160   \expandafter\newbox\csname #1\endcsname}
161 \providecommand*\setnamebox}[1]{%
162   \expandafter\setbox\csname #1\endcsname}
163 \providecommand*\unhnamebox}[1]{%
164   \expandafter\unhbox\csname #1\endcsname}
165 \providecommand*\unvnamebox}[1]{%
166   \expandafter\unvbox\csname #1\endcsname}
167 \providecommand*\namebox}[1]{%
168   \csname #1\endcsname}
169
170 %

```

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

```

171 \newcommand*\ifcsboxvoid}[1]{%
172   \ifcsname#1\endcsname%
173   \expandafter\ifvoid\csname #1\endcsname%
174   \expandafter\expandafter%
175   \expandafter\@firstoftwo%
176   \else%
177   \expandafter\expandafter%
178   \expandafter\@secondoftwo%
179   \fi%
180   \else%
181   \expandafter\@firstoftwo%
182   \fi%
183 }%
184 %

```

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

\usernamecount

```

185 \providecommand*\newnamecount}[1]{%
186   \expandafter\newcount\csname #1\endcsname}
187 \providecommand*\usernamecount}[1]{%
188   \csname #1\endcsname}
189
190 %

```

III Sectioning commands

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

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


```

191 \newcount\section@numR
192 \section@numR=\z@
193 %

```

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

```

194 \pst@rtedLfalse
195 \newif\ifpst@rtedR
196
197 %

```

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

```

198 \newcommand*{\beginnumberingR}{%
199 \ifnumberingR
200 \led@err@NumberingStarted
201 \endnumberingR
202 \fi
203 \global\l@dnumpstartsR \z@
204 \global\pst@rtedRfalse
205 \global\numberingRtrue
206 \global\advance\section@numR \@ne
207 \global\absline@numR \z@
208 \gdef\normal@page@breakR{ }
209 \gdef\l@prev@pbR{ }
210 \gdef\l@prev@nopbR{ }
211 \global\line@numR \z@
212 \global\@lockR \z@
213 \global\sub@lockR \z@
214 \global\sublines@Rfalse
215 \global\let\next@page@numR\relax
216 \global\let\sub@change\relax
217 \global\last@page@numR=-10000%
218 \global\stopmsdata@inserted@true%
219 \global\let\@msdata@list\relax%
220 \global\csundef{@msdata@add@msd@cR @dataR}%
221 \ifcontinuousnumberingwithcolumns%
222 \ifnumberpstart%
223 \addtocounter{pstartR}{1}%
224 \fi%
225 \else%
226 \setcounter{pstartR}{1}%
227 \fi%
228 \message{Section \the\section@numR R }%
229 \set@continuousnumberingforR%
230 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
231 \l@dend@stuff

```

```

232 \begingroup
233 \initnumbering@sectcountR
234 \gdef\eled@sectionsR@{ }%
235 \if@noeled@sec\else%
236   \makeatletter%
237   \InputIfFileExists%
238   {\l@auxdir\jobname.eledsec\the\section@numR R}%
239   {}%
240   {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@numR R}}%
241   \makeatother%
242   \immediate\openout\eled@sectioningR@out=\l@auxdir\jobname.eledsec\the\
section@numR R\relax%
243 \fi%
244 }
245 %

```

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

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

```

246 \def\endnumberingR{%
247   \ifnumberingR
248     \global\numberingRfalse
249     \normal@pars
250     \ifnum\l@dnumpstartsR=0%
251       \led@err@NumberingWithoutPstart%
252     \fi%
253     \ifl@dpairing
254       \global\pst@rtedRfalse
255     \else
256       \ifx\insertlines@listR\empty\else
257         \global\noteschanged@true
258       \fi
259       \ifx\line@listR\empty\else
260         \global\noteschanged@true
261       \fi
262     \fi
263     \ifnoteschanged@
264       \led@mess@NotesChanged
265     \fi
266   \else
267     \led@err@NumberingNotStarted
268   \fi
269 \endgroup
270 \if@noeled@sec\else%
271   \immediate\closeout\eled@sectioningR@out%
272 \fi%

```

```

273 }
274
275 %

```

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

`\save@section@number`

`\set@sectcountR`

```

276 \newcounter{chapterR}
277 \newcounter{sectionR}
278 \newcounter{subsectionR}
279 \newcounter{subsubsectionR}
280
281 \newcount\old@chapter%
282 \newcount\old@section%
283 \newcount\old@subsection%
284 \newcount\old@subsubsection%
285 \newcommand{\save@section@number}{%
286   \ifdefined\c@chapter%
287     \global\old@chapter\value{chapter}%
288   \fi%
289   \global\old@section\value{section}%
290   \global\old@subsection\value{subsection}%
291   \global\old@subsubsection\value{subsubsection}%
292 }%
293 \newcommand{\initnumbering@sectcountR}{
294   \ifdefined\c@chapter%
295     \setcounter{chapterR}{\old@chapter}%
296   \fi%
297   \setcounter{sectionR}{\old@section}%
298   \setcounter{subsectionR}{\old@subsection}%
299   \setcounter{subsubsectionR}{\old@subsubsection}%
300   \set@sectcountR%
301 }
302 \newcommand{\set@sectcountR}{%
303   \let\c@chapter\c@chapterR%
304   \let\c@section\c@sectionR%
305   \let\c@subsection\c@subsectionR%
306   \let\c@subsubsection\c@subsubsectionR%
307 }%
308 %

```

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

`\resumenumberingR`

`\ifresumenumberingR@start`

`\pausenumbering@page@numR`

```

309 \newcount\pausenumbering@page@numR%
310 \newcommand*{\pausenumberingR}{%
311   \ifx\next@page@numR\relax%
312     \global\pausenumbering@page@numR=0%
313   \else%
314     \global\pausenumbering@page@numR=\next@page@numR%

```

```

315 \fi%
316 \endnumberingR\global\numberingRtrue}
317 \newif\ifresumenumberingR@start%
318 \newcommand*{\resumenumberingR}{%
319 \ifnumberingR
320 \global\pst@rtedRtrue
321 \global\advance\section@numR \@ne
322 \global\resumenumberingR@starttrue%
323 \led@mess@SectionContinued{\the\section@numR R}%
324 \set@continuousnumberingforR%
325 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
326 \l@dend@stuff
327 \begingroup%
328 \initnumbering@sectcountR%
329 \else
330 \led@err@NumberingShouldHaveStarted
331 \endnumberingR
332 \beginnumberingR
333 \fi}
334
335 %

```

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

```

336 \newcommand*{\memorydumpL}{%
337 \endnumbering
338 \numberingtrue
339 \global\pst@rtedLtrue
340 \global\advance\section@num \@ne
341 \led@mess@SectionContinued{\the\section@num}%
342 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
343 \l@dend@stuff}
344
345 \newcommand*{\memorydumpR}{%
346 \endnumberingR
347 \numberingRtrue
348 \global\pst@rtedRtrue
349 \global\advance\section@numR \@ne
350 \led@mess@SectionContinued{\the\section@numR R}%
351 \line@list@stuffR{\jobname.\extensionchars\the\section@numR R}%
352 \l@dend@stuff}
353
354 %

```

IV Line counting

IV.1 Setting lineation reset

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

`\lineationR` `\lineationR{<word>}` is the macro used to select the lineation system for right texts. Its argument is a string: either `page`, `pstart` or `section`.

```

355 \newcommand*{\lineationR}[1]{%
356   \ifnumbering
357     \led@err@LineationInNumbered
358   \else
359     \def\@tempa{#1}\def\@tempb{page}%
360     \ifx\@tempa\@tempb
361       \global\bypage@Rtrue
362       \global\bypstart@Rfalse
363       \unless\ifnocritical@%
364         \Xpstart[] [false]%
365       \fi%
366     \else
367       \def\@tempb{pstart}%
368       \ifx\@tempa\@tempb
369         \global\bypage@Rfalse
370         \global\bypstart@Rtrue
371         \unless\ifnocritical@%
372           \Xpstart%
373         \fi%
374       \else
375         \def\@tempb{section}
376         \ifx\@tempa\@tempb
377           \global\bypage@Rfalse%
378           \global\bypstart@Rfalse%
379           \unless\ifnocritical@%
380             \Xpstart[] [false]%
381           \fi%
382         \else
383           \led@warn@BadLineation
384         \fi%
385       \fi
386     \fi
387   \fi}}
388 %

```

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

```

389 \newcommand{\set@continuousnumberingforR}{%
390   \ifcontinuousnumberingwithcolumns%
391     \ifnum\line@numR<\line@num%
392       \global\appto\next@line@list@stuffR{%
393         \expandafter\setlinenum\expandafter{\the\line@num}%
394       }%
395     \fi%
396     \ifnum\last@page@num>\last@page@numR%
397       \global\last@page@numR=\last@page@num%
398     \fi%
399   \fi%
400 }
401 %

```

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

```

402 \WithSuffix\newcommand\lineation*[1]{%
403   \lineation{#1}%
404   \lineationR{#1}%
405 }%
406 %

```

IV.2 Setting line number margin

\linenummargin You call `\linenummargin{<word>}` to specify which margin you want your right text's line numbers in; it takes one argument, a string. You can put the line numbers in the same margin on every page using `left` or `right`; or you can use `inner` or `outer` to get them in the inner or outer margins. You can change this within a numbered section, but the change may not take effect just when you would like; if it is done between paragraphs nothing surprising should happen.

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

It is defined only once time, in `reledmac`.

```

407 \newcount\line@marginR
408 %

```

By default put right text numbers at the right.

```

409 \line@marginR=\@ne
410
411 %

```

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

```

412 \newcommand{\linenummarginR}[1]{%
413   \l@getline@margin{#1}%
414   \ifnum\@l@dttempcntb>\m@ne%
415     \global\line@marginR=\@l@dttempcntb%

```

```

416 \fi%
417 }
418 \WithSuffix\newcommand\linenummargin*[1]{%
419 \l@getline@margin{#1}%
420 \ifnum\@l@tempcntb>\m@ne%
421 \global\line@marginR=\@l@tempcntb%
422 \global\line@margin=\@l@tempcntb%
423 \fi%
424 }
425 \ifmovecolumnsononrightpage%
426 \linenummargin{inner}%
427 \linenummarginR{outer}%
428 \fi%
429 %

```

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

```

\linenummarginColumns
\linenummarginColumnsR
430
431 \newcount\line@margin@columnsR%
432 \line@margin@columnsR=\m@ne%
433
434 \newcommand{\linenummarginColumns}[1]{%
435 \l@getline@margin{#1}%
436 \ifnum\@l@tempcntb>\m@ne%
437 \global\line@margin@columns=\@l@tempcntb%
438 \fi%
439 }%
440
441 \WithSuffix\newcommand\linenummarginColumns*[1]{%
442 \l@getline@margin{#1}%
443 \ifnum\@l@tempcntb>\m@ne%
444 \global\line@margin@columns=\@l@tempcntb%
445 \global\line@margin@columnsR=\@l@tempcntb%
446 \fi%
447 }%
448
449 \newcommand{\linenummarginColumnsR}[1]{%
450 \l@getline@margin{#1}%
451 \ifnum\@l@tempcntb>\m@ne%
452 \global\line@margin@columnsR=\@l@tempcntb%
453 \fi%
454 }%
455
456 %

```

`\linenumOnlyPagesForColumns` and `\linenumOnlyPagesForColumnsR` make the line numbers be printed only on left/right page for the left/right column.

```

457 \def\linenum@OnlyPages@ForColumns{%

```

```

458 \newcommand{\linenumOnlyPagesForColumns}[1]{%
459   \gdef\linenum@OnlyPages@ForColumns{#1}%
460 }%
461
462 \def\linenum@OnlyPages@ForColumnsR{%
463   \newcommand{\linenumOnlyPagesForColumnsR}[1]{%
464     \gdef\linenum@OnlyPages@ForColumnsR{#1}%
465   }%
466   %

```

IV.3 Setting lineation start and step

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

```

467 \newcounter{firstlinenumR}
468   \setcounter{firstlinenumR}{5}
469 \newcounter{linenumincrementR}
470   \setcounter{linenumincrementR}{5}
471 %

```

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

```

472 \newcounter{firstsublinenumR}
473   \setcounter{firstsublinenumR}{5}
474 \newcounter{sublinenumincrementR}
475   \setcounter{sublinenumincrementR}{5}
476
477 %

```

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

```

\firstlinenum
\linenumincrement
\firstsublinenum
\sublinenumincrement
\firstlinenum*
\linenumincrement*
\firstsublinenum*
\sublinenumincrement*
478 \WithSuffix\newcommand\firstlinenum*[1]{%
479   \setcounter{firstlinenumR}{#1}%
480   \setcounter{firstlinenum}{#1}%
481 }
482 \WithSuffix\newcommand\linenumincrement*[1]{%
483   \setcounter{linenumincrementR}{#1}%
484   \setcounter{linenumincrement}{#1}%
485 }
486 \WithSuffix\newcommand\firstsublinenum*[1]{%
487   \setcounter{firstsublinenumR}{#1}%
488   \setcounter{firstsublinenum}{#1}%
489 }

```



```

490 \WithSuffix\newcommand\sublinenumincrement*[1]{%
491   \setcounter{sublinenumincrementR}{#1}%
492   \setcounter{sublinenumincrement}{#1}%
493 }
494 %

```

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

```

\linenumincrementR
\firstsublinenumR
\sublinenumincrementR
495 \newcommand\firstlinenumR[1]{%
496   \setcounter{firstlinenumR}{#1}%
497 }
498 \newcommand\linenumincrementR[1]{%
499   \setcounter{linenumincrementR}{#1}%
500 }
501 \newcommand\firstsublinenumR[1]{%
502   \setcounter{subfirstlinenumR}{#1}%
503 }
504 \newcommand\sublinenumincrementR[1]{%
505   \setcounter{sublinenumincrementR}{#1}%
506 }
507 %

```

IV.4 Setting line flag

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

```

508 \newcommand{\setRlineflag}[1]{%
509   \gdef\Rlineflag{#1}%
510 }
511 \setRlineflag{R}
512 %

```

IV.5 Setting line number style

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

```

513 \newcommand*\linenumrepR[1]{\@arabic{#1}}
514 \newcommand*\sublinenumrepR[1]{\@arabic{#1}}
515
516 %

```

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

`\sublinenumberstyleR`

```

517 \newcommand*\linenumberstyleR[1]{%
518   \def\linenumrepR##1{\@nameuse{#1}{##1}}
519 \newcommand*\sublinenumberstyleR[1]{%
520   \def\sublinenumrepR##1{\@nameuse{#1}{##1}}
521 %

```

```

\linenumberstyle* And for both side.
\sublinenumberstyle*
522 \WithSuffix\newcommand\linenumberstyle*[1]{%
523   \linenumberstyle{#1}%
524   \linenumberstyleR{#1}%
525 }%
526
527 \WithSuffix\newcommand\sublinenumberstyle*[1]{%
528   \sublinenumberstyle{#1}%
529   \sublinenumberstyleR{#1}%
530 }%
531 %
532 %

```

IV.6 Print marginal line number

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

```

533 \newif\iflinenumberLevenifblank
534 \newif\iflinenumberRevenifblank
535 %

```

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

```

536 \newcommand*\leftlinenumR{%
537   \l@dlinenumR
538   \kern\linenumsep}
539 \newcommand*\rightlinenumR{%
540   \kern\linenumsep
541   \l@dlinenumR}
542 \newcommand*\l@dlinenumR{%
543   \numlabfont\linenumrepR{\line@numR}\@Rlineflag%
544   \ifsublines@R
545     \ifnum\subline@numR>\z@
546       \unskip\fullstop\sublinenumrepR{\subline@numR}%
547     \fi
548   \fi}
549
550 %

```

IV.7 Line-number counters and lists

IV.7.1 Correspond to those in `reledmac` for regular or left text

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

`\line@numR` The count `\line@numR` stores the line number that is used in the right text's marginal line numbering and in notes. The count `\subline@numR` stores a sub-line number that qualifies `\line@numR`. The count `\absline@numR` stores the absolute number of lines since the start of the right text section: that is, the number we have actually printed, no matter what numbers we attached to them.

The boolean `\ifsublines@R` is set to true when we start subline on rightside.

```

551 \newcount\line@numR
552 \newcount\subline@numR
553 \newcount\absline@numR
554 \newif\ifsublines@R%
555
556 %

```

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

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

```

557 \list@create{\line@listR}
558 \list@create{\insertlines@listR}
559 \list@create{\actionlines@listR}
560 \list@create{\actions@listR}
561
562 %

```

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

```

563 \newcount\page@numR
564
565 %

```

IV.7.2 Specific to `reledpar`

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

```

566 \list@create{\linesinpar@listL}
567 \list@create{\linesinpar@listR}
568 \list@create{\maxlinesinpar@list}
569
570 %

```

IV.8 Reading the line-list file

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

```

571 \newcommand{\list@clearing@regR}{%
572   \list@clear{\line@listR}%
573   \list@clear{\insertlines@listR}%
574   \list@clear{\actionlines@listR}%
575   \list@clear{\actions@listR}%
576   \list@clear{\linesinpar@listR}%
577   \list@clear{\linesonpage@listR}
578 }
579 %

```

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

```

580 \newcommand{\@par@sync@option}[1]{%
581   \IfStrEq{#1}{\@par@this@sync@option}%
582   {%
583     {\ifledRcol%
584       \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
585     }%
586     \else%
587       \led@warn@ChangeSyncOption{\jobname.\extensionchars\the\section@num}
588     }%
589     \fi%
590     \endinput%
591   }%
592 }%
593 %

```

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

IV.9 Commands within the line-list file

This section defines the commands that can appear within a line-list file, except for `\@lab` which is in a later section among the cross-referencing commands it is associated with.

The macros with `action` in their names contain all the code that modifies the action-code list.

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

```

592 \newcommand{\@nl@regR}{%
593   \ifx\l@dchset@num\relax \else
594     \advance\absline@numR \@ne
595     \csgdef{l@dchset@numR\the\absline@numR}{}%To remember this line have
596     been marked by a \setlinenum
597     \set@line@action

```

```

597 \let\l@dchset@num\relax
598 \advance\absline@numR \m@ne
599 \advance\line@numR \m@ne% % do we need this?
600 \fi
601 \advance\absline@numR \@ne
602 \ifx\next@page@numR\relax \else
603 \page@action
604 \let\next@page@numR\relax
605 \fi
606 \ifx\sub@change\relax \else
607 \ifnum\sub@change>\z@
608 \sublines@Rtrue
609 \else
610 \sublines@Rfalse
611 \fi
612 \sub@action
613 \let\sub@change\relax
614 \fi
615 \ifcase\@lockR
616 \or
617 \@lockR \tw@
618 \or\or
619 \@lockR \z@
620 \fi
621 \ifcase\sub@lockR
622 \or
623 \sub@lockR \tw@
624 \or\or
625 \sub@lockR \z@
626 \fi
627 \ifsublines@R
628 \ifnum\sub@lockR<\tw@
629 \advance\subline@numR \@ne
630 \fi
631 \else
632 \ifnum\@lockR<\tw@
633 \advance\line@numR \@ne \subline@numR \z@
634 \fi
635 \fi}
636
637
638 %

```

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

```

639 \newcount\last@page@numR
640 \last@page@numR=-10000
641
642 %

```

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

```

643 \newcount\@lockR
644 \newcount\sub@lockR
645
646 \newcommand*{\do@lockonR}{%
647   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
648   \ifsublines@R
649     \xright@appenditem{-1005}\to\actions@listR
650     \ifnum\sub@lockR=\z@
651       \sub@lockR \@ne
652     \else
653       \ifnum\sub@lockR=\thr@@
654         \sub@lockR \@ne
655       \fi
656     \fi
657   \else
658     \xright@appenditem{-1003}\to\actions@listR
659     \ifnum\@lockR=\z@
660       \@lockR \@ne
661     \else
662       \ifnum\@lockR=\thr@@
663         \@lockR \@ne
664       \fi
665     \fi
666   \fi}
667
668 %

```

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

```

669
670
671 \newcommand{\do@lockoffR}{%
672   \xright@appenditem{\the\absline@numR}\to\actionlines@listR
673   \ifsublines@R
674     \xright@appenditem{-1006}\to\actions@listR
675     \ifnum\sub@lockR=\tw@
676       \sub@lockR \thr@@
677     \else
678       \sub@lockR \z@
679     \fi
680   \else
681     \xright@appenditem{-1004}\to\actions@listR
682     \ifnum\@lockR=\tw@
683       \@lockR \thr@@
684     \else
685       \@lockR \z@
686     \fi
687   \fi}
688
689 %
690 %

```

`\n@num`

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

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

```

691 \newcount\insert@countR
692 %

```

- #2, a sequence of other line-list-file commands, executed to determine the ending line-number. This may also include other `\@ref` commands, corresponding to uses of `\edtext` within the first argument of another instance of `\edtext`.

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

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

```

693 \newcommand*{\@ref@regR}[2]{%
694   \global\advance\@edtext@level by 1%
695   \global\insert@countR=#1\relax
696   \loop\ifnum\insert@countR>\z@
697     \xright@appenditem{\the\absline@numR}\to\insertlines@listR
698     \global\advance\insert@countR \m@ne
699   \repeat
700 %

```

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.

```

701 \begingroup
702   \let\@ref=\dummy@ref
703   \let\@lopR\@gobble
704   \let\page@action=\relax
705   \let\sub@action=\relax
706   \let\set@line@action=\relax
707   \let\@lab=\relax
708   \let\@lemma=\relax
709   \let\@sw\@gobblethree%
710   #2
711   \global\endpage@num=\page@numR
712   \global\endline@num=\line@numR
713   \global\endsubline@num=\subline@numR
714 \endgroup
715 %

```

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

```

716 \xright@appenditem%
717   {\the\page@numR|\the\line@numR}%
718   \ifsublines@R \the\subline@numR \else 0\fi%
719   \the\endpage@num|\the\endline@num|%
720   \ifsublines@R \the\endsubline@num \else 0\fi}\to\line@listR
721 %

```

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

```

722 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
723   @edtext@level\endcsname}%
724 %

```

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

```

724 \@ref@reg@parseR{#2}%
725 %

```

Decrease edtext level counter.


```

726 \global\advance\@edtext@level by -1%
727 }
728 %

```

```

\@ref@reg@parseR 729 \newcommand{\@ref@reg@parseR}[1]{%
730 %

```

Declare and init boolean for lemma in this level.

```

731 \providebool{lemmacommand@the\@edtext@level}%
732 \boolfalse{lemmacommand@the\@edtext@level}%
733 %

```

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

```

734 #1%
735 % Now, we store the list of \protect\cs{sw} of this current \protect\cs{
edtext} as an element of
736 % the global list of list of \protect\cs{@sw} for a \protect\cs{edtext}
depth.
737 % \begin{macrocode}
738 \ifnum\@edtext@level>0%
739 \def\create@this@edtext@level{\expandafter\list@create\expandafter{
csname sw@list@edtextR@the\@edtext@level\endcsname}}%
740 \ifcsundef{sw@list@edtextR@the\@edtext@level}{\
create@this@edtext@level}{}%
741 \letcs{\@tmp}{sw@list@edtextR@the\@edtext@level}%
742 \letcs{\@tmpp}{sw@list@edtext@tmp@the\@edtext@level}%
743 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
744 \global\cslet{sw@list@edtextR@the\@edtext@level}{\@tmp}%
745 \fi%
746 }%
747 %

```

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

```

748 \renewcommand*{\@pend}[1]{%
749 \ifbypstart@global\line@num=0\fi%
750 \xright@appenditem{#1}\to\linesinpar@listL}
751 \renewcommand*{\@pendR}[1]{%
752 \ifbypstart@R\global\line@numR=0\fi
753 \xright@appenditem{#1}\to\linesinpar@listR}
754
755 %

```

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

```

756 \newcommand{\@pstart}[3]{%
757   \ifcsdef{minpage@pstart@#1}%
758     {\ifnumgreater{#2}{\csuse{minpage@pstart@#1}}%
759       {\csnumgdef{minpage@pstart@#1}{#2}}%
760     }%
761   }%
762   {\csnumgdef{minpage@pstart@#1}{#2}}
763   \csnumgdef{afterlines@pstart@#1L}{#3}%
764 }%
765
766 \newcommand{\@pstartR}[3]{%
767   \numdef{\@tmp}{#2-1}%Because we have not to know in which page the pstart
768   starts, but in which pair of facing page
769   \ifcsdef{minpage@pstart@#1}%
770     {\ifnumgreater{\@tmp}{\csuse{minpage@pstart@#1}}%
771       {\csnumgdef{minpage@pstart@#1}{\@tmp}}%
772     }%
773     {\csnumgdef{minpage@pstart@#1}{\@tmp}}
774   \csnumgdef{afterlines@pstart@#1R}{#3}%
775 }%
776 %

```

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

```

777 \renewcommand*{\@lopL}[1]{%
778   \xright@appenditem{#1}\to\linesonpage@listL}
779 \renewcommand*{\@lopR}[1]{%
780   \xright@appenditem{#1}\to\linesonpage@listR}
781
782 %

```

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`.

```

783 \newwrite\linenum@outR
784 %

```

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

`\first@linenum@out@Rtrue`

`\first@linenum@out@Rfalse`

```

785 \newif\iffirst@linenum@out@R
786 \first@linenum@out@Rtrue
787 %

```

`\line@list@stuffR` This is the right text version of the `\line@list@stuff{<file>}` macro. It is called by `\next@line@list@stuffR` 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 .

```

788
789 \let\next@line@list@stuffR\relax%
790 \newcommand*{\line@list@stuffR}[1]{%
791   \global\newtoggle{notfirststrun@#1}%
792   \IfFileExists{\l@auxdir#1}%
793     {\global\toggletrue{notfirststrun@#1}}%
794     {\global\togglefalse{notfirststrun@#1}}%
795   \read@linelist{#1}%
796   \iffirst@linenum@out@R
797     \global\first@linenum@out@Rfalse
798     \immediate\openout\linenum@outR=\l@auxdir#1%
799     \immediate\write\linenum@outR{\string\line@list@version{\
this@line@list@version}}%
800     \ifl@dpaging%
801       \immediate\write\linenum@outR{\string\@par@sync@option{\
@par@this@sync@option}}%
802     \fi%
803   \else
804     \if@minipage%
805       \leavevmode%
806     \fi%
807     \closeout\linenum@outR%
808     \openout\linenum@outR=\l@auxdir#1%
809   \fi%
810   \next@line@list@stuffR%
811   \global\let\next@line@list@stuffR\relax%
812 }%
813
814 %

```

`\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.

```

815 \newcommand*{\new@lineL}{%
816   \write\linenum@out{\string\@nl[\the\c@page][\thepage]}%
817 %

```

`\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.

```

818 \newcommand*{\new@lineR}{%
819   \write\linenum@outR{\string\@nl[\the\c@page][\thepage]}}
820 %

```

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

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

`\endsub`

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

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

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

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

`\endlock`

`\skipnumbering`

V Marking text for notes

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

```

\critext
\edtext
\set@line

```

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

V.1 Specific hooks and commands for notes

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

```

\newseries@par21 \newcommand{\newseries@par}[1]{%
822 %

```

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 temporarily the footnote not printed. We check the `nofamiliar` and `nocritical` `reledmac` options.

```

823 \unless\ifnofamiliar%
824 \csgdef{onlysideX@#1}{}%
825 \newnamebox{footins#1@kept}%
826 \fi%
827 \unless\ifnocritical%
828 \csgdef{Xonlyside@#1}{}%
829 \newnamebox{#1footins@kept}%
830 \fi%
831 %

```

V.1.2 Tools specific to familiar footnotes

```

832 \unless\ifnofamiliar%
833 %

```

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

So here, we only defined the new counter.

```

834 \newcounter{footnote#1@typeset}%
835 %

```

Familiar footnotes without marks The `\footnotexnomk` commands are for notes which are printed on the left side, while they are called in the right side. Basically, they set first toggle `\nomark@` to true, then call the `\footnotex`. and finally add the footnote counter in the footnote counter list.

First, check the `nofamiliar` option of `reledmac`.

So declare the list.

```
836 \expandafter\list@create\csname footnote#1@mk\endcsname%
837 %
```

Then, declare the `\footnotexnomk` command.

```
838 \expandafter\newcommand\csname footnote#1nomk\endcsname[1]{%
839 %
```

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

```
840 \toggletrue{nomk@}%
841 \csuse{footnote#1}{##1}%
842 \togglefalse{nomk@}%
843 %
```

Second, and last, step: store the footnote counter in the footnote counters list. We use some `\let`, because `\xright@appenditem` is difficult to use with `\expandafter`.

```
844 \letcs{\@tmp}{footnote#1@mk}%
845 \numdef\@tmpa{\csuse{c@footnote#1}}%
846 \global\xright@appenditem{\@tmpa}\to\@tmp%
847 \global\cslet{footnote#1@mk}{\@tmp}%
848 }%
849 %
```

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

```
850 \expandafter\newcommand\csname footnote#1mk\endcsname{%
851 %
```

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

```
852 \letcs{\@tmp}{footnote#1@mk}%
853 \gl@p\@tmp\to\@tmpa%
854 \global\cslet{footnote#1@mk}{\@tmp}%
855 %
```

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

```
856 \letcs{\old@footnote}{c@footnote#1}%
857 \setcounter{footnote#1}{\@tmpa}%
858 %
```

Define the footnote mark and print it

```

859 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
860 \csuse{@footnotemark#1}%
861 %

```

Restore previous footnote counter and finally add space.

```

862 \setcounter{footnote#1}{\old@footnote}%
863 \xspace%
864 }%
865 %

```

End of tools specific to familiar notes.

```

866 \fi
867 %

```

End of `\newseries@par`.

```

868 }%
869 %

```

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.

```

870 \newcommand{\save@familiarfootnote@number}{%
871 \unless\ifnofamiliar%
872 \def\do##1{\csxdef{saved@footnote##1}{\the\csname c@footnote##1\
endcsname}}%
873 \dolistloop{\@series}%
874 \fi%
875 \xdef\saved@footnote{\the\c@footnote}%
876 }
877 \newcommand{\get@familiarfootnote@number}{%
878 \unless\ifnofamiliar%
879 \def\do##1{\setcounter{footnote##1@typeset}{\csuse{saved@footnote##1}}%
880 %
881 \dolistloop{\@series}%
882 \fi%
883 \setcounter{footnote@typeset}{\saved@footnote}%
884 }
885 %

```

V.2 Create hooks

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

```

885 \unless\ifnocritical@%
886 \newhookcommand@series{Xonlyside}%
887 \fi%
888 \unless\ifnofamiliar@%
889 \newhookcommand@series{onlysideX}%
890 \fi
891
892
893 %

```

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

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

```

894 \newcommand{\init@series@par}{%
895 \def\do##1{\newseries@par{##1}}%
896 \dolistloop{\@series}%
897 }%
898 \init@series@par%
899 %

```

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. 53).

```

\footnote@reading900 \newcount\footnote@reading%
\footnote@typeset901 \newcounter{footnote@typeset}%
902 %

```

VI *Pstart numbers dumping and restoration*

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

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

```

\list@pstartL@pc903 \list@create{\list@pstartL@pc}%
\list@pstartR@pc904 \list@create{\list@pstartR@pc}%
905 %

```


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

```

\dump@pstartL@pc  \def\dump@pstartL@pc{%
\dump@pstartR@pc  \xright@appenditem{\the\c@pstartL}\to\list@pstartL@pc%
908               \global\cslet{numberpstart@L\the\l@dnumpstartsL}{\ifnumberpstart}%
909               }%
910
911 \def\dump@pstartR@pc{%
912   \xright@appenditem{\the\c@pstartR}\to\list@pstartR@pc%
913   \global\cslet{numberpstart@R\the\l@dnumpstartsR}{\ifnumberpstart}%
914   }%
915
916   %

```

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

```

\restore@pstartR@pc \def\restore@pstartL@pc{%
917 \def\restore@pstartL@pc{%
918   \ifx\list@pstartL@pc\empty\else%
919     \gl@p\list@pstartL@pc\to\@temp%
920     \global\c@pstartL=\@temp%
921   \fi%
922 }%
923 \def\restore@pstartR@pc{%
924   \ifx\list@pstartR@pc\empty\else%
925     \gl@p\list@pstartR@pc\to\@temp%
926     \global\c@pstartR=\@temp%
927   \fi%
928 }%
929 %

```

VII Parallel environments

The initial set up for parallel processing is deceptively simple.

pairs pages

chapterinpages The pairs environment is for parallel columns and the pages environment for parallel pages.

```

930 \newenvironment{pairs}{%}
931   \l@dpairingtrue
932   \l@dpagingfalse
933   \initnumbering@quote
934   \save@familiarfootnote@number%
935   \if@ledgroup%
936     \get@familiarfootnote@number%
937   \fi%

```

```

938 \save@section@number%
939 \at@begin@pairs%
940 }{%
941 \l@dpairingfalse
942 }
943
944 %

```

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

```

945 \newcommand{\AtBeginPairs}[1]{\gdef\at@begin@pairs{#1}}%
946 \def\at@begin@pairs{}%
947
948 %

```

The `pages` environment additionally sets the ‘column’ widths to the `\textwidth` (as known at the time the package is called). In this environment, there are two text in parallel on 2 pages.

```

949 \newenvironment{pages}{%
950 \l@dpairingtrue
951 \l@dpagingtrue
952 \initnumbering@quote
953 \save@familiarfootnote@number%
954 \if@ledgroup%
955 \get@familiarfootnote@number%
956 \fi%
957 \save@section@number%
958 \setlength{\Lcolwidth}{\textwidth}%
959 \setlength{\Rcolwidth}{\textwidth}%
960 }{%
961 \l@dpairingfalse
962 \l@dpagingfalse
963 \global\setlength{\Lcolwidth@pages}{\Lcolwidth}%
964 \global\setlength{\Rcolwidth@pages}{\Rcolwidth}%
965 }
966
967 %

```

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

```

968 \newif\ifinstanzaL
969 \newif\ifinstanzaR
970 %

```

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.

```

971 \newenvironment{Leftside}{%
972   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname\else%
973     \led@err@Leftside@PreviousNotPrinted%
974   \fi%
975   \ledRcolfalse
976   \setcounter{pstartL}{1}
977   \let\pstart\pstartL
978   \let\thepstart\thepstartL
979   \let\pend\pendL
980   \let\memorydump\memorydumpL
981   \Leftsidehook
982   \let\old@startstanza\@startstanza%
983   \def\@startstanza[##1][##2]{\global\instanzaLtrue\old@startstanza
[##1][##2]}%
984 }{
985   \expandafter\ifvoid\csname l@dLcolrawbox1\endcsname%
986     \led@error@missing@numbering{Leftside}%
987   \fi%
988   \Leftsidehookend}
989 %

```

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

```

\Leftsidehookend
\Rightsidehook
\Rightsidehookend
990 \newcommand*{\Leftsidehook}{}
991 \newcommand*{\Leftsidehookend}{}
992 \newcommand*{\Rightsidehook}{}
993 \newcommand*{\Rightsidehookend}{}
994
995 %

```

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.

```

996 \newenvironment{Rightside}{%
997   \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname\else%
998     \led@err@Rightside@PreviousNotPrinted%
999   \fi%
1000   \ledRcoltrue
1001   \let\beginnumbering\beginnumberingR
1002   \let\endnumbering\endnumberingR
1003   \let\pausenumbering\pausenumberingR
1004   \let\resumenumbering\resumenumberingR
1005   \let\memorydump\memorydumpR
1006   \let\thepstart\thepstartR
1007   \let\pstart\pstartR
1008   \let\pend\pendR
1009   \let\ledpb\ledpbR
1010   \let\lednopb\lednopbR

```

```

1011 \let\lineation\lineationR
1012 \Rightsidehook
1013 \let\old@startstanza\@startstanza%
1014 \def\@startstanza[##1][##2]{\global\instanzaRtrue\old@startstanza
1015 [##1][##2]}%
1016 \ledRcolfalse
1017 \expandafter\ifvoid\csname l@dRcolrawbox1\endcsname%
1018 \led@error@missing@numbering{Rightside}%
1019 \fi%
1020 \Rightsidehookend
1021 }
1022
1023 %

```

VIII Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than straight onto the vertical list, and when the paragraph ends we slice the paragraph into its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

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

\num@linesR Here are numbers and flags that are used internally in the course of the paragraph decomposition.

\one@lineR When we first form the paragraph, it goes into a box register, \l@dLcolrawbox or \l@dRcolrawbox for right text, instead of onto the current vertical list. The \ifnumberedpar@ flag will be true while a paragraph is being processed in that way. **\par@lineR** \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.

```

1024 \newcount\num@linesR
1025 \newbox\one@lineR
1026 \newcount\par@lineR
1027 %

```

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

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

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

```

1028 \newcounter{pstartL}
1029 \renewcommand{\thepstartL}{\bfseries\@arabic\c@pstartL}. }
1030 \newcounter{pstartR}
1031 \renewcommand{\thepstartR}{\bfseries\@arabic\c@pstartR}. }
1032
1033 \newcommand*{\pstartL}[2][1,2,usedefault]{%
1034   \if@nbreak%
1035     \let\@oldnbreak\@nbreaktrue%
1036   \else%
1037     \let\@oldnbreak\@nbreakfalse%
1038   \fi%
1039   \@nbreaktrue%
1040   \ifluatex%
1041     \xdef\l@luatextextdir@L{\the\textdir}%
1042     \xdef\l@luatexpardir@L{\the\pardir}%
1043     \xdef\l@luatexbodydir@L{\the\bodydir}%
1044   \fi%
1045   \ifnumbering \else%
1046     \led@err@PstartNotNumbered%
1047     \beginnumbering%
1048   \fi%
1049   \ifnumberedpar@%
1050     \led@err@PstartInPstart%
1051     \pend%
1052   \fi%
1053 }%
1054

```

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

```

1055 \ifpst@rtedL\else%
1056   \list@clear{\inserts@list}%
1057   \global\let\next@insert=\empty%
1058   \global\pst@rtedLtrue%
1059 \fi%
1060 \begingroup\everypar{}%
1061 %

```

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.

```

1062 \global\advance\l@dnumpstartsL \@one%
1063 \ifnum\l@dnumpstartsL>\l@dc@maxchunks%
1064   \led@err@TooManyPstarts%
1065   \global\l@dnumpstartsL=\l@dc@maxchunks%
1066 \fi%
1067 \global\setnamebox{\l@dLcolrawbox\the\l@dnumpstartsL}=\vbox\bgroup%
1068 %

```

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.

```

1069 \l@dzeropenalties%
1070 \ifautopar\else%
1071   \ifnumberpstart%
1072     \ifsidepstartnum%
1073       \else%
1074         \thepstartL%
1075       \fi%
1076     \fi%
1077   \fi%
1078 \@@start@every@pstart%
1079 \hsize=\Lcolwidth%
1080 \numberedpar@true%
1081 \iflabelpstart%
1082   \protected@edef\@currentlabel{\p@pstartL\thepstartL}%
1083 \fi%
1084 %

```

Dump the optional arguments

```

1085 \providetoggle{before@pstartL@the\l@dnumpstartsL @par}%
1086 \ifboolexpr{%
1087   test {\ifstrempy{#1}}%
1088   and test {\ifstrempy{#2}}%
1089 }%
1090 {%
1091   \ifat@every@pstart@star@%
1092   \global\togglefalse{before@pstartL@the\l@dnumpstartsL @par}%
1093   \else%
1094   \global\toggletrue{before@pstartL@the\l@dnumpstartsL @par}%
1095   \fi%
1096   \csgdef{before@pstartL@the\l@dnumpstartsL}{\at@every@pstart}%
1097 }%
1098 {%
1099   \ifstrempy{#1}{}%
1100   \global\toggletrue{before@pstartL@the\l@dnumpstartsL @par}%
1101   \csgdef{before@pstartL@the\l@dnumpstartsL}{\noindent#1}%
1102   }%
1103   \ifstrempy{#2}{}%
1104   {%
1105     \csgdef{before@pstartL@the\l@dnumpstartsL}{#2}%
1106     \global\togglefalse{before@pstartL@the\l@dnumpstartsL @par}%
1107   }%
1108 }%
1109 \at@every@pstart@call%
1110 %

```

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

```

1111 \ignorespaces%
1112 %

```

```

1113 }
1114 %

```

The same for right side.

```

1115 \newcommand*{\pstartR}[2][1,2,usedefault]{%
1116   \if@nbreak%
1117     \let\@oldnbreak\@nbreaktrue%
1118   \else%
1119     \let\@oldnbreak\@nbreakfalse%
1120   \fi%
1121   \@nbreaktrue%
1122   \ifluatex%
1123     \xdef\l@luatextextdir@R{\the\textdir}%
1124     \xdef\l@luatexpardir@R{\the\pardir}%
1125     \xdef\l@luatexbodydir@R{\the\bodydir}%
1126   \fi%
1127   \ifnumberingR \else%
1128     \led@err@PstartNotNumbered%
1129     \beginnumberingR%
1130   \fi%
1131   \ifnumberedpar@%
1132     \led@err@PstartInPstart%
1133     \pendR%
1134   \fi%
1135   \ifpst@rtedR\else%
1136     \list@clear{\inserts@listR}%
1137     \global\let\next@insertR=\empty%
1138     \global\pst@rtedRtrue%
1139   \fi%
1140   \begingroup\everypar{}%
1141   \global\advance\l@dnumstartsR \@ne%
1142   \ifnum\l@dnumstartsR>\l@dc@maxchunks%
1143     \led@err@TooManyPstarts%
1144     \global\l@dnumstartsR=\l@dc@maxchunks%
1145   \fi%
1146   \global\setnamebox{\l@dRcolrawbox\the\l@dnumstartsR}=\vbox\bgroup%
1147   \l@dzeropenalties%
1148   \ifautopar\else%
1149     \ifnumberpstart%
1150       \ifsidepstartnum\else%
1151         \thepstartR%
1152       \fi%
1153     \fi%
1154   \fi%
1155   \@at@start@every@pstart%
1156   \hsize=\Rcolwidth%
1157   \numberedpar@true%

```

```

1158 \iflabelpstart%
1159   \protected@edef\@currentlabel{\p@pstartR\thepstartR}%
1160 \fi%
1161 \providetoggle{before@pstartR@the\l@dnumpstartsR @par}%
1162 \ifboolexpr{%
1163   test {\ifstrempy{#1}}%
1164   and test {\ifstrempy{#2}}%
1165 }%
1166 {%
1167   \ifat@every@pstart@star@%
1168   \global\togglefalse{before@pstartR@the\l@dnumpstartsR @par}%
1169   \else%
1170   \global\toggletrue{before@pstartR@the\l@dnumpstartsR @par}%
1171   \fi%
1172   \csgdef{before@pstartR@the\l@dnumpstartsR}{\at@every@pstart}%
1173 }%
1174 {%
1175   \ifstrempy{#1}{}%
1176   \global\toggletrue{before@pstartR@the\l@dnumpstartsR @par}%
1177   \csgdef{before@pstartR@the\l@dnumpstartsR}{\noindent#1}%
1178   }%
1179   \ifstrempy{#2}{}%
1180   {%
1181     \csgdef{before@pstartR@the\l@dnumpstartsR}{#2}%
1182     \global\togglefalse{before@pstartR@the\l@dnumpstartsR @par}%
1183   }%
1184   }%
1185   \at@every@pstart@call%
1186   \ignorespaces%
1187 }
1188 %

```

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

```

1189 \newcommandx*\pendL}[2][1,2,usedefault]{%
1190   \ifnumbering \else%
1191     \led@err@PendNotNumbered%
1192   \fi%
1193   \ifnumberedpar@ \else%
1194     \led@err@PendNoPstart%
1195   \fi%
1196 %

```

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.

```

1197   \@at@end@every@pend%
1198   \endgraf\global\num@lines=\prevgraf\egroup%
1199   \global\par@line=0%
1200 %

```


End the group that was begun in the \pstart.

```

1201 \endgroup%
1202 \ignorespaces%
1203 \@oldnobreak%
1204 \dump@pstartL@pc%
1205 \ifnumberpstart%
1206   \addtocounter{pstartL}{1}%
1207   \ifcontinuousnumberingwithcolumns%
1208     \addtocounter{pstart}{1}%
1209   \fi%
1210 \fi
1211 \parledgroup@beforenotes@save{L}%
1212 %

```

Dump content of the optional argument.

```

1213 \providetoggle{after@pendL@the\l@dnumpstartsL @par}%
1214 \ifboolexpr{%
1215   test {\ifstrempy{#1}}%
1216   and test {\ifstrempy{#2}}%
1217 }%
1218 {%
1219   \ifat@every@pend@star%
1220   \global\togglefalse{after@pendL@the\l@dnumpstartsL @par}%
1221   \else%
1222     \global\toggletrue{after@pendL@the\l@dnumpstartsL @par}%
1223   \fi%
1224   \csgdef{after@pendL@the\l@dnumpstartsL}{\at@every@pend}%
1225 }%
1226 {%
1227   \ifstrempy{#1}{%
1228     \global\toggletrue{after@pendL@the\l@dnumpstartsL @par}%
1229     \csgdef{after@pendL@the\l@dnumpstartsL}{\noindent#1}%
1230   }%
1231   \ifstrempy{#2}{%
1232     {%
1233       \csgdef{after@pendL@the\l@dnumpstartsL}{#2}%
1234       \global\togglefalse{after@pendL@the\l@dnumpstartsL @par}%
1235     }%
1236   }%
1237 }%
1238 %

```

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

```

1239 \newcommandx*\pendR}[2][1,2,usedefault]{%
1240   \ifnumberingR \else%
1241     \led@err@PendNotNumbered%
1242   \fi%
1243   \ifnumberedpar@ \else%

```

```

1244 \led@err@PendNoPstart%
1245 \fi%
1246 \@at@end@every@pend%
1247 \endgraf\global\num@linesR=\prevgraf\egroup%
1248 \global\par@lineR=0%
1249 \endgroup%
1250 \ignorespaces%
1251 \@oldnobreak%
1252 \dump@pstartR@pc%
1253 \ifnumberpstart%
1254 \addtocounter{pstartR}{1}%
1255 \fi%
1256 \parledgroup@beforenotes@save{R}%
1257 \providetoggle{after@pendR@the\l@dnumpstartsR @par}%
1258 \ifboolexpr{%
1259   test {\ifstrempy{#1}}%
1260   and test {\ifstrempy{#2}}%
1261 }%
1262 {%
1263   \ifat@every@pend@star@%
1264   \global\togglefalse{after@pendR@the\l@dnumpstartsR @par}%
1265   \else%
1266   \global\toggletrue{after@pendR@the\l@dnumpstartsR @par}%
1267   \fi%
1268   \csgdef{after@pendR@the\l@dnumpstartsR}{\at@every@pend}}%
1269 {%
1270   \ifstrempy{#1}{}%
1271   \csgdef{after@pendR@the\l@dnumpstartsR}{\noindent#1}%
1272   \global\toggletrue{after@pendR@the\l@dnumpstartsR @par}%
1273   }%
1274   \ifstrempy{#2}{}%
1275   \csgdef{after@pendR@the\l@dnumpstartsR}{#2}%
1276   \global\togglefalse{after@pendR@the\l@dnumpstartsR @par}%
1277   }%
1278 }%
1279 }
1280
1281 %

```

\AtEveryPstartCall The `\AtEveryPstartCall` argument is called when the `\pstartL` or `\pstartR` is called. That is different of `\AtEveryPstart` the argument of which is called when the `\pstarts` are printed.

```

1282 \newcommand{\AtEveryPstartCall}[1]{\gdef\at@every@pstart@call{#1}}%
1283 \gdef\at@every@pstart@call{}%
1284 %

```

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

```

1285 \newif\ifprint@last@after@pendL%

```

```

1286 \newif\ifprint@last@after@pendR%
1287 %

```

VIII.2 Processing one line

For parallel texts we have to be able to process left and right lines independently. For sequential text we happily use the original `\do@line`. Otherwise ...

`\l@dleftbox` A line of left text will be put in the box `\l@dleftbox`, and analogously for a line of right text.

`\l@drightbox`

```

1288 \newbox\l@dleftbox
1289 \newbox\l@drightbox
1290
1291 %

```

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

`\countRline`

```

1292 \newcount\countLline
1293 \countLline \z@
1294 \newcount\countRline
1295 \countRline \z@
1296
1297 %

```

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

`\@donetotallinesL`

`\@donereallinesR`

`\@donetotallinesR`

```

1298 \newcount\@donereallinesL
1299 \newcount\@donetotallinesL
1300 \newcount\@donereallinesR
1301 \newcount\@donetotallinesR
1302
1303 %

```

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

```

1304 \newcommand*{\do@lineL}{%
1305 \letcs{\ifnumberpstart}{numberpstart@L\the\l@dpscl}%
1306 \advance\countLline \@ne%
1307 \ifvbox\namebox{1@dLcolrawbox\the\l@dpscl}%
1308 {\vbadness=10000%
1309 \splittopskip=\z@%
1310 \do@lineLhook%
1311 \l@demptyd@ta%
1312 \global\setbox\one@line=\vsplit\namebox{1@dLcolrawbox\the\l@dpscl}%
1313 to\baselineskip}%

```

```

1314 \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startL}{}%
1315 \unvbox\one@line \global\setbox\one@line=\lastbox%
1316 \@writepageofparL%
1317 \getline@numL%
1318 \ifnum\@lock>\@ne%
1319 \inserthangingsymboltrue%
1320 \else%
1321 \inserthangingsymbolfalse%
1322 \fi%
1323 \setbox\l@dleftbox%
1324 \hb@xt@ \Lcolwidth{%
1325 \ifl@dhiddenumber%
1326 \global\l@dhiddenumberfalse%
1327 \f@x@l@cks%
1328 \else%
1329 \affixline@num%
1330 \fi%
1331 \xifinlist{\the\l@dpscL}{\eled@sections@@}%
1332 {\unless\ifshiftedpstarts%
1333 \add@apparatusL%
1334 \fi%
1335 }%
1336 {\print@lineL}%
1337 }%
1338 \add@penaltiesL%
1339 \global\advance\@donereallinesL\@ne%
1340 \global\advance\@donetotallinesL\@ne%
1341 \else%
1342 \iflinenumberLevenifblank
1343 \new@lineL%
1344 \l@emptyd@ta%
1345 \getline@numL%
1346 \affixline@num%
1347 \setbox\l@dleftbox \hb@xt@ \Lcolwidth{%
1348 \l@dld@ta%
1349 \hspace*{\Lcolwidth}%
1350 \ledrlfill\l@dldr@ta%
1351 }%
1352 \else%
1353 \setbox\l@dleftbox \hb@xt@ \Lcolwidth{\hspace*{\Lcolwidth}}%
1354 \fi%
1355 \global\advance\@donetotallinesL\@ne%
1356 \fi%
1357 }%
1358
1359
1360 %

```

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

`\print@line` for handbook.

```

1361 \def\print@lineL{%
1362   \affixpstart@numL%
1363   \l@dld@ta%
1364   \add@apparatusL%
1365   \l@dlsn@te%
1366   \do@insidelineLhook%
1367   \hb@xt@ \Lcolwidth{\ledllfill\hb@xt@ \wd\one@line{%
1368     \ifluatex%
1369       \texdir\l@luatextextdir@L%
1370     \fi%
1371     \new@lineL%
1372     \inserthangingsymbolL%
1373     \l@dunhbox@line{\one@line}}\ledrlfill\l@drd@ta%
1374     \l@drsn@te}}%
1375
1376 %

```

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

```

1377 \def\print@eledsectionL{%%
1378   \addtocounter{pstartL}{-1}%
1379   \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{%
1380   \ifdefstring{\@eledsectmark}{L}{\ledsectnomark}%
1381   \numdef{\temp@}{\l@dpscL-1}%
1382   \xifinlist{\temp@}{\eled@sections@}{\@nobraektrue}{\@nobraekfalse}%
1383   \@eled@sectioningtrue%
1384   \bgroup%
1385     \ifluatex%
1386       \texdir\l@luatextextdir@L%
1387       \pardir\l@luatexpardir@L%
1388       \bodydir\l@luatexbodydir@L%
1389       \ifdefstring{\l@luatextextdir@L}{TRT}{\@RTLtrue}{%
1390       \fi%
1391       \csuse{eled@sectioning@the\l@dpscL}%
1392     \egroup%
1393     \@eled@sectioningfalse%
1394     \global\csundef{eled@sectioning@the\l@dpscL}%
1395     \if@RTL%
1396       \hspace{-3\paperwidth}%
1397     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1398     \else%
1399       \hspace{3\paperwidth}%
1400     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1401     \fi%
1402     \vskip\eledsection@correcting@skip%
1403   }
1404
1405 %

```

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

```

1406 \newcommand{\add@apparatusL}{%
1407   \if@firstlineofpage%
1408     \set@Xtxtbeforenotes%
1409     \set@txtbeforenotesX%
1410     \global\@firstlineofpagefalse%
1411   \fi%
1412   \ifdefstring{\ms@data@position}{msdata-regular}{%
1413     \insert@msdata%
1414     \add@inserts%
1415     \add@Xgroupbyline%
1416   }{%
1417     \add@inserts%
1418     \add@Xgroupbyline%
1419     \insert@msdata%
1420   }%
1421   \affixside@note%
1422 }%
1423 %

```

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

```

\dolineRhook
\doinlinedlineLhook
\doinlinedlineRhook
1424 \newcommand*\dolineLhook[1]{\gdef\dolineLhook{#1}}%
1425 \newcommand*\dolineRhook[1]{\gdef\dolineRhook{#1}}%
1426 \newcommand*\doinlinedlineLhook[1]{\gdef\doinlinedlineLhook{#1}}%
1427 \newcommand*\doinlinedlineRhook[1]{\gdef\doinlinedlineRhook{#1}}%
1428
1429 %

```

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

```

\dolineRhook
\do@insidelineLhook
\do@insidelineRhook
1430 \newcommand*\do@lineLhook{}
1431 \newcommand*\do@lineRhook{}
1432 \newcommand*\do@insidelineLhook{}
1433 \newcommand*\do@insidelineRhook{}
1434
1435 %

```

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

```

1436 \newcommand*\do@lineR{%
1437   \let\linenumrepL\linenumrep%
1438   \let\sublinenumrepL\sublinenumrep%
1439   \let\linenumrepR\linenumrepR%
1440   \let\sublinenumrepR\sublinenumrepR%
1441   \letcs{\ifnumberpstart}{numberpstart@R\the\l@dpscr}%

```

```

1442 \ledRcol@true%
1443 \advance\countRline \@ne%
1444 \ifvbox\namebox{l@dRcolrawbox\the\l@dpscR}%
1445 {\vbadness=10000%
1446 \splittopskip=\z@%
1447 \do@lineRhook%
1448 \l@emptyd@ta%
1449 \global\setbox\one@lineR=\vsplit\namebox{l@dRcolrawbox\the\l@dpscR}%
1450 to\baselineskip}%
1451 \IfStrEq{\splitfirstmarks\parledgroup@}{begin}{\
parledgroup@notes@startR}{}%
1452 \unvbox\one@lineR \global\setbox\one@lineR=\lastbox%
1453 \@writepageofparR%
1454 \getline@numR%
1455 \ifnum\@lockR>\@ne%
1456 \inserthangingsymbolRtrue%
1457 \else%
1458 \inserthangingsymbolRfalse%
1459 \fi%
1460 \setbox\l@drightbox%
1461 \hb@xt@ \Rcolwidth{%
1462 \ifl@dhiddenumber%
1463 \global\l@dhiddenumberfalse%
1464 \f@x@l@cksR%
1465 \else%
1466 \affixline@numR%
1467 \fi%
1468 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
1469 {\unless\ifshiftedpstarts%
1470 \add@apparatusR%
1471 \fi%
1472 }%
1473 {\print@lineR}%
1474 }%
1475 \add@penaltiesR%
1476 \global\advance\@donereallinesR\@ne%
1477 \global\advance\@donetotallinesR\@ne%
1478 \else%
1479 \iflinenumberRevenifblank%
1480 \new@lineR
1481 \l@emptyd@ta%
1482 \getline@numR%
1483 \setbox\l@drightbox \hb@xt@ \Rcolwidth{%
1484 \affixline@numR%
1485 \l@dld@ta%
1486 \hspace*{\Rcolwidth}%
1487 \ledrlfill\l@drd@ta%
1488 }%
1489 \else%
1490 \setbox\l@drightbox \hb@xt@ \Rcolwidth{\hspace*{\Rcolwidth}}%

```

```

1491 \fi%
1492 \global\advance\@donetotallinesR\@ne%
1493 \fi%
1494 \ledRcol@false%
1495 \let\linenumrep\linenumrepL%
1496 \let\sublinenumrep\sublinenumrepL%
1497 }
1498
1499
1500 %

```

`\print@lineR`
`\print@eledsectionR`
`\add@apparatusR`

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

```

1501 \newcommand{\add@apparatusR}{%
1502 \if@firstlineofpageR%
1503 \set@Xtxtbeforenotes%
1504 \set@txtbeforenotesX%
1505 \global\@firstlineofpageRfalse%
1506 \fi%
1507 \ifdefstring{\ms@data@position}{msdata-regular}{%
1508 \insert@msdata%
1509 \add@insertsR%
1510 \add@Xgroupbyline%
1511 }{%
1512 \add@insertsR%
1513 \add@Xgroupbyline%
1514 \insert@msdata%
1515 }%
1516 \affixside@noteR%
1517 }%
1518 %

```

VIII.3 Line and page number computation

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

```

1519 \newcommand*{\getline@numR}{%
1520 \global\advance\absline@numR \@ne
1521 \do@actionsR
1522 \do@ballastR
1523 \ifledgroupnotesR\else
1524 \ifnumberline
1525 \ifsublines@R
1526 \ifnum\sub@lockR<\tw@
1527 \global\advance\subline@numR \@ne
1528 \fi

```



```

1529         \else
1530             \ifnum\@lockR<\tw@
1531                 \global\advance\line@numR \@ne
1532                 \global\subline@numR \z@
1533             \fi
1534         \fi
1535     \fi
1536 \fi
1537 }
1538 \newcommand*\getline@numL}{%
1539     \global\advance\absline@num \@ne
1540     \do@actions
1541     \do@ballast
1542         \ifledgroupnotesL\@else
1543             \ifnumberline
1544                 \ifsublines@
1545                     \ifnum\sub@lock<\tw@
1546                         \global\advance\subline@num \@ne
1547                     \fi
1548                 \else
1549                     \ifnum\@lock<\tw@
1550                         \global\advance\line@num \@ne
1551                         \global\subline@num \z@
1552                     \fi
1553                 \fi
1554             \fi
1555         \fi
1556 }
1557
1558 %
1559 %

```

`\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.

```

1560 \newcommand*\do@ballastR}{\global\ballast@count=\z@
1561     \begingroup
1562         \advance\absline@numR \@ne
1563         \ifnum\next@actionlineR=\absline@numR
1564             \ifnum\next@actionR>-1001
1565                 \global\advance\ballast@count by -\c@ballast
1566             \fi
1567         \fi
1568     \endgroup}
1569 %

```

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

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

```

1570
1571 \newif\ifl@dskipversenumberR
1572 \newcommand*{\do@actions@fixedcodeR}{%
1573   \ifcase\@l@dttempcnta%
1574   \or% % 1001 = starting sublineation
1575     \global\sublines@Rtrue
1576   \or% % 1002 = ending sublineation
1577     \global\sublines@Rfalse
1578   \or% % 1003 = starting locking number
1579     \global\@lockR=\@one
1580   \or% % 1004 = ending locking number
1581     \ifnum\@lockR=\tw@
1582       \global\@lockR=\thr@@
1583     \else
1584       \global\@lockR=\z@
1585     \fi
1586   \or% % 1005 = starting locking subnumber
1587     \global\sub@lockR=\@one
1588   \or% % 1006 = ending locking subnumber
1589     \ifnum\sub@lockR=\tw@
1590       \global\sub@lockR=\thr@@
1591     \else
1592       \global\sub@lockR=\z@
1593     \fi
1594   \or% % 1007 = skipping numbering
1595     \l@dskipnumbertrue
1596   \or% % 1008 = skipping numbering in stanza
1597     \l@dskipversenumberRtrue%
1598   \or% % 1009 = hiding number
1599     \l@dhiddenumbertrue%
1600   \or% % 1010 = inserting msdata
1601     \add@msdata%
1602   \else%
1603     \led@warn@BadAction
1604   \fi%
1605 }
1606
1607
1608 \newcommand*{\do@actionsR}{%
1609   \global\let\do@actions@nextR=\relax
1610   \@l@dttempcntb=\absline@numR
1611   \ifnum\@l@dttempcntb<\next@actionlineR\else
1612     \ifnum\next@actionR>-1001\relax
1613     \unless\ifresumenumberingR@start%
1614       \@firstlineofpageRtrue%
1615     \fi%
1616     \global\page@numR=\next@actionR
1617     \ifbypage@R
1618       \ifcsdef{l@dchset@num@R\the\absline@numR}%
1619         {%

```

```

1620     \global\csundef{l@dchset@numR\the\absline@numR}%
1621   }%
1622   {%
1623     \unless\ifresumenumberingR@start%
1624       \global\line@numR \z@%
1625       \global\subline@numR \z@%
1626     \fi%
1627   }%
1628   \fi
1629   \global\resumenumberingR@startfalse%
1630   \add@msdata@firstlineofpage%
1631 \else
1632   \ifnum\next@actionR<-4999\relax % 9/05 added relax here
1633     \l@dtempcnta=-\next@actionR
1634     \advance\l@dtempcnta by -5001\relax
1635     \ifsublines@R
1636       \global\subline@numR=\l@dtempcnta
1637     \else
1638       \global\line@numR=\l@dtempcnta
1639     \fi
1640   \else
1641     \l@dtempcnta=-\next@actionR
1642     \advance\l@dtempcnta by -1000\relax
1643     \do@actions@fixedcodeR
1644   \fi
1645 \fi
1646 \ifx\actionlines@listR\empty
1647   \gdef\next@actionlineR{1000000}%
1648 \else
1649   \gl@p\actionlines@listR\to\next@actionlineR
1650   \gl@p\actions@listR\to\next@actionR
1651   \global\let\do@actions@nextR=\do@actionsR
1652 \fi
1653 \fi
1654 \do@actions@nextR}
1655
1656 %

```

VIII.4 Line number printing

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

```

\ch@cksub@l@ckR
\ch@ck@l@ckR
\fx@l@cksR
\affixline@numR
1657
1658 \newcommand*{\l@dcalcnun}[3]{%
1659   \ifnum #1 > #2\relax
1660     \l@dtempcnta = #1\relax
1661     \advance\l@dtempcnta by -#2\relax
1662     \divide\l@dtempcnta by #3\relax
1663     \multiply\l@dtempcnta by #3\relax

```

```

1664 \advance\@l@tempcnta by #2\relax
1665 \else
1666 \@l@tempcnta=#2\relax
1667 \fi}
1668
1669 \newcommand*\ch@cksub@l@ckR}{%
1670 \ifcase\sub@lockR
1671 \or
1672 \ifnum\sublock@disp=\@ne
1673 \@l@tempcntb \z@ \@l@tempcnta \@ne
1674 \fi
1675 \or
1676 \ifnum\sublock@disp=\tw@
1677 \else
1678 \@l@tempcntb \z@ \@l@tempcnta \@ne
1679 \fi
1680 \or
1681 \ifnum\sublock@disp=\z@
1682 \@l@tempcntb \z@ \@l@tempcnta \@ne
1683 \fi
1684 \fi}
1685
1686 \newcommand*\ch@ck@l@ckR}{%
1687 \ifcase\@lockR
1688 \or
1689 \ifnum\lock@disp=\@ne
1690 \@l@tempcntb \z@ \@l@tempcnta \@ne
1691 \fi
1692 \or
1693 \ifnum\lock@disp=\tw@
1694 \else
1695 \@l@tempcntb \z@ \@l@tempcnta \@ne
1696 \fi
1697 \or
1698 \ifnum\lock@disp=\z@
1699 \@l@tempcntb \z@ \@l@tempcnta \@ne
1700 \fi
1701 \fi}
1702
1703 \newcommand*\f@x@l@ckR}{%
1704 \ifcase\@lockR
1705 \or
1706 \global\@lockR \tw@
1707 \or \or
1708 \global\@lockR \z@
1709 \fi
1710 \ifcase\sub@lockR
1711 \or
1712 \global\sub@lockR \tw@
1713 \or \or

```

```

1714 \global\sub@lockR \z@
1715 \fi}
1716
1717
1718 \let\linenumberlistR\empty%
1719 \newcommand*{\affixline@numR}{%
1720 \ifledgroupnotesR@ \else \ifnumberline
1721 \ifl@dskipnumber
1722 \global\l@dskipnumberfalse
1723 \else
1724 \ifsublines@R
1725 \@l@dttempcntb=\subline@numR
1726 \l@dcalcnum{\subline@numR}{\c@firstsublinenumR}{\c@sublinenumincrementR
}%
1727 \ch@cksub@l@ckR
1728 \else
1729 \@l@dttempcntb=\line@numR
1730 \ifx\linenumberlistR\empty%
1731 \l@dcalcnum{\line@numR}{\c@firstlinenumR}{\c@linenumincrementR}%
1732 \else
1733 \@l@dttempcnta=\line@numR
1734 \edef\rem@inderR{\linenumberlistR,\number\line@numR,%
1735 \edef\sc@n@list{\def\noexpand\sc@n@list
1736 ###1,\number\@l@dttempcnta,###2|\def\noexpand\rem@inderR{###2}}}%
%
1737 \sc@n@list\expandafter\sc@n@list\rem@inderR|
1738 \ifx\rem@inderR\empty\advance\@l@dttempcnta\@ne\fi%
1739 \fi
1740 \ch@ck@l@ckR
1741 \fi
1742 \ifnum\@l@dttempcnta=\@l@dttempcntb
1743 \ifl@dskipversenumberR\else
1744 \if@twocolumn
1745 \if@firstcolumn
1746 \gdef\l@dld@ta{\llap{\leftlinenumR}}}%
1747 \else
1748 \gdef\l@drd@ta{\rlap{\rightlinenumR}}}%
1749 \fi
1750 \else
1751 \ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columnsR}{\m@ne}}}%
1752 {\@l@dttempcntb=\line@margin@columnsR}%
1753 {\@l@dttempcntb=\line@marginR}%
1754 \ifnum\@l@dttempcntb>\@ne
1755 \advance\@l@dttempcntb by\page@numR
1756 \fi
1757 \ifboolexpr{%
1758 bool {l@dprintingcolumns}%
1759 and (%
1760 (test {\ifdefstring{\linenum@OnlyPages@ForColumnsR}{left}})%

```

```

1761         and test {\ifnumodd{\page@numR}}%
1762         )%
1763         or%
1764         (test {\ifdefstring{\linenum@OnlyPages@ForColumnsR}{right}}%
1765         and not test {\ifnumodd{\page@numR}}%
1766         )%
1767     )%
1768 }%
1769 {}%
1770 {%
1771     \ifodd\@l@tempcntb%
1772     \gdef\l@drd@ta{\rlap{\{\rightlinenumR\}}}%
1773     \else%
1774     \gdef\l@dld@ta{\llap{\{\leftlinenumR\}}}%
1775     \fi%
1776 }%
1777 \fi
1778 \fi
1779 \fi
1780 \f@x@l@cksR
1781 \fi
1782 \fi
1783 \fi}
1784 %

```

VIII.5 Pstart number printing in side

The printing of the pstart number is like in `reledmac`, with two differences :

- Some commands have versions suffixed by R or L.
- The `\affixpstart@num` and `\affixpstart@numR` commands are called in the `\Pages` command. Consequently, the `pstartL` and `pstartR` counters must be reset at the beginning of this command.

```

\affixpstart@numL85
\affixpstart@numR86 \newcommand*{\affixpstart@numL}{%
\leftpstartnum87 \ifsidepstartnum
\rightpstartnumR88 \if@twocolumn
\leftpstartnumL89 \if@firstcolumn
\rightpstartnumL90 \gdef\l@dld@ta{\llap{\{\leftpstartnumL\}}}%
\ifpstartnumR91 \else
\gdef\l@drd@ta{\rlap{\{\rightpstartnumL\}}}%
\fi
\else
\ifboolexpr{bool {l@dprintingcolumns} and test {\ifnumgreater{\
line@margin@columns}{\m@ne}}}%
\@l@tempcntb=\line@margin@columns}%
\@l@tempcntb=\line@margin}%

```

```

1798 \ifnum\@l@tempcntb>\@ne
1799 \advance\@l@tempcntb \page@num
1800 \fi
1801 \ifodd\@l@tempcntb
1802 \gdef\l@drd@ta{\rlap{{\rightpstartnumL}}}%
1803 \else
1804 \gdef\l@dld@ta{\llap{{\leftpstartnumL}}}%
1805 \fi
1806 \fi
1807 \fi
1808 }
1809 \newcommand*{\affixpstart@numR}{%
1810 \ifsidepstartnum
1811 \if@twocolumn
1812 \if@firstcolumn
1813 \gdef\l@dld@ta{\llap{{\leftpstartnumR}}}%
1814 \else
1815 \gdef\l@drd@ta{\rlap{{\rightpstartnumR}}}%
1816 \fi
1817 \else
1818 \ifbool{expr{bool {l@dprintingcolumns} and test {\ifnumgreater{\line@margin@columnsR}{\m@ne}}}%
1819 {\@l@tempcntb=\line@margin@columnsR}%
1820 {\@l@tempcntb=\line@marginR}%
1821 \ifnum\@l@tempcntb>\@ne
1822 \advance\@l@tempcntb \page@numR
1823 \fi
1824 \ifodd\@l@tempcntb
1825 \gdef\l@drd@ta{\rlap{{\rightpstartnumR}}}%
1826 \else
1827 \gdef\l@dld@ta{\llap{{\leftpstartnumR}}}%
1828 \fi
1829 \fi
1830 \fi
1831 }
1832
1833 \newcommand*{\leftpstartnumL}{%
1834 \ifpstartnum
1835 \thepstartL
1836 \kern\linenumsep\global\pstartnumfalse\fi
1837 }
1838 \newcommand*{\rightpstartnumL}{%
1839 \ifpstartnum\kern\linenumsep
1840 \thepstartL
1841 \global\pstartnumfalse\fi
1842 }
1843 \newif\ifpstartnumR
1844 \pstartnumRtrue
1845 \newcommand*{\leftpstartnumR}{%
1846 \ifpstartnumR

```

```

1847 \thepstartR
1848 \kern\linenumsep\global\pstartnumRfalse\fi
1849 }
1850 \newcommand*{\rightpstartnumR}{
1851 \ifpstartnumR\kern\linenumsep
1852 \thepstartR
1853 \global\pstartnumRfalse\fi
1854 }
1855 %

```

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.

```

1856 \list@create{\inserts@listR}
1857 %

```

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

```

1858 \newcommand*{\add@insertsR}{%
1859 \global\let\add@inserts@nextR=\relax
1860 \ifx\inserts@listR\empty \else
1861 \ifx\next@insertR\empty
1862 \ifx\insertlines@listR\empty
1863 \global\noteschanged@true
1864 \gdef\next@insertR{100000}%
1865 \else
1866 \gl@p\insertlines@listR\to\next@insertR
1867 \fi
1868 \fi
1869 \ifnum\next@insertR=\absline@numR
1870 \gl@p\inserts@listR\to\@insertR
1871 \@insertR
1872 \global\let\@insertR=\undefined
1873 \global\let\next@insertR=\empty
1874 \global\let\add@inserts@nextR=\add@insertsR
1875 \fi
1876 \fi
1877 \add@inserts@nextR}
1878
1879 %

```

VIII.7 Penalties

`\add@penaltiesL` `\add@penaltiesL` is the last macro used by `\do@lineL`. It adds up the club, widow, and interline penalties, and puts a single penalty of the appropriate size back into the paragraph; these penalties get removed by the `\vsplit` operation. `\displaywidowpenalty`

and `\brokenpenalty` are not restored, since we have no easy way to find out where we should insert them.

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

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

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

```
1880 \newcommand*{\add@penaltiesL}{\}
1881 \newcommand*{\add@penaltiesR}{\}
1882
1883 %
```

VIII.8 Printing leftover notes

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

```

1884 \newcommand*{\flush@notesR}{%
1885   \iftoggle{notfirstrun@jobname.\extensionchars\the\section@numR R}{%
1886     \xloop%
1887     \ifx\inserts@listR\empty \else%
1888       \gl@p\inserts@listR\to\@insertR%
1889       \@insertR%
1890       \global\let\@insertR=\undefined%
1891     \repeat%
1892   }{}%
1893 }%
1894
1895
1896 %

```

IX Footnotes

IX.1 Footnotes output specific to \Pages

```

\print@Xnotes@forpages
\correct@Xfootins@box
\print@notesX@forpages
\correct@footinsX@box

```

The `\Xonlyside` and `\onlysideX` hooks for `\Pages` allow notes to be printed either in left or right pages only. The implementation of such features is delegated to `\print@Xnotes@forpages`, which replaces `\print@Xnotes` inside `\Pages`. Here is how we proceed⁶:

- If notes are to be printed in both sides, we just proceed the usual way: print the foot starts for the series, then the foot group.
- If notes are to be printed in the left side, we do these prints only for even pages ; if notes are to be printed in the right side, we do these prints only for odd pages.
- However, that is not enough. Because the problem does not only consists in printing notes in any particular page. It is also not to put aside room for notes in the pages where we do not want to print them. To take an example: if some note in the left side is too long by 160pt to be printed in full in the left page, we do not want to put aside 160pt a space for it in the following right page.
- To solve this problem, we change the magnification factor associated with notes before going to the next page. If we start a page where no notes are supposed to be printed, the magnification counter is set to 0. 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.

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

- 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.

```
1897 \newcommand\print@Xnotes@forpages[1]{%
1898 %
```

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

```
1899 \ifcseempty{Xonlyside@#1}{%
1900 \csuse{#1footstart}{#1}%
1901 \csuse{#1footgroup}{#1}%
1902 }%
1903 %
```

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

```
1904 {%
1905 \ifboolexpr{%
1906 ((test {\ifcsstring{Xonlyside@#1}{L}} and not test{\ifnumodd{\c@page
1907 }})%
1908 or%
1909 (test {\ifcsstring{Xonlyside@#1}{R}} and test{\ifnumodd{\c@page}}))%
1910 }%
1911 %
```

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

```
1911 {%
1912 \csuse{#1footstart}{#1}%
1913 \csuse{#1footgroup}{#1}%
1914 %
```

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.

```
1915 \global\count\csuse{#1footins}=\z@%
1916 \global\skip\csuse{#1footins}=\z@%
1917 \global\dimen\csuse{#1footins}=\maxdimen%
1918 }%
1919 %
```

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

```
1920 {%
1921 \global\count\csuse{#1footins}=\csuse{default@#1footins}%
1922 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
1923 \bgroup%
1924 \csuse{Xnotefontsize@#1}%
1925 \global\dimen\csuse{#1footins}=\csuse{Xmaxhnotes@#1}%
1926 \egroup%
1927 %
```

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

```
1928 \global\setnamebox{#1footins@kept}=\box\namebox{#1footins}%
1929 }%
1930 %
```

End of \print@Xnotes@forpages.

```
1931 }%
1932 }%
1933 %
```

And now, the same for familiar footnotes.

```
1934 \newcommand\print@notesX@forpages[1]{%
1935 \ifcempty{onlysideX@#1}{%
1936 \csuse{footstart#1}{#1}%
1937 \csuse{footgroup#1}{#1}%
1938 }%
1939 {%
1940 \ifboolexpr{%
1941 ((test {\ifcsstring{onlysideX@#1}{L}} and not test{\ifnumodd{\c@page
1942 }})%
1943 or%
1944 (test {\ifcsstring{onlysideX@#1}{R}} and test{\ifnumodd{\c@page}}))%
1945 }%
1946 \csuse{footstart#1}{#1}%
1947 \csuse{footgroup#1}{#1}%
1948 \global\count\csuse{footins#1}=\z@%
1949 \global\skip\csuse{footins#1}=\z@%
1950 \global\dimen\csuse{footins#1}=\maxdimen%
1951 }%
1952 {%
1953 \global\count\csuse{footins#1}=\csuse{default@footins#1}%
1954 \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
1955 \bgroup%
1956 \csuse{Xnotefontsize@#1}%
1957 \global\dimen\csuse{footins#1}=\csuse{maxhnotesX@#1}%
1958 \egroup%
1959 \global\setnamebox{footins#1@kept}=\box\namebox{footins#1}%
1960 }%
1961 }%
1962 }%
1963 }%
1964 %
```

`\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.

```
1965 \newcommand{\insert@notes@for@onlyside}{%
```

```

1966 \def\do##1{%
1967 \unless\ifnocritical@%
1968 \ifvoid\csuse{##1footins@kept}\else%
1969 \expandafter\insert\csname ##1footins\endcsname%
1970 \bgroup%
1971 \unvnamebox{##1footins@kept}%
1972 \egroup%
1973 \fi%
1974 \fi%
1975 \unless\ifnofamiliar@%
1976 \ifvoid\csuse{footins##1@kept}\else%
1977 \expandafter\insert\csname footins##1\endcsname%
1978 \bgroup%
1979 \unvnamebox{footins##1@kept}%
1980 \egroup%
1981 \fi%
1982 \fi%
1983 }%
1984 \dolistloop{\@series}%
1985 }%
1986 %

```

IX.2 Critical footnote printed in right side

`\edtext@later` and `\edtextnow` are used to print critical footnotes on the right side, while referring to the text on the left side. First, we create two counters, one for `\edtext@later` and the other for `\edtextnow`.

```

1987 \newcount\edtext@now%
1988 \newcount\edtext@later%
1989 %
1990 \newcommand{\edtextlater}[2]{%#1 lemma, #2 critical notes
1991 %

```

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

```

1992 \global\advance\edtext@later by \@ne%
1993 %
1994 % As we are in a pseudo-\cs{edtext}, we now need to get the \cs{sameword}
1995 data stored in the auxiliary file for this \cs{edtextlater}.
1996 \advance\@edtext@level by \@ne%
1997 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1998 {\global\let\sw@inthisedtext\empty}%
1999 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\endcsname\to\sw@inthisedtext}%
2000 %

```

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

```

2000 %
2001 \csxdef{edtext@later@\the\edtext@later}{%
2002 %

```

\edtextnow will insert a empty \edtext.

```

2003 \noexpand\edtext{%
2004 }%
2005 %

```

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

```

2006 {%
2007 \unexpanded{%
2008 \lemma{\no@expands #1}%
2009 }%
2010 \noexpand\xxref{start:edtext:later:\the\edtext@later}{end:edtext:
later:\the\edtext@later}%
2011 \noexpand\linenum{||||\edfont@info}%
2012 \unexpanded{\def\sw@inthisedtext}{\expandonce{\sw@inthisedtext}}%
2013 %

```

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.

```

2014 \ifledRcol%
2015 \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@true}}%
2016 \else%
2017 \unexpanded{\appto\@beforeinsertofthisedtext{\ledRcol@false}}%
2018 \fi%
2019 %

```

And the footnote command of this \edtextlater.

```

2020 \unexpanded{#2}%
2021 }%
2022 }%
2023 %

```

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

```

2024 \edlabel{start:edtext:later:\the\edtext@later}%
2025 \flag@start@later%
2026 \bgroup%
2027 \def\lemma##1{%
2028 \ifledRcol%
2029 \write\linenum@outR{\string\@lemma}%
2030 \else%
2031 \write\linenum@out{\string\@lemma}%
2032 \fi%
2033 }%

```

```

2034 \renewcommand{\do}[1]{\expandafter\renewcommandx\csname ##1footnote\
endcsname[2][1,usedefault]{}\unskip}%\unskip because of a spurious space in
\newcommandx
2035 \dolistloop{\@series}%
2036 #2%
2037 \egroup%
2038 \showlemma{#1}%
2039 \edlabel{end:edtext:later:\the\edtext@later}%
2040 \flag@end@later%
2041 %

```

We decrease the counter increased at the beginning.

```

2042 \advance\@edtext@level by -\@ne%
2043 }%
2044 %

```

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

```

2045 \newcommand{\edtextnow}[0]{%
2046 \global\advance\edtext@now by \@ne
2047 \csuse{edtext@later@\the\edtext@now}%
2048 }%
2049 %

```

X Cross referencing

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

```

2050 \list@create{\labelref@listR}
2051
2052 %

```

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

\l@dmake@labelsR This is the right text version of \l@dmake@labels, taking account of \@Rlineflag.

```

2053 \def\l@dmake@labelsR#1|#2|#3|#4|#5|#6{%
2054 \expandafter\ifx\csname the@label\csuse{XR@prefix}#6\endcsname%
2055 \relax%
2056 \else%
2057 \led@warn@DuplicateLabel{\csuse{XR@prefix}#6}%
2058 \fi%
2059 \expandafter\gdef\csname the@label\csuse{XR@prefix}#6\endcsname
{#1|#2|#3|#4|#5|\@Rlineflag}%
2060 \global\providetoggle{label@#6@ledRcol}%False is the default value of
this toggle, which tell us if a label is linked to a right or a left side
2061 \global\toggletrue{label@#6@ledRcol}%

```

```

2062 \ignorespaces}
2063 \AtBeginDocument{%
2064   \def\l@dmake@labelsR#1|#2|#3|#4|#5|#6{%
2065 }
2066
2067 %

```

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

It is defined on `reledmac`.

XI Side notes

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

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

`\sidenotemargin*`

`\sidenotemarginR`

```

2068 \WithSuffix\newcommand\sidenotemargin*[1]{%
2069   \l@dgetsidenote@margin{#1}
2070   \global\sidenote@marginR=\@l@tempcntb
2071   \global\sidenote@margin=\@l@tempcntb
2072 }
2073 \newcommand{\sidenotemarginR}[1]{%
2074   \l@dgetsidenote@margin{#1}%
2075   \global\sidenote@marginR=\@l@tempcntb%
2076 }%
2077 \newcount\sidenote@marginR
2078 \global\sidenote@marginR=\@ne%
2079
2080 \ifmovecolumnspostiononrightpage%
2081   \sidenotemargin{inner}%
2082   \sidenotemarginR{outer}%
2083 \fi%
2084
2085 %

```

`\@morespace@rightnote@leftcolumn`

`\@morespace@leftnote@rightcolumn`

`\get@sidenote@morespace@columns`

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.


```

2086 \newdimen\@morespace@leftnote@rightcolumn%
2087 \newdimen\@morespace@rightnote@leftcolumn%
2088
2089
2090
2091 \newcommand{\get@sidenote@morespace@columns}{%
2092 %

```

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

```

2093 \if@sidenotesmarginpage%
2094 %

```

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

```

2095 \global\@morespace@leftnote@rightcolumn=\dimexpr\intercolumns@width + \
Lcolwidth\relax%
2096 \global\@morespace@rightnote@leftcolumn=\dimexpr\intercolumns@width + \
Rcolwidth\relax%
2097 %

```

If \if@sidenotesmarginpage is False

```

2098 \else%
2099 \global\@morespace@leftnote@rightcolumn=\z@%
2100 \global\@morespace@rightnote@leftcolumn=\z@%
2101 %

```

End of \get@sidenote@morespace@columns.

```

2102 \fi%
2103 }%
2104 %

```

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

```

2105 \newcommand*{\affixside@noteR}{%
2106 \prepare@edindex@for@note{\the\page@numR|\the\line@numR|\the\
subline@numR|\the\page@numR|\the\line@numR|\the\subline@numR|}%
2107 \def\sidenotecontent@{}%
2108 \numgdef\itemcount@{0}%
2109 \def\do##1{%
2110 \ifnumequal{\itemcount@}{0}%
2111 {%
2112 \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
2113 {\appto\sidenotecontent@{\@sidenotessep ##1}%
2114 }%
2115 \numgdef\itemcount@{\itemcount@+1}%
2116 }%
2117 \dolistloop{\l@dcsnotetext}%
2118 \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%

```

```

2119 \gdef\@templ@d{%
2120 \gdef\@templ@n{\l@dcstetext\l@dcstetext@l\l@dcstetext@r}%
2121 \ifx\@templ@d\@templ@n \else%
2122 \if@twocolumn%
2123 \if@firstcolumn%
2124 \setl@dlp@rbox{##1}{\sidenotecontent@}%
2125 \else%
2126 \setl@drp@rbox{\sidenotecontent@}%
2127 \fi%
2128 \else%
2129 \@l@tempcntb=\sidenote@marginR%
2130 \ifnum\@l@tempcntb>\@ne%
2131 \advance\@l@tempcntb by\page@numR%
2132 \fi%
2133 \ifodd\@l@tempcntb%
2134 \setl@drp@rbox{\sidenotecontent@}%
2135 \gdef\sidenotecontent@{%
2136 \numdef{\itemcount@}{0}%
2137 \dolistloop{\l@dcstetext@l}%
2138 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
2139 \setl@dlp@rbox{\sidenotecontent@}%
2140 \else%
2141 \setl@dlp@rbox{\sidenotecontent@}%
2142 \gdef\sidenotecontent@{%
2143 \numdef{\itemcount@}{0}%
2144 \dolistloop{\l@dcstetext@r}%
2145 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
2146 \setl@drp@rbox{\sidenotecontent@}%
2147 \fi%
2148 \fi%
2149 \fi%
2150 \advance\@edindex@fornote@\m@ne%
2151 }
2152
2153 %

```

XII Verse

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

```

\inserthangingsymbolL54 \newif\ifinserthangingsymbolR
\inserthangingsymbolR55 \newcommand{\inserthangingsymbolL}{%
2156 \ifinserthangingsymbol%
2157 \ifinstanzaL%
2158 \hskip \@@undefined{sza@00}{0}{\expandafter%

```

```

2159 \noexpand\csname sza@0@\endcsname\stanzaindentbase%
2160 \@hangingsymbol%
2161 \fi%
2162 \fi%
2163 }%
2164 \newcommand{\inserthangingsymbolR}{%
2165 \ifinserthangingsymbolR%
2166 \ifinstanzaR%
2167 \hskip \ifundefined{sza@0@}{0}{\expandafter%
2168 \noexpand\csname sza@0@\endcsname\stanzaindentbase%
2169 \@hangingsymbol%
2170 \fi%
2171 \fi%
2172 }%
2173 %

```

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.

```

2174 \chardef\next=\catcode`&
2175 \catcode`&=\active
2176
2177 %

```

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

```

2178 \newenvironmentx{astanza}[2][1,2,usedefault]{%
2179 \catcode`\&=\active
2180 \global\stanza@count\@ne\stanza@modulo\@ne
2181 \ifnum\usernamecount{sza@0@}=\z@
2182 \let\stanza@hang\relax
2183 \let\endlock\relax
2184 \else
2185 \rightskip\z@ plus 1fil\relax
2186 \fi
2187 \ifnum\usernamecount{szp@0@}=\z@
2188 \let\sza@penalty\relax
2189 \fi
2190 \def&{%
2191 \endlock\mbox{}%
2192 \sza@penalty
2193 \global\advance\stanza@count\@ne
2194 \@astanza@line}%
2195 \def\&{\@stopastanza}%
2196 \ifboolexpr{%
2197 not test{\ifdefined{\at@every@stanza}}%
2198 and test{\ifstrempy{#1}}%
2199 and test{\ifstrempy{#2}}}%
2200 {\pstart[][\at@every@stanza]\at@start@every@stanza}%
2201 {\pstart[#1][#2]\at@start@every@stanza}%

```

```

2202 \@astanza@line
2203 \let\par\relax\ignorespaces%No paragraph in verses
2204 }{}
2205
2206 %

```

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

```

2207 \newcommandx{\@stopastanza}[2][1,2,usedefault]{%
2208 \endlock\mbox{}%
2209 \ifboolexpr{%
2210 not test{\ifdefvoid{\at@every@stop@stanza}}%
2211 and test{\ifstrempy{#1}}%
2212 and test{\ifstrempy{#2}}%
2213 }%
2214 {\before@every@stop@stanza\pend[][\at@every@stop@stanza]}%
2215 {\before@every@stop@stanza\pend[#1][#2]}%
2216 }%
2217 %

```

\@astanza@line This gets put at the start of each line in the environment. It sets up the paragraph style — each line is treated as a paragraph.

```

2218 \newcommand*{\@astanza@line}{%
2219 \ifnum\value{stanzaindentrepetition}=0
2220 \ifcsdef{sza@number\stanza@count @}{%
2221 {%
2222 \parindent=\csname sza@number\stanza@count @\endcsname\
2223 stanzaindentbase%
2224 }{%
2225 \led@err@StanzaIndentNotDefined%
2226 }%
2227 \else
2228 \ifcsdef{sza@number\stanza@modulo @}{%
2229 \parindent=\csname sza@number\stanza@modulo @\endcsname\
2230 stanzaindentbase%
2231 \managestanza@modulo%
2232 }{%
2233 \led@err@StanzaIndentNotDefined%
2234 }%
2235 \fi
2236 \endgraf
2237 \stanza@hang%
2238 \ignorespaces}
2239 %

```

Lastly reset the modified category codes.

```

2240 \catcode`\&=\next
2241
2242 %

```

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

```

\thestanzaR
2243 \newcounter{stanzaL}
2244 \newcounter{stanzaR}
2245 \renewcommand{\thestanzaL}{%
2246   \textbf{\arabic{stanzaL}}%
2247 }
2248 \renewcommand{\thestanzaR}{%
2249   \textbf{\arabic{stanzaR}}%
2250 }
2251 %
2252 %

```

XIII Fixing babel and polyglossia

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

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

```

\ifl@dusedbabel A flag for checking if babel has been used as a package.
\l@dusedbabelfalse
2253 \newif\ifl@dusedbabel
\l@dusedbabeltrue
2254 %

```

`\l@dchecklang`

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

```

2255 \patchcmd{\bbl@set@language}{%
2256   {\select@language{\language}}}%
2257   {\edef\language{#1}\select@language{\language}}}%
2258   {}%
2259   {}%
2260
2261 %

```

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

```
\selectlanguage \selectlanguage is a babel command. \theledlanguageL and \theledlanguageR
\l@duselanguage are the names of the languages of the left and right texts. \l@duselanguage is similar
\theledlanguageL to \selectlanguage.
\theledlanguageR \newcommand*{\l@duselanguage}[1]{%
2262 \gdef\theledlanguageL{%
2263 \gdef\theledlanguageR{%
2264
2265
2266 %
```

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

```
2267 \AtBeginDocument{%
2268 \ifundefined{xpg@main@language}{%
2269 \ifundefined{bbl@main@language}{%
2270 %
```

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

```
2271 \l@dusedbabelfalse
2272 }{%
2273 %
```

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.

```
2274 \l@dusedbabeltrue
2275 \let\l@doldselectlanguage\selectlanguage
2276 \let\l@doldbbl@set@language\bbl@set@language
2277 \renewcommand{\selectlanguage}[1]{%
2278 \l@doldselectlanguage{#1}%
2279 \ifledRcol \gdef\theledlanguageR{#1}%
2280 \else \gdef\theledlanguageL{#1}%
2281 \fi}
2282 %
```

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

```
2283 \renewcommand*{\l@duselanguage}[1]{%
2284 \expandafter\l@doldselectlanguage\expandafter{#1}}
2285 %
```

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

```
2286 \gdef\theledlanguageL{\bbl@main@language}%
2287 \gdef\theledlanguageR{\bbl@main@language}%
2288 }%
2289 }
2290 %
```

If use polyglossia

```

2291 { \let\old@otherlanguage\otherlanguage%
2292 \renewcommand{\otherlanguage}[2][]{%
2293 \selectlanguage[#1]{#2}%
2294 \ifledRcol \gdef\theledlanguageR{#2}%
2295 \else \gdef\theledlanguageL{#2}%
2296 \fi}%
2297 \renewcommand{\l@duselanguage}[1]{%
2298 \csuse{no\language@ @numbers}\select@language{#1}%
2299 }%
2300 \gdef\theledlanguageL{\xpg@main@language}%
2301 \gdef\theledlanguageR{\xpg@main@language}%
2302 %

```

That is it.

```

2303 }}
2304 %

```

XIV Counts and boxes for parallel texts

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

`\maxchunks` The maximum number of chunk pairs before printing has to be called for. The default is
`\l@dc@maxchunks` 5120 chunk pairs.

```

2305 \newcount\l@dc@maxchunks
2306 \newcommand{\maxchunks}[1]{\l@dc@maxchunks=#1}
2307 \maxchunks{5120}
2308
2309 %

```

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

`\l@dnumpstartsR`

```

2310 \newcount\l@dnumpstartsR
2311
2312 %

```

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

`\l@pscR`

```

2313 \newcount\l@dp scl
2314 \newcount\l@dp scR
2315
2316 %

```

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

```

2317 \newcommand*\l@dsetupprawboxes}{%
2318   \@l@tempcntb=\l@dc@maxchunks
2319   \loop\ifnum\@l@tempcntb>\z@
2320     \newnamebox{\l@dLcolrawbox\the\@l@tempcntb}
2321     \newnamebox{\l@dRcolrawbox\the\@l@tempcntb}
2322     \advance\@l@tempcntb \m@ne
2323   \repeat}
2324
2325 %

```

`\l@dsetupmaxlinecounts` To be able to synchronise left and right texts we need to know the maximum number of text lines there are in each pair of chunks. `\l@dsetupmaxlinecounts` creates `\maxchunks` new counts called `\l@dmaxlinesinpar1`, etc., and `\l@dzeromaxlinecounts` zeroes all of them.

```

2326 \newcommand*\l@dsetupmaxlinecounts}{%
2327   \@l@tempcntb=\l@dc@maxchunks
2328   \loop\ifnum\@l@tempcntb>\z@
2329     \newnamecount{\l@dmaxlinesinpar\the\@l@tempcntb}
2330     \advance\@l@tempcntb \m@ne
2331   \repeat}
2332 \newcommand*\l@dzeromaxlinecounts}{%
2333   \begingroup
2334   \@l@tempcntb=\l@dc@maxchunks
2335   \loop\ifnum\@l@tempcntb>\z@
2336     \global\usenamecount{\l@dmaxlinesinpar\the\@l@tempcntb}=\z@
2337     \advance\@l@tempcntb \m@ne
2338   \repeat
2339   \endgroup}
2340
2341 %

```

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

```

2342 \AtBeginDocument{%
2343   \l@dsetupprawboxes
2344   \l@dsetupmaxlinecounts
2345   \l@dzeromaxlinecounts
2346   \l@dnumpstartsL=\z@
2347   \l@dnumpstartsR=\z@
2348   \l@dpscL=\z@
2349   \l@dpscR=\z@}
2350
2351 %

```


XV Checking text to be processed

```

\if@pstarts \check@pstarts returns \@pstartstrue if there are any unprocessed chunks.
\@pstartstrue
\@pstartsfalse
\check@pstarts
2352 \newif\if@pstarts
2353 \newcommand*{\check@pstarts}{\%
2354   \@pstartsfalse
2355   \ifnum\l@dnumpstartsL>\l@dpscL
2356     \@pstartstrue
2357   \else
2358     \ifnum\l@dnumpstartsR>\l@dpscR
2359       \@pstartstrue
2360     \fi
2361   \fi
2362 }
2363
2364 %

```

```

\ifaraw@text \checkraw@text checks whether the current Left or Right box is void or not. If
\araw@texttrue one or other is not void it sets \araw@texttrue, otherwise both are void and it sets
\araw@textfalse \araw@textfalse.
\checkraw@text
2365 \newif\ifaraw@text
2366 \newcommand*{\checkraw@text}{\%
2367   \araw@textfalse
2368   \ifvbox\namebox{1@dLcolrawbox\the\l@dpscL}
2369     \araw@texttrue
2370   \else
2371     \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}
2372       \araw@texttrue
2373     \fi
2374   \fi
2375 }
2376
2377 %

```

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

```

2378 \newcommand*{\@writelinesinparL}{\%
2379   \edef\next{\%
2380     \write\linenum@out{\string\@pend[\the\@donereallinesL]}}\%
2381   \next
2382   \global\@donereallinesL \z@}
2383 \newcommand*{\@writelinesinparR}{\%
2384   \edef\next{\%
2385     \write\linenum@outR{\string\@pendR[\the\@donereallinesR]}}\%
2386   \next
2387   \global\@donereallinesR \z@}
2388
2389 %

```

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

`\@writepageofparR`

```

2390 \newcommand*{\@writepageofparL}[0]{%
2391   \ifnum\@donereallinesL=\z@%
2392     \edef\next{%
2393       \write\linenum@out{\string\@pstart{\the\l@dpscl}{\the\c@page}{\the\
numpagelinesL}}%
2394     }%
2395     \next%
2396   \fi%
2397 }%
2398 \newcommand*{\@writepageofparR}[0]{%
2399   \ifnum\@donereallinesR=\z@%
2400     \edef\next{%
2401       \write\linenum@outR{\string\@pstartR{\the\l@dpscR}{\the\c@page}{\the\
numpagelinesR}}%
2402     }%
2403     \next%
2404   \fi%
2405 }%
2406 %

```

XVI Parallel columns

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

`\@eledsectionR`

```

2407 \newsavebox{\@eledsectionL}%
2408 \newsavebox{\@eledsectionR}%
2409 %

```

`\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.

```

2410 \newcommand*{\Columns}{%
2411   \ifl@dpairing%
2412     \led@err@Columns@InsideEnv%
2413   \fi%
2414   \ifboolexpr{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{
l@dLcolrawbox1}}}{%
2415     \ifcsboxvoid{l@dRcolrawbox1}{%
2416       \ifcsboxvoid{l@dLcolrawbox1}%
2417       {\led@err@Columns@WithoutEnv}%
2418       {\led@err@Columns@WithoutRightside}%
2419     }%
2420     {\led@err@Columns@WithoutLeftside}%
2421   }{%
2422     \global\l@dprintingcolumnstrue%
2423     \eledsection@correcting@skip=-\baselineskip% Correction for sections'
titles

```

```

2424 \ifnum\l@dnumstartsL=\l@dnumstartsR\else
2425 \led@err@BadLeftRightPstarts{\the\l@dnumstartsL}{\the\l@dnumstartsR}%
2426 \fi
2427 %

```

Start a group and zero counters, etc.

```

2428 \begingroup
2429 \l@dzeropenalties
2430 \endgraf\global\num@lines=\prevgraf
2431 \global\num@linesR=\prevgraf
2432 \global\par@line=\z@
2433 \global\par@lineR=\z@
2434 \global\l@dpscL=\z@
2435 \global\l@dpscR=\z@
2436 \get@familiarfootnote@number%
2437 \get@intercolumns@width%
2438 \get@sidenote@morespace@columns%
2439 %

```

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

```

2440 \check@pstarts
2441 \loop\if@pstarts
2442 \global\pstartnumtrue
2443 \global\pstartnumRtrue
2444 %

```

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

```

2445 \global\advance\l@dpscL \@ne
2446 \global\advance\l@dpscR \@ne
2447 \restore@pstartL@pc%
2448 \restore@pstartR@pc%
2449 %

```

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

```

2450 \Columns@print@before@pstart%
2451 %

```

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

```

2452 \checkraw@text
2453 { \loop\ifaraw@text
2454 %

```

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

```

2455 \l@duselanguage{\theledlanguageL}%
2456 \do@lineL

```

```

2457 \xifinlist{\the\l@dpscl}{\eled@sections@@}
2458 {%
2459 \ifdefstring{\@eledsectmark}{L}%
2460 {\csuse{eled@sectmark@the\l@dpscl}%
2461 }}%
2462 \global\csundef{eled@sectmark@the\l@dpscl}%
2463 \savebox{\@eledsectionL}{\parbox[t]{}[t]{\Lcolwidth}{\vbox
{} \print@eledsectionL}}% \vbox{}-> prevent alignment troubles with RTL
language
2464 }%
2465 {}%
2466 \l@duselanguage{\theledlanguageR}%
2467 \do@lineR
2468 \xifinlist{\the\l@dpscr}{\eled@sectionsR@@}
2469 {%
2470 \ifdefstring{\@eledsectmark}{R}%
2471 {\csuse{eled@sectmark@the\l@dpscr R}%
2472 }}%
2473 \global\csundef{eled@sectmark@the\l@dpscr R}%
2474 \savebox{\@eledsectionR}{\parbox[t]{}[t]{\Rcolwidth}{\vbox
{} \print@eledsectionR}}% \vbox{}-> prevent alignment troubles with RTL
language
2475 {}%
2476 \hb@xt@ \hsize{%
2477 \ifdefstring{\columns@position}{L}{\hfill }%
2478 \print@leftcolumn%
2479 \print@columnseparator%
2480 \print@rightcolumn%
2481 \ifdefstring{\columns@position}{R}{\hfill}%
2482 }%
2483 \checkraw@text
2484 \checkverseL
2485 \checkverseR
2486 \checkpb@columns
2487 \repeat}
2488 %

```

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.

```

2489 \@writelinesinparL
2490 \@writelinesinparR
2491 \check@pstarts
2492 \ifbypstart@%
2493 \write\linenum@out{\string\@set[1]}
2494 \resetprevline@
2495 \fi
2496 \ifbypstart@R
2497 \write\linenum@outR{\string\@set[1]}
2498 \resetprevline@

```

```

2499     \fi
2500     \Columns@print@after@pend%
2501   \repeat
2502 %

```

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.

```

2503   \flush@notes
2504   \flush@notesR
2505 \endgroup
2506 %

```

```

2507 \global\l@dpscl=\z@
2508 \global\l@dpscR=\z@
2509 \global\l@dnumpsL=\z@
2510 \global\l@dnumpsR=\z@
2511 \global\l@dprintingcolumnsfalse%
2512 \ignorespaces
2513   \global\instanzaLfalse%
2514   \global\instanzaRfalse%
2515 }%
2516 }%
2517
2518 %

```

\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).

```

2519 \def\print@columnseparator{%
2520   \ifdim\beforecolumnseparator<0pt%
2521     \hfill%
2522   \else%
2523     \hspace{\beforecolumnseparator}%
2524   \fi%
2525   \columnseparator%
2526   \ifdim\aftercolumnseparator<0pt%
2527     \hfill%
2528   \else%
2529     \hspace{\aftercolumnseparator}%
2530   \fi%
2531 }%
2532 %

```

\get@intercolumns@width The `\intercolumns@width` is calculated by `\get@intercolumns@width`. This length depends of many parameters:

- Columns width;
- columns position;

- columns separator;
- space between columns and columns separator, which can be fixed by user or automatically calculated by `reledpar`.

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

```

2533 \newdimen\intercolumns@width%
2534 \newcommand{\get@intercolumns@width}{%
2535 %   \begin{macrocode}
2536   \global\intercolumns@width=\z@%
2537   \ifdefstring{\columns@position}{C}%
2538 %
```

First case, the columns are centered.

```

2539   {%
2540 %
```

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

```

2541   \ifboolexpr{%
2542     test {\ifdimless{\beforecolumnseparator}{\z@}}%
2543     and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2544   }%
2545   {%
2546     \global\intercolumns@width=\dimexpr%
2547       \columnrulewidth +%
2548       (\textwidth - \lcolwidth - \rcolwidth - \columnrulewidth)*1/2%
2549     \relax%
2550   }%
2551   {%
2552 %
```

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

```

2553   \ifboolexpr{%
2554     test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2555     and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2556   }%
2557   {%
2558     \global\intercolumns@width=\dimexpr%
2559       \columnrulewidth + \beforecolumnseparator + \aftercolumnseparator
2560 %
2561     \relax%
2562   }%
2563   {%
2564 %
```

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

```

2564 \ifboolexpr{%
2565     test {\ifdimgreater{\beforecolumnseparator}{\z@}}}%
2566     and test {\ifdimless{\aftercolumnseparator}{\z@}}}%
2567 }%
2568 {%
2569     \global\intercolumns@width=\dimexpr%
2570         \columnrulewidth + \beforecolumnseparator +%
2571         (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
beforecolumnseparator) * 1/3%
2572     \relax%
2573 }%
2574 {}%
2575 %

```

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

```

2576 \ifboolexpr{%
2577     test {\ifdimless{\beforecolumnseparator}{\z@}}}%
2578     and test {\ifdimgreater{\aftercolumnseparator}{\z@}}}%
2579 }%
2580 {%
2581     \global\intercolumns@width=\dimexpr%
2582         \columnrulewidth + \aftercolumnseparator +%
2583         (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
aftercolumnseparator) * 1/3%
2584     \relax%
2585 }%
2586 {}%
2587 %

```

Now, we have finished with the case the columns are centered aligned.

```

2588 }%
2589 %

```

Other case, the columns are left or right aligned.

```

2590 {%
2591 %

```

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

```

2592 \ifboolexpr{%
2593     test {\ifdimless{\beforecolumnseparator}{\z@}}}%
2594     and test {\ifdimless{\aftercolumnseparator}{\z@}}}%
2595 }%
2596 {%
2597     \global\intercolumns@width=\dimexpr%
2598         \columnrulewidth +%
2599         (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth)*2/3%
The total of the width before and after column separator
2600     \relax%

```

```

2601 }%
2602 {}%
2603 %

```

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

```

2604 \ifboolexpr{%
2605   test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2606   and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2607 }%
2608 {%
2609   \global\intercolumns@width=\dimexpr%
2610   \columnrulewidth + \beforecolumnseparator + \aftercolumnseparator
2611 %
2612   \relax%
2613 }%
2614 {}%
2615 %

```

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

```

2615 \ifboolexpr{%
2616   test {\ifdimgreater{\beforecolumnseparator}{\z@}}%
2617   and test {\ifdimless{\aftercolumnseparator}{\z@}}%
2618 }%
2619 {%
2620   \global\intercolumns@width=\dimexpr%
2621   \columnrulewidth + \beforecolumnseparator +%
2622   (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
beforecolumnseparator) * 1/2%
2623   \relax%
2624 }%
2625 {}%
2626 %

```

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

```

2627 \ifboolexpr{%
2628   test {\ifdimless{\beforecolumnseparator}{\z@}}%
2629   and test {\ifdimgreater{\aftercolumnseparator}{\z@}}%
2630 }%
2631 {%
2632   \global\intercolumns@width=\dimexpr%
2633   \columnrulewidth + \aftercolumnseparator +%
2634   (\textwidth - \Lcolwidth - \Rcolwidth - \columnrulewidth - \
aftercolumnseparator) * 1/2%
2635   \relax%
2636 }%
2637 {}%
2638 %

```


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

```
2639 }%
2640 %
```

End of \get@intercolumns@width.

```
2641 }%
2642 %
```

`\print@leftcolumn` and `\print@rightcolumn` print the line number of the left or right column respectively. Normally, it is defined by the content of the `Leftside` or `Rightside` environments. But if `\movecolumnsonrightpage` is set to `TRUE`, it also depends on the page number.

```
2643 \newcommand{\print@leftcolumn}{%
2644   \if\page@num>\page@numR%
2645     \l@tempcntb=\page@num%
2646   \else%
2647     \l@tempcntb=\page@numR%
2648   \fi%
2649   \ifboolexpr{%
2650     not bool {movecolumnsonrightpage}%
2651     or test {\ifnumodd{\l@tempcntb}}%
2652   }%
2653   {%
2654     \unhbox\l@leftbox%
2655     \ifhbox\@eledsectionL%
2656       \usebox{\@eledsectionL}%
2657     \fi%
2658   }%
2659   {%
2660     \unhbox\l@rightbox%
2661     \ifhbox\@eledsectionR%
2662       \usebox{\@eledsectionR}%
2663     \fi%
2664   }%
2665 }%
2666
2667 \newcommand{\print@rightcolumn}{%
2668   \if\page@num>\page@numR%
2669     \l@tempcntb=\page@num%
2670   \else%
2671     \l@tempcntb=\page@numR%
2672   \fi%
2673   \ifboolexpr{%
2674     not bool {movecolumnsonrightpage}%
2675     or test {\ifnumodd{\l@tempcntb}}%
2676   }%
2677   {%
2678
```

```

2679 \unhbox\l@drightrightbox%
2680 \ifhbox\@eledsectionR%
2681 \usebox{\@eledsectionR}%
2682 \fi%
2683 }%
2684 {%
2685 \unhbox\l@dleftleftbox%
2686 \ifhbox\@eledsectionL%
2687 \usebox{\@eledsectionL}%
2688 \fi%
2689 }%
2690 }%
2691 %

```

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

```

2692
2693 \newcommand{\checkpb@columns}{%
2694 \newif\if@pb
2695 \newif\if@nopb
2696 \IfStrEq{\led@pb@setting}{before}{
2697 \numdef{\next@absline}{\the\absline@num+1}%
2698 \numdef{\next@abslineR}{\the\absline@numR+1}%
2699 \xifinlist{\next@absline}{\l@prev@pb}{\@pbtrue}{}%
2700 \xifinlist{\next@abslineR}{\l@prev@pbR}{\@pbtrue}{}%
2701 \xifinlist{\next@absline}{\l@prev@nopb}{\@nopbtrue}{}%
2702 \xifinlist{\next@abslineR}{\l@prev@nopbR}{\@nopbtrue}{}%
2703 }{}
2704 \IfStrEq{\led@pb@setting}{after}{
2705 \xifinlist{\the\absline@num}{\l@prev@pb}{\@pbtrue}{}%
2706 \xifinlist{\the\absline@numR}{\l@prev@pbR}{\@pbtrue}{}%
2707 \xifinlist{\the\absline@num}{\l@prev@nopb}{\@nopbtrue}{}%
2708 \xifinlist{\the\absline@numR}{\l@prev@nopbR}{\@nopbtrue}{}%
2709 }{}
2710 \if@nopb\nopagebreak[4]\enlargethispage{\baselineskip}\fi
2711 \if@pb\pagebreak[4]\fi
2712 }
2713 %

```

\columnseparator The separator between line pairs in parallel columns is in the form of a vertical rule extending a little below the baseline and with a height slightly greater than the \baselineskip. The width of the rule is \columnrulewidth (initially 0pt so the rule is invisible).

```

2714 \newcommand*\columnseparator{%
2715 \smash{\rule[-0.2\baselineskip]{\columnrulewidth}{1.05\baselineskip}}}
2716 \newdimen\columnrulewidth
2717 \columnrulewidth=\z@
2718

```

2719 %

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

```
2720 \newcommand*{\columnspan}[1]{%
2721   \xdef\columns@position{#1}%
2722 }%
2723 \xdef\columns@position{R}%
2724 %
```

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

```
2725 \newlength{\beforecolumnseparator}%
2726 \setlength{\beforecolumnseparator}{-2pt}%
2727
2728 \newlength{\aftercolumnseparator}%
2729 \setlength{\aftercolumnseparator}{-2pt}%
2730
2731 %
```

`setwidthliketwocolumns` 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 `\setnoteposition...` macros are called in `\xxxfootstart` in a **non-parallel** typesetting context to fix the position of notes block.

```
2732 \newcommand{\setwidthliketwocolumns}{%
2733   \get@intercolumns@width%As \columns is not necessary called before \
   setwidthliketwocolumns
2734   \hsize=\dimexpr \Lcolwidth + \intercolumns@width + \Rcolwidth \relax%
2735 }%
2736
2737 \newcommand{\setpositionliketwocolumns@L}{%
2738   \renewcommand{\ledrlfill}{\hfill}%
2739 }%
2740
2741 \newcommand{\setnotespositionliketwocolumns@L}{%
2742 }%
2743
2744 \newcommand{\setpositionliketwocolumns@C}{%
2745   \doinsidelinehook{\hfill}%
2746   \renewcommand{\ledrlfill}{\hfill}%
2747 }%
2748
2749 \newcommand{\setnotespositionliketwocolumns@C}{%

```

```

2750 \newdimen\temp%
2751 \newdimen\tempa%
2752 \temp=\hsize%
2753 \tempa=\Lcolwidth%
2754 \advance\tempa\Rcolwidth%
2755 \advance\temp-\tempa%
2756 \divide\temp by 2%
2757 \leftskip=\temp%
2758 \rightskip=-\temp%
2759 }%
2760
2761 \newcommand{\setpositionliketwocolumns@R}{%
2762 \doinsidelinehook{\hfill}%
2763 }%
2764 %

```

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

```

2765 \newcommand{\Columns@print@before@pstart}{%
2766 \ifboolexpr{%
2767   test{\ifcsstring{before@pstartL@the\l@dpscL}{\at@every@pstart}}%
2768   and test {\ifcsstring{before@pstartR@the\l@dpscR}{\at@every@pstart}}%
2769   and test {\ifdefempty{\at@every@pstart}}}%
2770   {%
2771   \ifboolexpr{not togl{before@pstartR@the\l@dpscR @par} and not togl{
2772 before@pstartL@the\l@dpscL @par}}{%
2773     \csuse{before@pstartL@the\l@dpscL}%
2774     \csuse{before@pstartR@the\l@dpscR}%
2775   }{%
2776     \hb@xt@ \hsize{%
2777       \ifdefstring{\columns@position}{L}{\hfill }%
2778       \par\parbox[t] [] [t]{\Lcolwidth}{%
2779         \csuse{before@pstartL@the\l@dpscL}%
2780       }%
2781       \print@columnseparator%
2782       \parbox[t] [] [t]{\Rcolwidth}{%
2783         \set@sectcountR%
2784         \csuse{before@pstartR@the\l@dpscR}%
2785       }%
2786       \ifdefstring{\columns@position}{R}{\hfill}%
2787     }%
2788   }%
2789 }%
2790 \global\csundef{before@pstartL@the\l@dpscL}%
2791 \global\csundef{before@pstartR@the\l@dpscR}%
2792 }%
2793 \newcommand{\Columns@print@after@pend}{%

```

```

2794 \ifboolexpr{%
2795   test{\ifcsstring{after@pendL@the\l@dpscl}{\at@every@pend}}}%
2796   and test {\ifcsstring{after@pendR@the\l@dpscr}{\at@every@pend}}}%
2797   and test {\ifdefempty{\at@every@pend}}}%
2798   {%
2799   {%
2800     \ifboolexpr{not togl{after@pendR@the\l@dpscr @par} and not togl{
after@pendL@the\l@dpscl @par}}{%
2801       \csuse{after@pendL@the\l@dpscl}%
2802       \csuse{after@pendR@the\l@dpscr}%
2803     }{%
2804       \hb@xt@ \hsize{%
2805         \ifdefstring{\columns@position}{L}{\hfill }%
2806         \parbox[t] [] [t]{\Lcolwidth}{%
2807           \csuse{after@pendL@the\l@dpscl}%
2808         }%
2809         \print@columnseparator%
2810         \parbox[t] [] [t]{\Rcolwidth}{%
2811           \set@sectcountR%
2812           \csuse{after@pendR@the\l@dpscr}%
2813         }%
2814         \ifdefstring{\columns@position}{R}{\hfill}%
2815       }%
2816     }%
2817   }%
2818   \global\csundef{after@pendL@the\l@dpscl}%
2819   \global\csundef{after@pendR@the\l@dpscr}%
2820 }%
2821 %

```

XVII Parallel pages

This is considerably more complicated than parallel columns.

XVII.1 Specific counters

`\numpagelinesL` Counts for the number of lines on a left or right page, and the smaller of the number of
`\numpagelinesR` lines on a pair of facing pages.
`\l@dminpagelines`

```

2822 \newcount\numpagelinesL
2823 \newcount\numpagelinesR
2824 \newcount\l@dminpagelines
2825
2826 %

```

XVII.2 Main macro

`\Pages` The `\Pages` command results in the previous Left and Right texts being typeset on matching facing pages. There should be equal numbers of chunks in the left and right texts.

```

2827 \newcommand*{\Pages}[1][1,usedefault]{%
2828   \ifl@dpairing%
2829     \led@err@Pages@InsideEnv%
2830   \fi%
2831   \ifbool{test{\ifcsboxvoid{l@dRcolrawbox1}} or test{\ifcsboxvoid{
l@dLcolrawbox1}}}{%
2832     \ifcsboxvoid{l@dRcolrawbox1}{%
2833       \ifcsboxvoid{l@dLcolrawbox1}%
2834       {\led@err@Pages@WithoutEnv}%
2835       {\led@err@Pages@WithoutRightside}%
2836     }%
2837     {\led@err@Pages@WithoutLeftside}%
2838   }{%
2839     \ifstrequal{#1}{mainmatter}{\Pages@mainmattertrue}{\Pages@mainmatterfalse
}%
2840     \eledsection@correcting@skip=-2\baselineskip% line correcting for section
titles.
2841     \parledgroup@notespacing@set@correction%
2842     \typeout{}%
2843     \typeout{***** PAGES *****}%
2844     \ifnum\l@dnumpstartsL=\l@dnumpstartsR\else%
2845       \led@err@BadLeftRightPstarts{\the\l@dnumpstartsL}{\the\l@dnumpstartsR}%
2846     \fi%
2847   }%

```

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

```

2848   \cleartol@devenpage%
2849   \global\l@dprintingpagetrue%
2850   \begingroup%
2851   %

```

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

```

2852   \setlength{\Lcolwidth}{\Lcolwidth@pages}%
2853   \setlength{\Rcolwidth}{\Rcolwidth@pages}%
2854   %

```

```

2855   \l@dzeropenalties%
2856   \endgraf\global\num@lines=\prevgraf%
2857   \global\num@linesR=\prevgraf%
2858   \global\par@line=\z@%
2859   \global\par@lineR=\z@%
2860   \global\l@dpscL=\z@%
2861   \global\l@dpscR=\z@%

```

```

2862 \writtenlinesLfalse%
2863 \writtenlinesRfalse%
2864 \get@familiarfootnote@number%
2865 %

```

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

```

2866 \let\print@Xnotes\print@Xnotes@forpages%
2867 \let\print@notesX\print@notesX@forpages%
2868 %

```

Check if there are chunks to be processed.

```

2869 \check@pstarts%
2870 \loop\if@pstarts%
2871 %

```

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

```

2872 \global\advance\l@dpscL \@ne%
2873 \global\advance\l@dpscR \@ne%
2874 %

```

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

```

2875 \getlinesfromparlistL%
2876 \getlinesfromparlistR%
2877 \l@dcalc@maxoftwo{\@cs@linesinparL}{\@cs@linesinparR}%
2878 {\usernamecount{l@dmaxlinesinpar\the\l@dpscL}}%
2879 \check@pstarts%
2880 \repeat%
2881 %

```

Zero the counts again, ready for the next bit.

```

2882 \global\l@dpscL=\z@%
2883 \global\l@dpscR=\z@%
2884 %

```

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

```

2885 \getlinesfrompagelistL%
2886 \getlinesfrompagelistR%
2887 \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
2888 {\l@dminpagelines}%
2889 %

```

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.

```

2890 \check@pstarts%
2891 \if@pstarts%
2892 %

```

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

```
2893 \global\advance\l@dpscL \@ne%
2894 \global\advance\l@dpscR \@ne%
2895 \restore@pstartL@pc%
2896 \restore@pstartR@pc%
2897 %
```

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

```
2898 \global\@donereallinesL=\z%
2899 \global\@donetotallinesL=\z%
2900 \global\@donereallinesR=\z%
2901 \global\@donetotallinesR=\z%
2902 %
```

Start a loop over the boxes (chunks).

```
2903 \checkraw@text%
2904 %
2905 % \begingroup
2906 { \loop\ifaraw@text%
2907 %
```

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

```
2908 \checkpageL%
2909 \l@duselanguage{\the\ledlanguageL}%
2910 { \loop\ifl@dsamepage%
2911 %
```

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

```
2912 \ifdefstring{\@eledsectnotoc}{L}{\ledsectnotoc}{}%
2913 \csuse{before@pstartL@the\l@dpscL}%
2914 \global\csundef{before@pstartL@the\l@dpscL}%
2915 \do@lineL%
2916 \xifinlist{\the\l@dpscL}{\eled@sections@@}
2917 {\print@eledsectionL}%
2918 {}%
2919 \advance\numpagelinesL \@ne%
2920 %
```

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


```

2921         \ifshiftedpstarts%
2922             \ifdim\ht\l@dleftbox>0pt%
2923                 \parledgroup@correction@notespacing{L}%
2924                 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2925             \else%
2926                 \xifinlist{\the\l@dpscL}{\eled@sections@@}%
2927                 {\add@apparatusL}%
2928                 {}%
2929                 \unless\ifadvancedshiftedpstarts%
2930                     \dimen0=\pagetotal%
2931                     \advance\dimen0 by \baselineskip%
2932                     \global\pagetotal=\dimen0%
2933                 \else%
2934                     \ifnomaxlines%
2935                         \numdef{\@tmp}{\the\l@dpscL+1}%
2936                         \ifcsdef{minpage@pstart@\@tmp}{%
2937 minpage@pstart@\@tmp}}%
2938                             {\dimen0=\pagetotal%
2939                             \advance\dimen0 by \baselineskip%
2940                             \global\pagetotal=\dimen0%
2941                             }%
2942                             {}%
2943                             }{}%
2944                         \fi%
2945                     \fi%
2946                 \fi%
2947             \else%
2948                 \parledgroup@correction@notespacing{L}%
2949                 \hb@xt@ \hsize{\ledstrutL\unhbox\l@dleftbox}%
2950             \fi%
2951 %

```

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

```

2952         \get@nextboxL%
2953         \global\l@dskipversenumberfalse%
2954         \ifprint@last@after@pendL%
2955             \csuse{after@pendL@\the\l@dpscL}%
2956             \global\csundef{after@pendL@\the\l@dpscL}%
2957         \fi%
2958         \checkpageL%
2959         \checkverseL%
2960         \checkpbL%
2961         \repeat%
2962 %

```

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.

```

2963 \ifl@dpagefull%
2964 \@writelinesonpageL{\the\numpagelinesL}%
2965 \else%
2966 \@writelinesonpageL{1000}%
2967 \fi%
2968 %

```

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.

```

2969 \numpagelinesL \z@%
2970 \parledgroup@correction@notespacing@init%
2971 \clearl@dleftpage }%
2972 %

```

Now do the same for the right text.

```

2973 \checkpageR%
2974 \l@duselanguage{\theledlanguageR}%
2975 {
2976 \loop\ifl@dsamepage%
2977 \set@sectcountR%
2978 \ifdefstring{\@eledsectnotoc}{R}{\ledsectnotoc}{}%
2979 \csuse{before@pstartR@the\l@dpscR}%
2980 \global\csundef{before@pstartR@the\l@dpscR}%
2981 \do@lineR%
2982 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
2983 {\print@eledsectionR}%
2984 {}%
2985 \advance\numpagelinesR \@ne%
2986 \ifshiftedpstarts%
2987 \ifdim\ht\l@drightbox>0pt%
2988 \parledgroup@correction@notespacing{R}%
2989 \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
2990 \else%
2991 \xifinlist{\the\l@dpscR}{\eled@sectionsR@@}%
2992 {\add@apparatusR}%
2993 {}%
2994 \unless\ifadvancedshiftedpstarts%
2995 \dimen0=\pagetotal%
2996 \advance\dimen0 by \baselineskip%
2997 \global\pagetotal=\dimen0%
2998 \else%
2999 \ifnomaxlines%
3000 \numdef{\@tmp}{\the\l@dpscR+1}%
3001 \ifcsdef{minpage@pstart@\@tmp}{%
\ifnumless{\the\c@page}{\csuse{
minpage@pstart@\@tmp}}}%

```

```

3002         {\dimen0=\pagetotal%
3003          \advance\dimen0 by \baselineskip%
3004          \global\pagetotal=\dimen0%
3005         }%
3006     {}%
3007 }{}%
3008 \fi%
3009 \fi%
3010 \fi%
3011 \else%
3012     \parledgroup@correction@notespacing{R}%
3013     \hb@xt@ \hsize{\ledstrutR\unhbox\l@drightbox}%
3014 \fi%
3015 \get@nextboxR%
3016 \global\l@dskipversenumberRfalse%
3017 \ifprint@last@after@pendR%
3018     \csuse{after@pendR@the\l@dpscR}%
3019     \global\csundef{after@pendR@the\l@dpscR}%
3020 \fi%
3021 \checkpageR%
3022 \checkverseR%
3023 \checkpbR%
3024 \repeat%
3025 \ifl@dpagfull%
3026     \@writelinesonpageR{\the\numpagelinesR}%
3027 \else%
3028     \@writelinesonpageR{1000}%
3029 \fi%
3030 \numpagelinesR=\z@%
3031 \parledgroup@correction@notespacing@init%
3032 %

```

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

```

3033     \clearl@drightpage}%
3034 %

```

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.

```

3035     \checkraw@text%
3036     \ifaraw@text%
3037         \getlinesfrompagelistL%
3038         \getlinesfrompagelistR%
3039         \l@dcalc@minoftwo{\@cs@linesonpageL}{\@cs@linesonpageR}%
3040         {\l@dminpagelines}%
3041     \fi%
3042 \repeat}%
3043 %

```

We have now output the text from all the chunks.

```

3044 \fi%
3045 %

```

Make sure that there are no inserts hanging around.

```

3046 \flush@notes%
3047 \flush@notesR%
3048 \endgroup%
3049 %

```

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

```

3050 \global\l@dpscL=\z@%
3051 \global\l@dpscR=\z@%
3052 \global\l@dnumstartsL=\z@%
3053 \global\l@dnumstartsR=\z@%
3054 \global\instanzaLfalse%
3055 \global\instanzaRfalse%
3056 \global\l@dprintingpagesfalse%
3057 %

```

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

```

3058 \ifnum\edtext@later=\edtext@now%
3059 \else%
3060 \led@error@edtext@later@now%
3061 \fi%
3062 \global\edtext@later=\z@%
3063 \global\edtext@now=\z@%
3064 %

```

Prevent final notes from overlapping the line number

```

3065 \finish@Pages@notes%
3066 \ignorespaces}}%
3067
3068
3069 %

```

XVII.3 Ensure all notes are printed at the end of parallel pages

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

```

3070 \newcommand{\finish@Pages@notes}{%
3071 \def\do##1{%
3072 %

```

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

```

3073 \ifnocritical@%
3074 \global\newnamebox{##1footins}%
3075 \fi
3076 \ifnofamiliar@%
3077 \global\newnamebox{footins##1}%
3078 \fi
3079 %

```

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

```

3080 \ifvoid\csuse{##1footins}%
3081 \ifvoid\csuse{footins##1}\else%
3082 \newpage\null%
3083 \listbreak%
3084 \fi%
3085 \else%
3086 \newpage\null%
3087 \listbreak%
3088 \fi%
3089 }%
3090 \dolistloop{\@series}%
3091 }%
3092 %

```

XVII.4 Struts

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

```

\ledstrutR
3093 \newcommand*{\ledstrutL}{\}
3094 \newcommand*{\ledstrutR}{\}
3095
3096 %

```

XVII.5 Page clearing

`\cleartoevenpage` `\cleartoevenpage`, which is defined in the memoir class, is like `\clear(double)page` except that we end up on an even page. `\cleartol@devenpage` is similar except that it first checks to see if it is already on an empty page.

```

3097 \providecommand{\cleartoevenpage}[1][\@empty]{%
3098 \clearpage
3099 \ifodd\c@page\hbox{##1}\clearpage\fi}
3100
3101 \newcommand*{\cleartol@devenpage}{%
3102 \ifdim\pagetotal<\topskip% on an empty page
3103 \else
3104 \clearpage
3105 \Pages@mainmatter%
3106 \fi
3107 \ifodd\c@page%

```

```

3108 \ifprevpgnotnumbered%
3109 \addtocounter{par@page}{-1}%
3110 \fi%
3111 \ifdef{\prevpgstyle}{\thispagestyle{\prevpgstyle}}{%
3112 \hbox{}}\clearpage%
3113 \fi%
3114 }%
3115 %

```

`\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`.

```

3116 \newcommand*{\clearl@dleftpage}{%
3117 \ifdim\pagetotal=0pt\hbox{}\fi%
3118 \newpage%
3119 \insert@notes@for@onlyside%
3120 \ifodd\c@page\else
3121 \led@err@LeftOnRightPage
3122 \hbox{}\fi%
3123 \cleardoublepage
3124 \fi}
3125
3126 \newcommand*{\clearl@drightpage}{%
3127 \ifdim\pagetotal=0pt\hbox{}\fi%
3128 \newpage%
3129 \insert@notes@for@onlyside%
3130 \ifodd\c@page
3131 \led@err@RightOnLeftPage
3132 \hbox{}\fi%
3133 \cleartoevenpage
3134 \fi}
3135
3136 %

```

XVII.6 Lines managing

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

```

\cs@linesinparL
\getlinesfromparlistR
\cs@linesinparR
3137 \newcommand*{\getlinesfromparlistL}{%
3138 \ifx\linesinpar@listL\empty
3139 \gdef\cs@linesinparL{0}%
3140 \else
3141 \gl@p\linesinpar@listL\to\cs@linesinparL

```

```

3142 \fi}
3143 \newcommand*\getlinesfromparlistR{%
3144 \ifx\linesinpar@listR\empty
3145 \gdef\@cs@linesinparR{0}%
3146 \else
3147 \gl@p\linesinpar@listR\to\@cs@linesinparR
3148 \fi}
3149
3150 %

```

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

```

3151 \newcommand*\getlinesfrompagelistL{%
3152 \ifx\linesonpage@listL\empty
3153 \gdef\@cs@linesonpageL{1000}%
3154 \else
3155 \gl@p\linesonpage@listL\to\@cs@linesonpageL
3156 \fi}
3157 \newcommand*\getlinesfrompagelistR{%
3158 \ifx\linesonpage@listR\empty
3159 \gdef\@cs@linesonpageR{1000}%
3160 \else
3161 \gl@p\linesonpage@listR\to\@cs@linesonpageR
3162 \fi}
3163
3164 %

```

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

```

3165 \newcommand*\@writelinesonpageL}[1]{%
3166 \edef\next{\write\linenum@out{\string\@lopL{#1}}}%
3167 \next}
3168 \newcommand*\@writelinesonpageR}[1]{%
3169 \edef\next{\write\linenum@outR{\string\@lopR{#1}}}%
3170 \next}
3171
3172 %

```

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

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

```

3173 \newcommand*\l@dcalc@maxoftwo}[3]{%
3174 \ifnum #2>#1\relax
3175 #3=#2\relax
3176 \else

```

```

3177     #3=#1\relax
3178     \fi}
3179 \newcommand*{\l@dcalc@minoftwo}[3]{%
3180     \ifnum #2<#1\relax
3181         #3=#2\relax
3182     \else
3183         #3=#1\relax
3184     \fi}
3185
3186 %

```

XVII.7 Page break managing

`\ifl@dsamepage` `\l@dsamepagetrue` `\l@dsamepagefalse` `\ifl@dpagfull` `\l@dpagfulltrue` `\l@dpagfullfalse`
`\checkpageL` `\checkpageR`

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

```

3187 \newif\ifl@dsamepage
3188 \l@dsamepagetrue
3189 \newif\ifl@dpagfull
3190
3191 \newcommand*{\checkpageL}{%
3192     \l@dpagfulltrue
3193     \l@dsamepagetrue
3194     \check@goal
3195     \ifdim\pagetotal<\ledthegoal
3196         \ifnum\numpagelinesL<\l@dminpagelines
3197         \else
3198             \ifnomaxlines%
3199             \else%
3200                 \l@dsamepagefalse%
3201                 \l@dpagfullfalse%
3202             \fi%
3203         \fi
3204     \else
3205         \l@dsamepagefalse
3206         \l@dpagfulltrue
3207     \fi%
3208     \ifprint@last@after@pendL%
3209         \l@dpagfullfalse%
3210         \l@dsamepagefalse%
3211         \print@last@after@pendLfalse%
3212     \fi%
3213 }%
3214
3215 \newcommand*{\checkpageR}{%

```



```

3216 \l@dpagetrue
3217 \l@dsamepagetrue
3218 \check@goal
3219 \ifdim\pagetotal<\ledthegoal
3220 \ifnum\numpagelinesR<\l@dmnpagelines
3221 \else
3222 \ifnomaxlines%
3223 \else%
3224 \l@dsamepagefalse%
3225 \l@dpagetruefalse%
3226 \fi%
3227 \fi
3228 \else
3229 \l@dsamepagefalse
3230 \l@dpagetrue
3231 \fi%
3232 \ifprint@last@after@pendR%
3233 \l@dpagetruefalse%
3234 \l@dsamepagefalse%
3235 \print@last@after@pendRfalse%
3236 \fi%
3237 }%
3238
3239 %

```

\checkpbL \checkpbL and \checkpbR are called after each line is printed, and after the page is checked. These commands correct page breaks depending on \ledpb and \lednopb.

```

3240 \newcommand{\checkpbL}{
3241 \IfStrEq{\led@pb@setting}{after}{
3242 \xifinlist{the\absline@num}{\l@prev@pb}{\l@dpagetrue\
3243 \l@dsamepagefalse}{
3244 }{}
3245 \IfStrEq{\led@pb@setting}{before}{
3246 \numdef{\next@absline}{the\absline@num+1}
3247 \xifinlist{\next@absline}{\l@prev@pb}{\l@dpagetrue\
3248 \l@dsamepagefalse}{
3249 \xifinlist{\next@absline}{\l@prev@nopb}{\l@dpagetruefalse\
3250 \l@dsamepagetrue}{
3251 }{}
3252 }
3253 \newcommand{\checkpbR}{
3254 \IfStrEq{\led@pb@setting}{after}{
3255 \xifinlist{the\absline@numR}{\l@prev@pbR}{\l@dpagetrue\
3256 \l@dsamepagefalse}{
3257 \xifinlist{the\absline@numR}{\l@prev@nopbR}{\l@dpagetruefalse\
3258 \l@dsamepagetrue}{

```

```

3256     }{}
3257     \IfStrEq{\led@pb@setting}{before}{
3258       \numdef{\next@abslineR}{\the\absline@numR+1}
3259       \xifinlist{\next@abslineR}{\l@prev@pbR}{\l@dpagetrue\
l@dsamepagefalse}{}
3260       \xifinlist{\next@abslineR}{\l@prev@nopbR}{\l@dpagetrue\
l@dsamepagetrue}{}
3261     }{}
3262   }
3263   %

```

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

```

3264 \newcommand{\checkverseL}{%
3265 \ifinstanzaL%
3266   \iflednopbinverse%
3267     \ifinserthangingsymbol%
3268       \numgdef{\prev@abslineverse}{\the\absline@num-1}%
3269       \IfStrEq{\led@pb@setting}{after}{\lednopbnum{\prev@abslineverse}}{}%
3270       \IfStrEq{\led@pb@setting}{before}{%
3271         \ifnum\numpagelinesL<3%
3272           \ledpbnum{\prev@abslineverse}%
3273         \fi%
3274       }{}%
3275     \fi%
3276   \fi%
3277 }
3278 \newcommand{\checkverseR}{%
3279 \ifinstanzaR%
3280   \iflednopbinverse%
3281     \ifinserthangingsymbolR%
3282       \numgdef{\prev@abslineverse}{\the\absline@numR-1}%
3283       \IfStrEq{\led@pb@setting}{after}{\lednopbnumR{\prev@abslineverse}}{}%
3284       \IfStrEq{\led@pb@setting}{before}{%
3285         \ifnum\numpagelinesR<3%
3286           \ledpbnumR{\prev@abslineverse}%
3287         \fi%
3288       }{}%
3289     \fi%
3290   \fi%
3291 }
3292 \fi%
3293 }
3294 %

```

`\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`.

```

3295 \newdimen\ledthegoal
3296 \ifshiftedpstarts
3297     \newcommand*{\@goalfraction}{0.95}
3298 \else
3299     \newcommand*{\@goalfraction}{0.9}
3300 \fi
3301
3302 \newcommand*{\check@goal}{%
3303     \ledthegoal=\@goalfraction\pagegoal}
3304 \newcommand{\setgoalfraction}[1]{%
3305     \xdef\@goalfraction{#1}%
3306 }
3307 %

```

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

```

\ifwrittenlinesL
3308 \newif\ifwrittenlinesL
3309 \newif\ifwrittenlinesR
3310
3311 %

```

XVII.8 Getting boxes content

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

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

```

3312 \newif\if@getnextbox%
3313 %

```

`\get@nextboxL` If the current box is not empty (i.e., still contains some lines) nothing is done. Otherwise
`\get@nextboxR` if and only if a synchronisation point is reached the next box is started.

```

3314 \newcommand*{\get@nextboxL}{%
3315     \ifvbox\namebox{1@dLcolrawbox\the\1@dpscl}% box is not empty
3316 %

```

The current box is not empty; do nothing.

```

3317 \else%
3318 %

```

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

```

3319 \getnextboxtrue%Should be local, but be cautious
3320 %

```

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

```

3321 \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscL}>\@donetotallinesL
3322 \parledgroup@notes@endL%
3323 \unless\ifnosyncpstarts%
3324 \@getnextboxfalse%
3325 %

```

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.

```

3326 \ifnomaxlines%
3327 \ifdim\pagetotal<\ledthegoal%
3328 \numdef{\@tmp}{\l@dpscL+1}%
3329 \ifcsdef{afterlines@pstart@\@tmp R}{%
3330 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp R}}%
3331 %
3332 {}%
3333 {\ifcsdef{minpage@pstart@\@tmp}%
3334 {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
3335 {\ifnum\numpagelinesL=1@dminpagelines%
3336 \getnextboxtrue%
3337 \fi%
3338 }%
3339 {\@getnextboxtrue}}%
3340 {\@getnextboxtrue}%
3341 }%
3342 {}%
3343 \fi%
3344 \fi%
3345 \fi%
3346 \else%
3347 \ifnomaxlines%
3348 \numdef{\@tmp}{\the\l@dpscL+1}%
3349 \ifcsdef{minpage@pstart@\@tmp}{%
3350 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}%
3351 {\ifdimgreater{\pagetotal}{\ledthegoal}%
3352 {\@getnextboxtrue}%
3353 {\@getnextboxfalse}%
3354 }%
3355 {\@getnextboxtrue}%
3356 }{}%
3357 \fi%
3358 \fi%
3359 %

```

Sufficient lines have been output.

```

3360 \if@getnextbox%

```

```

3361 \ifnum\usernamecount{l@dmaxlinesinpar\the\l@dpscl}=\@donetotallinesL
3362 \parledgroup@notes@endL
3363 \fi
3364 \ifwrittenlinesL\else
3365 %

```

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

```

3366 \@writelinesinparL
3367 \writtenlinesLtrue
3368 \fi
3369 \ifnum\l@dnumstartsL>\l@dpscl
3370 %

```

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.

```

3371 \writtenlinesLfalse
3372 \ifbypstart@
3373 \global\line@num=0%
3374 \resetprevline@%
3375 \fi
3376 % Add the content of the optional argument of the previous \protect\cs{pend
3377 }.
3378 % \begin{macrocode}
3379 \csuse{after@pendL@\the\l@dpscl}%
3380 \global\csundef{after@pendL@\the\l@dpscl}%
3381 %

```

Check the number of lines

```

3381 \l@dcalc@maxoftwo{\the\usernamecount{l@dmaxlinesinpar\the\l@dpscl}}%
3382 {\the\@donetotallinesL}%
3383 {\usernamecount{l@dmaxlinesinpar\the\l@dpscl}}%
3384 \global\@donetotallinesL \z@
3385 %

```

Go to the next pstart

```

3386 \global\advance\l@dpscl \@ne
3387 \global\pstartnumtrue%
3388 \restore@pstartL@pc%
3389 %

```

Add notes of parallel ledgroup.

```

3390 \parledgroup@notes@endL
3391 \parledgroup@correction@notespadding@final{L}
3392 \else
3393 %
3394 \print@last@after@pendLtrue%
3395 \fi

```

```

3396 \fi
3397 \fi}
3398 %

3399 \newcommand*{\get@nextboxR}{%
3400 \ifvbox\namebox{1@dRcolrawbox\the\l@dpscR}% box is not empty
3401 \else% box is empty
3402 \@getnextboxtrue%
3403 \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscR}>\@donetotallinesR
3404 \parledgroup@notes@endR
3405 \unless\ifnosyncpstarts%
3406 \@getnextboxfalse%
3407 \ifnomaxlines%
3408 \ifdim\pagetotal<\ledthegoal%
3409 \numdef{\@tmp}{\l@dpscR+1}%
3410 \ifcsdef{afterlines@pstart@\@tmp L}{%
3411 \ifnumless{\numpagelinesL}{\csuse{afterlines@pstart@\@tmp L}}
3412 {}}%
3413 {\ifcsdef{minpage@pstart@\@tmp}%
3414 {\ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
3415 {\ifnum\numpagelinesR=\l@dminpagelines%
3416 \@getnextboxtrue%
3417 \fi%
3418 }%
3419 {\@getnextboxtrue}}}%
3420 {\@getnextboxtrue}%
3421 }%%
3422 }%
3423 {}%
3424 \fi%
3425 \fi%
3426 \fi%
3427 \else%
3428 \ifnomaxlines%
3429 \numdef{\@tmp}{\the\l@dpscR+1}%
3430 \ifcsdef{minpage@pstart@\@tmp}{%
3431 \ifnumless{\the\c@page}{\csuse{minpage@pstart@\@tmp}}}%
3432 {\ifdimgreater{\pagetotal}{\ledthegoal}%
3433 {\@getnextboxtrue}%
3434 {\@getnextboxfalse}%
3435 }%
3436 {\@getnextboxtrue}%
3437 }-{}
3438 \fi%
3439 \fi%
3440 \if@getnextbox%
3441 \ifnum\usernamecount{1@dmaxlinesinpar\the\l@dpscR}=\@donetotallinesR
3442 \parledgroup@notes@endR
3443 \fi

```

```

3444 \ifwrittenlinesR\else
3445 \@writelinesinparR
3446 \writtenlinesRtrue
3447 \fi
3448 \ifnum\l@dnumpstartsR>\l@dpscR
3449 \writtenlinesRfalse
3450 \ifbypstartR
3451 \global\line@numR=0%
3452 \resetprevline@%
3453 \fi
3454 \csuse{after@pendR@the\l@dpscR}%
3455 \global\csundef{after@pendR@the\l@dpscR}%
3456 \l@dcalc@maxoftwo{\the\usernamecount{l@dmaxlinesinpar\the\l@dpscR}}%
3457 {\the\@donetotallinesR}%
3458 {\usernamecount{l@dmaxlinesinpar\the\l@dpscR}}%
3459 \global\@donetotallinesR \z@
3460 \global\advance\l@dpscR \@ne
3461 \global\pstartnumRtrue%
3462 \restore@pstartR@pc%
3463 \parledgroup@notes@endR
3464 \parledgroup@correction@notespacing@final{R}
3465 \else
3466 \print@last@after@pendRtrue%
3467 \fi
3468 \fi
3469 \fi}
3470
3471 %

```

XVIII Page numbering

XVIII.1 Global options

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

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

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

`\par@patch@thepage` `\par@patch@thepage` patches `\thepage` in order to use the value of `par@page` counter and not the value of `page`. It must be called after any redefinition of `\thepage`.

That is why we insert it at the end of the \LaTeX macro `\pagenumbering`, which is called by some `\xxxmatter` commands. In cases when we are using the `memoir` class, we insert it at the end of `\@mempnum`. When using `\pagenumbering`, we also need to restart `par@page` counter. Consequently, we have wrapped `\par@patch@thepage` and counter restart in `\par@patch@pagenumbering`. We also call `\par@patch@thepage` it at the beginning of the document.

```

3472
3473 \newcommand{\par@patch@thepage}{%
3474   \ifboolexpr{%
3475     bool{sameparallelpagenumber}%
3476     or bool{prevpgnotnumbered}%
3477   }%
3478   {%
3479     \patchcmd{\thepage}%
3480       {page}{par@page}%
3481       {}{}%
3482     {\led@error@fail@patch@thepage}%
3483   }{}%
3484 }%
3485
3486 \newcommand{\par@patch@pagenumbering}{%
3487   \ifboolexpr{%
3488     bool{sameparallelpagenumber}%
3489     or bool{prevpgnotnumbered}%
3490   }%
3491   {%
3492     \setcounter{par@page}{1}%
3493   }%
3494   {}%
3495   \par@patch@thepage%
3496 }%
3497
3498 \ifl@dmemoir%
3499   \apptocmd{\@mempnum}%
3500     {\par@patch@pagenumbering}%
3501     {}{}%
3502     {\led@error@fail@patch@@mempnum}%
3503
3504 \else%
3505   \apptocmd{\pagenumbering}%
3506     {\par@patch@pagenumbering}%
3507     {}{}%
3508     {\led@error@fail@patch@pagenumbering}%
3509 \fi%
3510
3511 \AtBeginDocument{\par@patch@thepage}%
3512 %

```

`\@outputpage` As its name says, `\@outputpage` is a \LaTeX 's macro called in the output routine. It is this

macro which increases the page counter.. We patch it in order to increase, conditionally, the `par@page` counter.

```

3513 \AtBeginDocument{%
3514   \apptocmd{\@outputpage}{%
3515     \ifsameparallelpagenumbers%
3516       \ifl@dprintingpages%
3517         \ifodd\c@page\else%
3518           \stepcounter{par@page}%
3519         \fi%
3520       \else%
3521         \stepcounter{par@page}%
3522       \fi%
3523     \else%
3524       \stepcounter{par@page}%
3525     \fi%
3526   }%
3527   {}%
3528   {\led@error@fail@patch@{@outputpage}}%
3529 }
3530 %

```

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

```

3531 \newcounter{par@page}%
3532 \setcounter{par@page}{1}%
3533 %

```

XVIII.2 *mainmatter* option of `\Pages`

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

```

\ifPages@mainmatter34 \newif\ifPages@mainmatter
\Pages@mainmatter35 \newcommand{\Pages@mainmatter}{%
3536   \ifPages@mainmatter%
3537     \pagenumbering{arabic}%
3538     \addtocounter{page}{1}%
3539     \addtocounter{par@page}{-1}%
3540     \patchcmd{\thepage}{page}{par@page}{}{}%
3541   \fi%
3542 }
3543 %

```

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 3544 \pretocmd{\chapter}{%
3545 \ifl@printingpages%
3546 \vbox{}%
3547 \fi%
3548 }%
3549 {}%
3550 {}%
3551 %
```

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

```
3552 \newcommand{\eledsectnotoc}[1]{\xdef\@eledsectnotoc{#1}}
3553 \eledsectnotoc{R}
3554 %
```

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

```
3555 \newcommand{\eledsectmark}[1]{\xdef\@eledsectmark{#1}}
3556 \eledsectmark{L}
3557 %
```

`\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`.

```
3558 \newskip\eledsection@correcting@skip
3559 %
```

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

```
3560 \newwrite\eled@sectioningR@out
3561 %
```

XX Page break/no page break, depending on the specific line

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

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

```

3562 \def\l@prev@pbR{}
3563 \def\l@prev@nopbR{}
3564 %

```

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

```

3565 \newcommand{\ledpbR}{\write\linenum@outR{\string\led@pbR}}
3566 \newcommand{\ledpbnumR}[1]{\write\linenum@outR{\string\led@pbnumR{#1}}}
3567 \newcommand{\lednopbR}{\write\linenum@outR{\string\led@nopbR}}
3568 \newcommand{\lednopbnumR}[1]{\write\linenum@outR{\string\led@nopbnumR{#1}}}
3569 %

```

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

```

3570 \newcommand{\led@pbR}{\listxadd{\l@prev@pbR}{\the\absline@numR}}
3571 \newcommand{\led@pbnumR}[1]{\listxadd{\l@prev@pbR}{#1}}
3572 \newcommand{\led@nopbR}{\listxadd{\l@prev@nopbR}{\the\absline@numR}}
3573 \newcommand{\led@nopbnumR}[1]{\listxadd{\l@prev@nopbR}{#1}}
3574 %

```

XXI Parallel ledgroup

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

```

3575 \newmarks\parledgroup@
3576 \newmarks\parledgroup@series
3577 \newmarks\parledgroup@type
3578 %

```

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

```

3579 \newcommand{\parledgroup@notes@startL}{%
3580 \ifnum\usenamecount{l@dmxlinesinpar\the\l@dpscl}>0%
3581 \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
bhooknoteX@\splitfirstmarks\parledgroup@series}}{}%
3582 \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
bhookXnote@\splitfirstmarks\parledgroup@series}}{}%
3583 \fi%

```

```

3584 \global\ledgroupnotesL@true%
3585 \insert@noterule@ledgroup{L}%
3586 }
3587 \newcommand{\parledgroup@notes@startR}{%
3588 \ifnum\usenamecount{1@dmaxlinesinpar\the\l@dpscR}>0%
3589 \IfStrEq{\splitfirstmarks\parledgroup@type}{footnoteX}{\csuse{
bhooknoteX@\splitfirstmarks\parledgroup@series}}{}%
3590 \IfStrEq{\splitfirstmarks\parledgroup@type}{Xfootnote}{\csuse{
bhookXnote@\splitfirstmarks\parledgroup@series}}{}%
3591 \fi%
3592 \global\ledgroupnotesR@true%
3593 \insert@noterule@ledgroup{R}%
3594 }
3595 %

```

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

```

3596 \newcommand{\parledgroup@notes@endL}{%
3597 \global\ledgroupnotesL@false%
3598 }
3599 \newcommand{\parledgroup@notes@endR}{%
3600 \global\ledgroupnotesR@false%
3601 }
3602 %

```

`\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`

```

3603 \newcommand{\insert@noterule@ledgroup}[1]{
3604 \IfStrEq{\splitbotmarks\parledgroup@}{begin}{%
3605 \IfStrEq{\splitbotmarks\parledgroup@type}{Xfootnote}{
3606 \csuse{ifledgroupnotes#1@}
3607 \vskip\skip\csuse{mp\splitbotmarks\parledgroup@series footins}
3608 \csuse{\splitbotmarks\parledgroup@series footnoterule}
3609 \fi
3610 }
3611 {}
3612 \IfStrEq{\splitbotmarks\parledgroup@type}{footnoteX}{
3613 \csuse{ifledgroupnotes#1@}
3614 \vskip\skip\csuse{mpfootins\splitbotmarks\parledgroup@series}
3615 \csuse{footnoterule\splitbotmarks\parledgroup@series}
3616 \fi
3617 }{}
3618 }
3619 {}
3620 }
3621 %

```

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

```

3622 \newcommand{\setparledgroupnotespacing}[1]{\gdef\@parledgroupnotespacing
    {#1}}
3623 \newcommand{\@parledgroupnotespacing}{}
3624 %

```

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

```

3625 \dimdef{\parledgroup@notespacing@correction}{0pt}
3626 \newcommand{\parledgroup@notespacing@set@correction}{%
3627   {\@getfirstseries\csuse{Xnotefontsize@\@firstseries}%We suppose all the
    series has the same footnote size setup
3628   \@parledgroupnotespacing\dimgdef{\temp@spacing}{\baselineskip}}%
3629   \dimgdef{\parledgroup@notespacing@correction}{\baselineskip-\temp@spacing}
    }%
3630 }
3631 %

```

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

```

3632 \newcommand{\parledgroup@correction@notespacing@init}{
3633   \dimdef{\parledgroup@notespacing@correction@accumulated}{0pt}
3634   \dimdef{\parledgroup@notespacing@correction@modulo}{0pt}
3635 }
3636 \parledgroup@correction@notespacing@init
3637 %

```

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

```

3638 \newcommand{\parledgroup@correction@notespacing@final}[1]{
3639   \ifparledgroup
3640   \vspace{\parledgroup@notespacing@correction@accumulated}
3641   \parledgroup@correction@notespacing@init%
3642   \ifstrequal{#1}{L}{
3643     \numdef{\@checking}{\the\l@dpscL-1}
3644   }{
3645     \numdef{\@checking}{\the\l@dpscR-1}
3646   }
3647   \dimdef{\@beforenotes@current@diff}{\csuse{\@parledgroup@beforenotes@
    \@checking L}-\csuse{\@parledgroup@beforenotes@\@checking R}}%
3648   \ifstrequal{#1}{L}%

```

```

3649     {% Left
3650       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{-\
@beforenotes@current@diff}}%
3651     }%
3652     {% Right
3653       \ifdimgreater{\@beforenotes@current@diff}{0pt}{\vspace{\
@beforenotes@current@diff}}{}
3654     }%
3655     \fi
3656   }
3657   %

```

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

```

3658 \newcommand{\parledgroup@correction@notespacing}[1]{%
3659   \csuse{ifledgroupnotes#10}%
3660   \vspace{-\parledgroup@notespacing@correction}%
3661   \dimdef{\parledgroup@notespacing@correction@accumulated}{\
parledgroup@notespacing@correction@accumulated+\
parledgroup@notespacing@correction}%
3662   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo+\
parledgroup@notespacing@correction}%
3663   \ifdimless{\parledgroup@notespacing@correction@modulo}{\baselineskip
}{\advance\numpagelinesL -\@ne%
3664   \dimdef{\parledgroup@notespacing@correction@modulo}{\
parledgroup@notespacing@correction@modulo-\baselineskip}%
3665   }% mean greater than equal
3666   \fi%
3667 }
3668 %

```

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

```

3669 \dimdef\parledgroup@beforenotesL{0pt}
3670 \dimdef\parledgroup@beforenotesR{0pt}
3671 %

```

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

```

3672 \newcommand{\parledgroup@beforenotes@save}[1]{
3673   \ifparledgroup
3674     \csdimgdef{@parledgroup@beforenotes@the\csuse{1@dnumstarts#1}#1}{\
csuse{parledgroup@beforenotes#1}}
3675     \csdimgdef{parledgroup@beforenotes#1}{Opt}
3676   \fi
3677 }
3678 %

```

XXII Compatibility with eledmac

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

```

3679 \ifeledmaccompat%
3680
3681
3682   \unless\ifnocritical@
3683   \let\onlyXside\Xonlyside
3684   \fi
3685 \fi
3686 %

```

XXIII The End

</code>

Appendix A Some things to do when changing version

Appendix A.1 Migration to eledpar 1.4.3

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

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

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

See the following two examples:

With standard `\hangingsymbol`:

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

With the modification of the `hangingsymbol`:

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

Appendix A.2 Migration from eledpar to reledpar

As for migration from eledmac to reledmac:

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

Appendix A.2.1 Deprecated options

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

Appendix A.2.2 `\renewcommand` replaced with command

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

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

Appendix A.2.3 Commands the names of which have changed

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

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

Appendix A.3 Migration to reledpar 2.2.0

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

Appendix A.4 Migration to reledpar 2.3.0

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

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

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

Appendix A.5 Migration to reledpar 2.4.0

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

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

Appendix A.6 Migration to reledpar 2.5.0

If you use either `\stanza` or *astanza* environment, please read Appendix A.12 p. 370.

Appendix A.7 Migration to reledpar 2.6.0

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

Appendix A.8 Migration to reledpar 2.6.1

If you use `perpage` package to control footnote numbering, please read the handbook on 6.3.3 p. 15.

Appendix A.9 Migration to reledpar 2.14.0

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

Appendix A.10 Migration to reledpar 2.14.1

The execution of the code sent to `\doinsidelineLhook` and `\doinsidelineRhook` has been moved to a place where `\markboth` can work.

Normally, it should not interfere with your prior use of `\doinsidelineLhook` and `\doinsidelineRhook`.

Please contact us if that causes problems.

Appendix A.11 Migration to v. 2.17.2

If you change the `\Lcolwidth` and `\Rcolwidth` length inside the `pages` environment, the contents were aligned to right. Now, it is aligned to left. Please contact us if you need to keep the older behavior.

References

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

Index

Symbols

<code>\@adv</code>	1
<code>\@astanza@line</code>	1
<code>\@cs@linesinparL</code>	1
<code>\@cs@linesinparR</code>	1
<code>\@cs@linesonpageL</code>	1
<code>\@cs@linesonpageR</code>	1
<code>\@donereallinesL</code>	1
<code>\@donereallinesR</code>	1
<code>\@donetotallinesL</code>	1
<code>\@donetotallinesR</code>	1
<code>\@eledsectionL</code>	1

\@eledsectionR	1
\@lab	1
\@lopL	1
\@lopR	1
\@morespace@leftnote@rightcolumn	1
\@morespace@rightnote@leftcolumn	1
\@nl	1
\@nl@regR	1
\@outputpage	1
\@par@sync@option	1
\@par@this@sync@option	1
\@parledgroupnotespacing	1
\@pend	1
\@pendR	1
\@pstart	1
\@pstartR	1
\@pstartsfalse	1
\@pstartstrue	1
\@ref	1
\@ref@reg@parseR	1
\@ref@regR	1
\@set	1
\@stopastanza	1
\@writelinesinparL	1
\@writelinesinparR	1
\@writelinesonpageL	1
\@writelinesonpageR	1
\@writepageofparL	1
\@writepageofparR	1
CLASSmemoir	128
COMMAND\+	55
COMMAND\@Rlineflag	87, 165
COMMAND\@adv	46, 164
COMMAND\@cs@linesinparL	118
COMMAND\@cs@linesonpageL	119
COMMAND\@eledsectionL	98
COMMAND\@eledsectionR	98
COMMAND\@eledsectmark	130
COMMAND\@eledsectnotoc	130
COMMAND\@goalfraction	13, 122
COMMAND\@l@dttempcnta	81
COMMAND\@lab	44, 88, 164
COMMAND\@lopL	50, 112, 119
COMMAND\@lopR	50, 119
COMMAND\@mempnum	128
COMMAND\@morespace@leftnote@rightcolumn	88, 89
COMMAND\@morespace@rightnote@leftcolumn	88, 89
COMMAND\@namedef	31
COMMAND\@namuse	31
COMMAND\@nl	44, 51, 88, 164

COMMAND\@nl@regR	44
COMMAND\@outputpage	127, 128
COMMAND\@page	88
COMMAND\@par@sync@option	44
COMMAND\@parledgroupnotespadding	133
COMMAND\@pend	49
COMMAND\@pendR	49
COMMAND\@pstart	49
COMMAND\@pstartstrue	97
COMMAND\@ref	47–49, 52, 164
COMMAND\@ref@later	48
COMMAND\@ref@reg@parseargR	48
COMMAND\@ref@regR	47
COMMAND\@set	46, 164
COMMAND\@sw	48
COMMAND\AtBeginPairs	9, 58, 162, 166
COMMAND\AtEveryPend	10, 162–164
COMMAND\AtEveryPend*	10
COMMAND\AtEveryPstart	2, 10, 20, 66, 99, 162–164
COMMAND\AtEveryPstart*	10
COMMAND\AtEveryPstartCall	2, 20, 66, 163
COMMAND\AtEveryStanza	166
COMMAND\AtEveryStopStanza	166
COMMAND\Clear the right lines for \read@linelist	43
COMMAND\Columns	8, 20, 56, 98, 107, 159–161, 163, 165
COMMAND\Columns@print@after@pend	108
COMMAND\Columns@print@before@pstart	108
COMMAND\Lcolwidth	9, 11, 110, 138
COMMAND\Lcolwidthth	167
COMMAND\Leftsidehook	159
COMMAND\Leftsidehookend	159
COMMAND\MakePerPage	15
COMMAND\Pages	4, 8, 11, 13, 14, 20, 28, 55, 56, 78, 82–84, 110, 116, 127, 129, 133, 159, 161, 163, 165, 167
COMMAND\Pages@mainmatter	129
COMMAND\Pairs	55
COMMAND\Rcolwidth	9, 11, 110, 138
COMMAND\Rcolwidthth	167
COMMAND\Rightsidehook	159
COMMAND\Rightsidehookend	159
COMMAND\Rlineflag	136
COMMAND\Xendlineflag	16, 165
COMMAND\Xlineflag	16, 137, 165
COMMAND\Xmaxhnotes	14
COMMAND\Xnoteswidthliketwocolumns	10, 162
COMMAND\Xonlyside	15, 82, 137, 166
COMMAND\Xtxtbeforenotes	166
COMMAND\&	21
COMMAND\absline@numR	43
COMMAND\add@apparatusL	70

COMMAND\add@apparatusR	72
COMMAND\add@penalties	81
COMMAND\add@penaltiesL	80
COMMAND\advanceline	46, 52, 164
COMMAND\affixline@num	75
COMMAND\affixline@numR	75, 159, 160
COMMAND\affixpstart@num	78
COMMAND\affixpstart@numR	78
COMMAND\affixside@note	89
COMMAND\aftercolumnseparator	9, 107, 161, 167, 168
COMMAND\araw@textfalse	97
COMMAND\araw@texttrue	97
COMMAND\at@begin@pairs	58
COMMAND\autopar	20
COMMAND\ballast@count	81
COMMAND\baselineskip	106, 134
COMMAND\bbI@set@language	93, 94, 164
COMMAND\beforecolumnseparator	9, 107, 161, 168
COMMAND\begin	21
COMMAND\beginnumbering	10, 16–18, 20, 33, 44, 107, 159–161, 163, 168
COMMAND\beginnumberingR	37, 51
COMMAND\bf	160
COMMAND\bfseries	160
COMMAND\brokenpenalty	81
COMMAND\chapter	130, 159
COMMAND\check@goal	122
COMMAND\check@pstarts	97
COMMAND\checkpageL	112, 120
COMMAND\checkpb@columns	106
COMMAND\checkpbL	121
COMMAND\checkpbR	121
COMMAND\checkraw@text	97
COMMAND\checkverseL	122
COMMAND\checkverseR	122
COMMAND\clear(double)page	117
COMMAND\clearl@dleftpage	118
COMMAND\clearl@drighthpage	118
COMMAND\clearpage	118, 163
COMMAND\cleartoevenpage	117
COMMAND\cleartol@devenpage	117, 129
COMMAND\columnrulewidth	9, 106
COMMAND\columns	167
COMMAND\columns@position	107
COMMAND\columnseparator	9
COMMAND\columnspostion	9, 161
COMMAND\continuousnumberingwithcolumns	168
COMMAND\correct@Xfootins@box	163
COMMAND\correct@footinsX@box	163
COMMAND\critext	162
COMMAND\csname	57

COMMAND\displaywidowpenalty	80
COMMAND\do@actions	73
COMMAND\do@actions@fixedcode	159
COMMAND\do@actions@nextR	73
COMMAND\do@actionsR	73, 159
COMMAND\do@ballast	81
COMMAND\do@ballastR	73
COMMAND\do@insidelineLhook	161
COMMAND\do@insidelineRhook	161
COMMAND\do@line	67
COMMAND\do@line(L/R)	70
COMMAND\do@lineL	67, 80, 159, 160
COMMAND\do@lineLhook	159
COMMAND\do@lineR	70, 159, 160, 162
COMMAND\do@lineRhook	159
COMMAND\do@lockoff	164
COMMAND\do@lockoffR	47
COMMAND\do@lockon	164
COMMAND\do@lockonR	46
COMMAND\doinsidelineLhook	21, 138, 162, 167
COMMAND\doinsidelineRhook	21, 138, 162, 167
COMMAND\doinsidelinehook	21
COMMAND\dolineLhook	21, 162
COMMAND\dolineRhook	21, 162
COMMAND\dolinehook	21
COMMAND\edindex	162
COMMAND\edlabel	160, 162, 163
COMMAND\edlineref	167
COMMAND\edtext	16, 47, 52, 86, 162, 166, 167
COMMAND\edtext@later	85, 87, 116
COMMAND\edtext@now	116
COMMAND\edtextlater	16, 85, 86, 166, 167
COMMAND\edtextnow	16, 85–87, 166, 167
COMMAND\eled@sectioningR@out	130
COMMAND\eledchapter	163
COMMAND\eledsection	162–164
COMMAND\eledsection@correcting@skip	130
COMMAND\eledsectmark	23, 130
COMMAND\eledsectnotoc	23, 130
COMMAND\eledxxx	162
COMMAND\end	21
COMMAND\endgraf	64
COMMAND\endlock	52, 164
COMMAND\endnumbering	10, 17, 20, 34, 159, 163, 168
COMMAND\endnumberingR	51
COMMAND\endsub	52, 164
COMMAND\endumbering	17
COMMAND\expandafter	54
COMMAND\extensionchars	32
COMMAND\firstlinenum	18, 161, 164, 165

COMMAND\firstsublinenum	161, 164, 165
COMMAND\fix@page	45, 164
COMMAND\flag@end	52, 159, 162
COMMAND\flag@start	52, 162
COMMAND\flush@notesR	81
COMMAND\footnote	56
COMMAND\footnoteX	53, 54
COMMAND\footnoteX@reading	165
COMMAND\footnoteXmk	15
COMMAND\footnoteXnomk	15, 54
COMMAND\frontmatter	14, 24
COMMAND\get@intercolumns@width	101, 105
COMMAND\get@nextboxL	160
COMMAND\get@nextboxR	160
COMMAND\get@sidenote@morespace@columns	88, 89
COMMAND\getline@numL	72
COMMAND\getline@numR	72
COMMAND\getlinesfrompagelistL	119
COMMAND\getlinesfrompagelistR	119
COMMAND\getlinesfromparlistL	118
COMMAND\getlinesfromparlistR	118
COMMAND\gl@p	54
COMMAND\goalfraction	136
COMMAND\hangingsymbol	136, 160
COMMAND\hfill	101, 107
COMMAND\hidenumbering	18, 164
COMMAND\if@getnextbox	123
COMMAND\if@sidenotesmarginpage	88, 89
COMMAND\ifPages@mainmatter	129
COMMAND\ifbypage@	164
COMMAND\ifbypstart@R	164
COMMAND\ifcsboxvoid	32
COMMAND\ifdim	101
COMMAND\ifinserthangingsymbol	90
COMMAND\ifinserthangingsymbolR	90
COMMAND\ifl@dpagefull	120
COMMAND\ifl@dpaging	28, 162
COMMAND\ifl@dpairing	28, 159
COMMAND\ifl@dsamelang	162
COMMAND\ifl@dsamepage	120
COMMAND\ifl@pagefull	112
COMMAND\ifledRcol	28
COMMAND\iflinenumberLevenifblank	42
COMMAND\iflinenumberRevenifblank	42
COMMAND\iflledRcol	160
COMMAND\ifmovecolumnspostiononrightpage	27
COMMAND\ifnumberedpar@	60
COMMAND\ifnumberingR	160
COMMAND\ifnumberpstart	56
COMMAND\ifpst@rtedL	33, 34, 61, 159

COMMAND\ifpst@rtedR	33
COMMAND\ifsublines@	46
COMMAND\ifsublines@R	43
COMMAND\insert@countR	47
COMMAND\insert@noterule@ledgroup	132
COMMAND\insert@notes@for@onlyside	84
COMMAND\insertlines@list	47
COMMAND\insertlines@listR	47
COMMAND\inserts@list	60
COMMAND\inserts@listR	80
COMMAND\intercolumns@width	101
COMMAND\l@d@nums	52
COMMAND\l@d@set	46, 52, 164
COMMAND\l@dLcolrawbox	60
COMMAND\l@dLcolrawbox1	96
COMMAND\l@dLcolrawbox2	96
COMMAND\l@dRcolrawbox	60
COMMAND\l@dcalc@maxoftwo	119
COMMAND\l@dcalc@minoftwo	119
COMMAND\l@dchecklang	159, 161
COMMAND\l@dcsnote	161
COMMAND\l@dleftbox	67, 112, 163
COMMAND\l@dlinenumR	42, 159
COMMAND\l@dlsnote	161
COMMAND\l@dmake@labels	87
COMMAND\l@dmaxlinesinpar	111
COMMAND\l@dmaxlinesinpar1	96
COMMAND\l@dminpagelines	111, 159
COMMAND\l@dnumppstartsL	95, 159
COMMAND\l@dprintingcolumnstrue	162
COMMAND\l@dprintingpagestrue	162
COMMAND\l@dpscL	99, 111, 125
COMMAND\l@dpscR	99, 111
COMMAND\l@drsnote	161
COMMAND\l@dsetupmaxlinecounts	96
COMMAND\l@duselanguage	94, 159
COMMAND\l@dzeromaxlinecounts	96
COMMAND\l@prev@nopbR	130
COMMAND\l@prev@pbR	130
COMMAND\labelpstarttrue	160
COMMAND\labelref@list	88
COMMAND\labelref@listR	87
COMMAND\lang	93
COMMAND\last@page@numR	45
COMMAND\led	160
COMMAND\led@nopbR	131
COMMAND\led@nopbnumR	131
COMMAND\led@pbR	131
COMMAND\led@pbnumR	131
COMMAND\ledinnernote	22

COMMAND\ledleftnote	22
COMMAND\lednopb	22, 106, 121
COMMAND\lednopbR	131
COMMAND\lednopbnumR	131
COMMAND\ledouternote	22
COMMAND\ledpb	106, 121
COMMAND\ledpbR	131
COMMAND\ledpbnumR	131
COMMAND\ledrightnote	22
COMMAND\ledsidenote	22
COMMAND\ledstrutL	159
COMMAND\ledstrutR	159, 165
COMMAND\ledthegoal	122
COMMAND\ledtrutL	159, 165
COMMAND\leftlinenumR	42, 159
COMMAND\lemma	86
COMMAND\let	54
COMMAND\line@list@R	48
COMMAND\line@list@stuff	44, 51
COMMAND\line@margin	38
COMMAND\line@marginR	38, 159
COMMAND\line@numR	43
COMMAND\lineation	18, 163, 168
COMMAND\lineation*	18, 38, 162
COMMAND\lineationR	18, 37, 163, 166
COMMAND\linenum	86
COMMAND\linenum@out	88
COMMAND\linenum@outR	50
COMMAND\linenumOnlyPagesForColumns	19, 39, 168
COMMAND\linenumOnlyPagesForColumnsR	19, 39, 168
COMMAND\linenumberLevenifblanktrue	19, 166
COMMAND\linenumberRevenifblank	166
COMMAND\linenumberRevenifblanktrue	19
COMMAND\linenumberlist	18, 138, 166
COMMAND\linenumberlistR	18, 138, 166
COMMAND\linenumberstyle	18, 137, 167
COMMAND\linenumberstyle*	137
COMMAND\linenumberstyleR	18, 137
COMMAND\linenumincrement	18, 161, 164, 165
COMMAND\linenummargin	18, 19, 38, 159, 164–166
COMMAND\linenummargin*	18, 38, 165
COMMAND\linenummarginColumns	19, 39
COMMAND\linenummarginColumns*	19
COMMAND\linenummarginColumnsR	19
COMMAND\linenummarginR	18, 19, 38, 165
COMMAND\linenumrepR	41, 159
COMMAND\linesinpar@listL	49, 118
COMMAND\linesonpage@listL	50, 119
COMMAND\lock@off	47
COMMAND\lock@on	46

COMMAND\mainmatter	2, 14, 24, 165
COMMAND\makeatletter	70
COMMAND\markboth	138, 167
COMMAND\maxchunks	8, 21, 96
COMMAND\maxdimen	82, 83
COMMAND\maxhnotesX	14
COMMAND\memorydump	17, 36
COMMAND\movecolumnspostiononrightpage	105
COMMAND\msdata	159
COMMAND\n@num	163
COMMAND\new@lineL	51
COMMAND\new@lineR	51
COMMAND\newhookcommand@series	55
COMMAND\newif	163
COMMAND\newpage	117, 118, 163
COMMAND\newseries	56
COMMAND\newseries@	52
COMMAND\newseries@par	52, 55, 56
COMMAND\next@line@list@stuffR	51, 168
COMMAND\noeledxxx	162
COMMAND\nomark@	54
COMMAND\nomaxlines	49
COMMAND\notesXwidthliketwocolumns	10, 162
COMMAND\num@lines	81
COMMAND\num@lines(R)	60
COMMAND\numberingR	35
COMMAND\numberlinefalse	7
COMMAND\numberonlyfirstinline	160
COMMAND\numberpstartfalse	17, 168
COMMAND\numberpstarttrue	17, 160, 165
COMMAND\one@line	60, 90
COMMAND\one@lineR	60
COMMAND\onlyXside	137
COMMAND\onlysideX	15, 82, 84, 163, 166
COMMAND\otherlanguage	164
COMMAND\page@action	46, 164
COMMAND\pagenumbering	128, 164
COMMAND\pages	14
COMMAND\pagetotal	112, 163
COMMAND\par@line	81
COMMAND\par@line(R)	60
COMMAND\par@patch@pagenumbering	128
COMMAND\par@patch@thepage	127, 128
COMMAND\par@sync@option	26
COMMAND\parledgroup@	131
COMMAND\parledgroup@beforenotes@save	135
COMMAND\parledgroup@beforenotesL	134
COMMAND\parledgroup@beforenotesR	134
COMMAND\parledgroup@correction@notespacing	134
COMMAND\parledgroup@correction@notespacing@final	133

COMMAND\parledgroup@correction@notespacing@init	133
COMMAND\parledgroup@notes@endL	132
COMMAND\parledgroup@notes@endR	132
COMMAND\parledgroup@notes@startL	131
COMMAND\parledgroup@notes@startR	131
COMMAND\parledgroup@notespacing@correction	133, 134
COMMAND\parledgroup@notespacing@correction@accumulated	134
COMMAND\parledgroup@notespacing@correction@modulo	134
COMMAND\parledgroup@notespacing@set@correction	133
COMMAND\parledgroup@series	131
COMMAND\parledgroup@type	131
COMMAND\parledgroupnotespacing	136
COMMAND\parledgrouptrue	22
COMMAND\patchcmd	164
COMMAND\pausenumbering	10, 35, 36, 166
COMMAND\pend	4, 7, 8, 10, 12, 20–22, 56, 60, 64–66, 95, 108, 161, 162, 164, 165
COMMAND\pendL	161
COMMAND\pendR	161
COMMAND\pends	20
COMMAND\perpage	15
COMMAND\prev@nopbR	131
COMMAND\prev@pbR	131
COMMAND\prevpgstyle	28
COMMAND\print@Xnotes	82
COMMAND\print@Xnotes@forpages	82, 84, 163
COMMAND\print@columnseparator	101, 161
COMMAND\print@eledsectionL	69
COMMAND\print@leftcolumn	105
COMMAND\print@line	69
COMMAND\print@lineL	68
COMMAND\print@notesX@forpages	163
COMMAND\print@rightcolumn	105
COMMAND\printlinenumR	137
COMMAND\printlinesR	165
COMMAND\pstart	4, 7, 8, 10, 12, 17, 20–22, 37, 52, 56, 60, 61, 65, 66, 95, 99, 108, 160, 161, 163–165, 167
COMMAND\pstartL	66, 161
COMMAND\pstartR	66, 160, 161
COMMAND\pstartinfootnote	163
COMMAND\raw@text	95
COMMAND\read@linelist	43, 44, 140, 164
COMMAND\ref@reg	47
COMMAND\ref@regR	47, 164
COMMAND\relax	137
COMMAND\reledmac	164
COMMAND\renewcommand	136
COMMAND\resumenumbering	10, 35, 36, 162, 166
COMMAND\resumenumberingR	37, 162
COMMAND\rightlinenumR	42, 159
COMMAND\sameword	86
COMMAND\section	159

COMMAND\section@num	32
COMMAND\selectlanguage	20, 93, 94
COMMAND\set@continuousnumberingforR	37
COMMAND\set@line	52, 164
COMMAND\set@line@action	46, 164
COMMAND\setRlineflag	19, 136
COMMAND\setgoalfraction	13, 136
COMMAND\sethangingsymbol	21
COMMAND\setline	46, 52, 164
COMMAND\setlinenum	46, 52, 164
COMMAND\setmsdataposition	167
COMMAND\setnoteposition...	107
COMMAND\setparledgroupnotespacing	136, 165
COMMAND\setposition...	107
COMMAND\setstanzaindents	9, 21, 136
COMMAND\setwidthliketwocolumns	107
COMMAND\sidenotemargin	22, 161
COMMAND\sidenotemargin*	22, 161
COMMAND\sidenotemarginR	22, 167
COMMAND\skipnumbering	18, 163
COMMAND\sloppy	9
COMMAND\stanza	8, 9, 18, 21, 58, 91, 137, 160
COMMAND\stanzanumtrue	22
COMMAND\startlock	52, 164
COMMAND\startsub	52, 164
COMMAND\sub@action	46, 164
COMMAND\sub@off	88
COMMAND\sub@on	88
COMMAND\subline@numR	43
COMMAND\sublinenumberstyle	18, 137
COMMAND\sublinenumberstyle*	137
COMMAND\sublinenumberstyleR	18, 137
COMMAND\sublinenumincrement	161, 164, 165
COMMAND\sublinenumrepR	41, 159
COMMAND\syntaxonly	165
COMMAND\sza@0@	21
COMMAND\textheight	14
COMMAND\textwidth	58
COMMAND\the@labelX	165
COMMAND\theledlanguageL	94
COMMAND\theledlanguageR	94
COMMAND\thepage	23, 127
COMMAND\thepstartL	18, 160
COMMAND\thepstartR	18, 160
COMMAND\thestanzaL	22
COMMAND\thestanzaR	22
COMMAND\ vbox	62
COMMAND\ vskip	132
COMMAND\ vsplit	80
COMMAND\widthliketwocolumns	10

COMMAND\widthliketwocolumnsfalse	10
COMMAND\widthliketwocolumnstrue	10
COMMAND\xflagref	165
COMMAND\xright@appenditem	54
COMMAND\xspace	25
COMMAND\xxxfootstart	107
COMMAND\xxxmatter	128
ENVIRONMENTLeftside	11, 58, 105
ENVIRONMENTRightside	11, 59, 105
ENVIRONMENTastanza	21, 91, 92, 137, 165, 168
ENVIRONMENTcolumns	35, 165
ENVIRONMENTledgroup	6, 166
ENVIRONMENTleft	18
ENVIRONMENTpages	28, 35, 57, 138, 165, 167
ENVIRONMENTpairs	57, 165
PACKAGEEDMAC	138
PACKAGEEDSTANZA	138
PACKAGEEledmac	163
PACKAGEEledpar	163
PACKAGETABMAC	138
PACKAGEbabel	20, 93, 94, 165
PACKAGEedmac	138
PACKAGEeledmac	5, 95, 135, 136, 138, 161, 164
PACKAGEeledpar	5, 6, 14, 40, 136, 137, 161, 162
PACKAGEetoolbox	32, 101, 130
PACKAGEledmac	6
PACKAGEledpar	1, 6
PACKAGEMemoir	138
PACKAGEMusixtex	162
PACKAGEperpage	2, 15, 137, 165
PACKAGEpolyglossia	20, 93–95, 166
PACKAGEreledmac	1, 3, 5–8, 10, 14, 18, 20–23, 25, 27, 28, 33, 34, 38, 40, 42–47, 49–56, 68, 78, 87, 88, 90, 111, 116, 136, 137, 164–167
PACKAGEreledpar	1, 3, 5–13, 21–27, 37, 42, 43, 49–53, 55, 56, 87, 102, 136–138, 164
PACKAGEsetspace	3, 23
PACKAGESyntonly	165
PACKAGExkeyval	25
PACKAGEX	166

A

\absline@numR	1
\actionlines@listR	1
\actions@listR	1
\add@apparatusL	1
\add@apparatusR	1
\add@inserts@nextR	1
\add@insertsR	1
\add@penaltiesL	1
\add@penaltiesR	1
\advanceline	1

<code>\affixline@numR</code>	1
<code>\affixpstart@numL</code>	1
<code>\affixpstart@numR</code>	1
<code>\affixside@noteR</code>	1
<code>\aftercolumnseparator</code>	1, 9
<code>\araw@textfalse</code>	1
<code>\araw@texttrue</code>	1
<code>astanza (environment)</code>	21
<code>\AtBeginPairs</code>	1, 8
<code>\AtEveryPstartCall</code>	1
<code>\autopar</code>	20

B

<code>\bbl@set@language</code>	1
<code>\beforecolumnseparator</code>	1, 9
<code>\beginnumbering</code>	16
<code>\beginnumberingR</code>	1

C

<code>\c@firstlinenumR</code>	1
<code>\c@firstsublinenumR</code>	1
<code>\c@linenumincrementR</code>	1
<code>\c@sublinenumincrementR</code>	1
<code>\ch@ck@l@ckR</code>	1
<code>\ch@cksub@l@ckR</code>	1
<code>\chapter</code>	1
<code>\chapterinpages</code>	1
<code>\check@goal</code>	1
<code>\check@pstarts</code>	1
<code>\checkpageL</code>	1
<code>\checkpageR</code>	1
<code>\checkpb@columns</code>	1
<code>\checkpbL</code>	1
<code>\checkpbR</code>	1
<code>\checkraw@text</code>	1
<code>\checkverseL</code>	1
<code>\checkverseR</code>	1
<code>\clearl@dleftpage</code>	1
<code>\clearl@drighthpage</code>	1
<code>\cleartoevenpage</code>	1
<code>\cleartol@devenpage</code>	1
<code>\columnrulewidth</code>	1, 9
<code>\Columns</code>	1, 8
<code>\columns@position</code>	1
<code>\Columns@print@after@pend</code>	1
<code>\Columns@print@before@pstart</code>	1
<code>\columnseparator</code>	1, 9
<code>\columnsposition</code>	1, 9
<code>\correct@footinsX@box</code>	1
<code>\correct@Xfootins@box</code>	1

\countLline	1
\countRline	1
\critext	1

D

\do@actions@fixedcodeR	1
\do@actions@nextR	1
\do@actionsR	1
\do@ballastR	1
\do@insidelineLhook	1
\do@insidelineRhook	1
\do@lineL	1
\do@lineLhook	1
\do@lineR	1
\do@lineRhook	1
\do@lockoff	1
\do@lockoffR	1
\do@lockon	1
\do@lockonR	1
\doinsidelineLhook	1, 21
\doinsidelineRhook	1, 21
\dolineLhook	1, 21
\dolineRhook	1, 21
\dump@pstartL@pc	1
\dump@pstartR@pc	1

E

\edlabel	1
\edtext	1
\edtext@later	1
\edtext@now	1
\edtextlater	1, 16
\edtextnow	1, 16
\eled@sectioningR@out	1
\eledsection@correcting@skip	1
\eledsectmark	1, 23
\eledsectnotoc	1, 23
\endlock	1
\endnumbering	1, 16
\endnumberingR	1
\endsub	1
environments:	
astanza	21
Leftside	16
pages	11
pairs	8
Rightside	16

F

\f@x@l@cksR	1
-------------	---

<code>\finish@Pages@notes</code>	1
<code>\first@linenum@out@Rfalse</code>	1
<code>\first@linenum@out@Rtrue</code>	1
<code>\firstlinenum</code>	1, 18
<code>\firstlinenum*</code>	1, 18
<code>\firstlinenumR</code>	1, 18
<code>\firstsublinenum</code>	1, 18
<code>\firstsublinenum*</code>	1, 18
<code>\firstsublinenumR</code>	1, 18
<code>\fix@page</code>	1
<code>\flag@end</code>	1
<code>\flag@start</code>	1
<code>\flush@notesR</code>	1
<code>\footnote@reading</code>	1
<code>\footnote@typeset</code>	1
<code>\footnoteXmk</code>	15
<code>\footnoteXnomk</code>	15

G

<code>\get@familiarfootnote@number</code>	1
<code>\get@intercolumns@width</code>	1
<code>\get@nextboxL</code>	1
<code>\get@nextboxR</code>	1
<code>\get@sidenote@morespace@columns</code>	1
<code>\getline@numR</code>	1
<code>\getlinesfrompagelistL</code>	1
<code>\getlinesfrompagelistR</code>	1
<code>\getlinesfromparlistL</code>	1
<code>\getlinesfromparlistR</code>	1
<code>\goalfraction</code>	1

H

<code>\hidenumbering</code>	18
-----------------------------------	----

I

<code>\if@getnextbox</code>	1
<code>\if@pstarts</code>	1
<code>\if@sidenotesmarginpage</code>	1
<code>\ifaraw@text</code>	1
<code>\ifcsboxvoid</code>	1
<code>\iffirst@linenum@out@R</code>	1
<code>\ifinstanzaL</code>	1
<code>\ifinstanzaR</code>	1
<code>\ifl@dpagfull</code>	1
<code>\ifl@dpaging</code>	1
<code>\ifl@dpairing</code>	1
<code>\ifl@dsamepage</code>	1
<code>\ifl@dusedbabel</code>	1
<code>\ifledRcol</code>	1
<code>\iflinenumberLevenifblank</code>	1

<code>\iflinenumberRevenifblank</code>	1
<code>\ifmovecolumnsonrightpage</code>	1
<code>\ifnomaxlines</code>	1
<code>\ifnosyncpstarts</code>	1
<code>\ifPages@mainmatter</code>	1
<code>\ifprevpgnotnumbered</code>	1
<code>\ifprint@last@after@pendL</code>	1
<code>\ifprint@last@after@pendR</code>	1
<code>\ifpst@rtedL</code>	1
<code>\ifpst@rtedR</code>	1
<code>\ifpstartnumR</code>	1
<code>\ifresumenumberingR@start</code>	1
<code>\ifsameparallelpagenum</code>	1
<code>\ifshiftedpstarts</code>	1
<code>\ifsublines@R</code>	1
<code>\ifwidthliketwocolumns</code>	1
<code>\ifwrittenlinesL</code>	1
<code>\init@series@par</code>	1
<code>\initnumbering@sectcountR</code>	1
<code>\insert@countR</code>	1
<code>\insert@noterule@ledgroup</code>	1
<code>\insert@notes@for@onlyside</code>	1
<code>\inserthangingsymbolL</code>	1
<code>\inserthangingsymbolR</code>	1
<code>\insertlines@listR</code>	1
<code>\inserts@listR</code>	1
<code>\intercolumns@width</code>	1

L

<code>\l@d@set</code>	1
<code>\l@dc@maxchunks</code>	1
<code>\l@dcalc@maxoftwo</code>	1
<code>\l@dcalc@minoftwo</code>	1
<code>\l@dcalcnun</code>	1
<code>\l@dchecklang</code>	1
<code>\l@dleftbox</code>	1
<code>\l@dlinenumR</code>	1
<code>\l@dmake@labelsR</code>	1
<code>\l@dminpagelines</code>	1
<code>\l@dnumpstartsL</code>	1
<code>\l@dnumpstartsR</code>	1
<code>\l@dpagefullfalse</code>	1
<code>\l@dpagefulltrue</code>	1
<code>\l@drightbox</code>	1
<code>\l@dsamepagefalse</code>	1
<code>\l@dsamepagetrue</code>	1
<code>\l@dsetupmaxlinecounts</code>	1
<code>\l@dsetuprawboxes</code>	1
<code>\l@dskipversenumberR</code>	1
<code>\l@dusedbabelfalse</code>	1

<code>\l@dusedbabeltrue</code>	1
<code>\l@duselanguage</code>	1
<code>\l@dzeromaxlinecounts</code>	1
<code>\l@pscl</code>	1
<code>\l@pscr</code>	1
<code>\labelref@listR</code>	1
<code>\last@page@numR</code>	1
<code>\Lcolwidth</code>	1, 9, 11
<code>\led@err@BadLeftRightPstarts</code>	1
<code>\led@err@Columns@InsideEnv</code>	1
<code>\led@err@Columns@WithoutEnv</code>	1
<code>\led@err@Columns@WithoutLeftside</code>	1
<code>\led@err@Columns@WithoutRightside</code>	1
<code>\led@err@LeftOnRightPage</code>	1
<code>\led@err@Leftside@PreviousNotPrinted</code>	1
<code>\led@err@Pages@InsideEnv</code>	1
<code>\led@err@Pages@WithoutEnv</code>	1
<code>\led@err@Pages@WithoutLeftside</code>	1
<code>\led@err@Pages@WithoutRightside</code>	1
<code>\led@err@RightOnLeftPage</code>	1
<code>\led@err@Rightside@PreviousNotPrinted</code>	1
<code>\led@err@TooManyPstarts</code>	1
<code>\led@error@edtext@later@now</code>	1
<code>\led@error@fail@patch@mempnum</code>	1
<code>\led@error@fail@patch@outputpage</code>	1
<code>\led@error@fail@patch@pagenumbering</code>	1
<code>\led@error@fail@patch@thepage</code>	1
<code>\led@error@missing@numbering</code>	1
<code>\led@error@note@called@onleftside</code>	1
<code>\led@error@note@called@onrightside</code>	1
<code>\led@nopbnumR</code>	1
<code>\led@nopbR</code>	1
<code>\led@pbnunR</code>	1
<code>\led@pbR</code>	1
<code>\led@warn@ChangeSyncOption</code>	1
<code>\led@warn@setting@in@rightside</code>	1
<code>\lednopbnum</code>	1
<code>\lednopbnumR</code>	1
<code>\ledpbnunR</code>	1
<code>\ledpbR</code>	1
<code>\ledstrutL</code>	1
<code>\ledstrutR</code>	1
<code>\ledthegoal</code>	1
<code>\leftlinenumR</code>	1
<code>\leftpstartnumL</code>	1
<code>\leftpstartnumR</code>	1
<code>Leftside (environment)</code>	16
<code>\Leftsidehook</code>	1
<code>\Leftsidehookend</code>	1
<code>\line@list@stuffR</code>	1

<code>\line@listR</code>	<u>1</u>
<code>\line@marginR</code>	<u>1</u>
<code>\line@numR</code>	<u>1</u>
<code>\lineation*</code>	<u>1</u> , 18
<code>\lineationR</code>	<u>1</u> , 18
<code>\linenum@outR</code>	<u>1</u>
<code>\linenumberLevenifblanktrue</code>	19
<code>\linenumberRevenifblanktrue</code>	19
<code>\linenumberstyle*</code>	<u>1</u> , 18
<code>\linenumberstyleR</code>	<u>1</u> , 18
<code>\linenumincrement</code>	<u>1</u> , 18
<code>\linenumincrement*</code>	<u>1</u> , 18
<code>\linenumincrementR</code>	<u>1</u> , 18
<code>\linenummargin</code>	<u>1</u>
<code>\linenummargin*</code>	<u>1</u> , 18
<code>\linenummarginColumns</code>	<u>1</u> , 18
<code>\linenummarginColumns*</code>	<u>1</u> , 18
<code>\linenummarginColumnsR</code>	<u>1</u> , 18
<code>\linenummarginR</code>	<u>1</u> , 18
<code>\linenumOnlyPagesForColumns</code>	<u>1</u> , 19
<code>\linenumOnlyPagesForColumnsR</code>	<u>1</u> , 19
<code>\linenumrepR</code>	<u>1</u>
<code>\linesinpar@listL</code>	<u>1</u>
<code>\linesinpar@listR</code>	<u>1</u>
<code>\list@clearing@regR</code>	<u>1</u>
<code>\list@pstartL@pc</code>	<u>1</u>
<code>\list@pstartR@pc</code>	<u>1</u>
<code>\lock@off</code>	<u>1</u>

M

<code>\maxchunks</code>	<u>1</u> , 8
<code>\maxlinesinpar@list</code>	<u>1</u>
<code>\memorydump</code>	17
<code>\memorydumpL</code>	<u>1</u>
<code>\memorydumpR</code>	<u>1</u>

N

<code>\n@num</code>	<u>1</u>
<code>\namebox</code>	<u>1</u>
<code>\new@lineL</code>	<u>1</u>
<code>\new@lineR</code>	<u>1</u>
<code>\newnamebox</code>	<u>1</u>
<code>\newnamecount</code>	<u>1</u>
<code>\newseries@par</code>	<u>1</u>
<code>\next@line@list@stuffR</code>	<u>1</u>
<code>\notesXwidthliketwocolumns</code>	10
<code>\num@linesR</code>	<u>1</u>
<code>\numberpstartfalse</code>	17
<code>\numberpstarttrue</code>	17
<code>\numpagelinesL</code>	<u>1</u>

\numpagelinesR 1

O

\one@lineR 1
 \onlysideX 15
 optionadvancedshiftedpstarts 12, 13
 optionauxdir 166
 optioncontinuousnumberingwithcolumns 10, 166–168
 optionmovecolumnsonrightpage 11, 167
 optionnomaxlines 12, 13, 26
 optionnopbinverse 167
 optionnosyncpstarts 13, 26, 123
 optionprevpgnotnumbered 167
 optionprevpgstyle 167
 optionshiftedpstarts 6, 13, 26
 optionsidenotesmarginpage 22, 27, 168
 optionwidthliketwocolumns 10, 168

P

\page@action 1
 \page@numR 1
 \Pages 1, 11
 pages (environment) 11
 \Pages@mainmatter 1
 pairs (environment) 8
 \par@lineR 1
 \par@patch@pagenumbering 1
 \par@patch@thepage 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
 \parledgrouptype@ 1
 \pausenumbering@page@numR 1
 \pausenumberingR 1
 \pend 20
 \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	20
\pstartL	1
\pstartR	1

R

\Rcolwidth	1, 9, 11
\read@linelist	1
\reledpar@error	1
\reledpar@warning	1
\restore@pstartL@pc	1
\restore@pstartR@pc	1
\resumenumberingR	1
\rightlinenumR	1
\rightpstartnumL	1
\rightpstartnumR	1
Rightside (environment)	16
\Rightsidehook	1
\Rightsidehookend	1
\Rlineflag	1

S

\save@familiarfootnote@number	1
\save@section@number	1
\section@numR	1
\selectlanguage	1
\set@continuousnumberingforR	1
\set@line	1
\set@line@action	1
\set@sectcountR	1
\setgoalfraction	1, 13
\sethangingsymbol	21
\setline	1
\setlinenum	1
\setnamebox	1
\setnotepositionliketwocolumns@C	1
\setnotepositionliketwocolumns@L	1
\setnotepositionliketwocolumns@R	1
\setpositionliketwocolumns@C	1
\setpositionliketwocolumns@L	1
\setpositionliketwocolumns@R	1
\setRlineflag	19
\setwidthliketwocolumns	1

<code>\sidenote@marginR</code>	1
<code>\sidenotemargin*</code>	1
<code>\sidenotemarginR</code>	1
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 18
<code>\startlock</code>	1
<code>\startsub</code>	1
<code>\sub@action</code>	1
<code>\subline@numR</code>	1
<code>\sublinenumberstyle*</code>	1, 18
<code>\sublinenumberstyleR</code>	1, 18
<code>\sublinenumincrement</code>	1, 18
<code>\sublinenumincrement*</code>	1, 18
<code>\sublinenumincrementR</code>	1, 18
<code>\sublinenumrepR</code>	1

T

<code>\theledlanguageL</code>	1
<code>\theledlanguageR</code>	1
<code>\thepar@page</code>	1
<code>\thepstartL</code>	18
<code>\thepstartR</code>	18
<code>\thestanzaL</code>	1, 22
<code>\thestanzaR</code>	1, 22

U

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

W

<code>\widthliketwocolumns</code>	10
---	----

X

<code>\Xendlineflag</code>	16
<code>\Xlineflag</code>	16
<code>\Xnoteswidthliketwocolumns</code>	10
<code>\Xonlyside</code>	15

Change History

v.2.15.0.	
General: Fix a bug with <code>\msdata</code> when using multiple <code>\beginnumbering... \endnumbering</code>	1
v0.1.0.	
General: First public release	1
v0.2.0.	
<code>\Columns</code> : Added <code>\l@dchecklang</code> and <code>\l@duselanguage</code> to <code>\Columns</code>	99
<code>\Pages</code> : Added <code>\l@duselanguage</code> to <code>\Pages</code>	112
General: Added section of babel related code	93
Fix babel problems	1
v0.3.0.	
<code>\Pages</code> : Added <code>\ledstrutL</code> to <code>\Pages</code>	112
Added <code>\ledstrutR</code> to <code>\Pages</code>	114
<code>\Rightsidehookend</code> : Added <code>\Leftsidehook</code> , <code>\Leftsidehookend</code> , <code>\Rightsidehook</code> and <code>\Rightsidehookend</code>	59
<code>\affixline@numR</code> : Changed <code>\affixline@numR</code> to match new <code>eledmac</code>	75
<code>\do@actions@nextR</code> : Used <code>\do@actions@fixedcode</code> in <code>\do@actionsR</code>	73
<code>\do@lineL</code> : Added <code>\do@lineLhook</code> to <code>\do@lineL</code>	67
Simplified <code>\do@lineL</code> by using macros for some common code	67
<code>\do@lineR</code> : Changed <code>\do@lineR</code> similarly to <code>\do@lineL</code>	70
<code>\flag@end</code> : Removed extraneous spaces from <code>\flag@end</code>	52
<code>\ifledRcol</code> : Moved <code>\ifl@dpairing</code> to <code>eledmac</code>	28
<code>\ifpst@rtedR</code> : Moved <code>\ifpst@rtedL</code> to <code>eledmac</code>	33
<code>\l@dlinenumR</code> : Simplified <code>\leftlinenumR</code> and <code>\rightlinenumR</code> by introducing <code>\l@dlinenumR</code>	42
<code>\l@dnumpstartsR</code> : Moved <code>\l@dnumpstartsL</code> to <code>eledmac</code>	95
<code>\ledstrutR</code> : Added <code>\ledstrutL</code> and <code>\ledstrutR</code>	117
<code>\sublinenumrepR</code> : Added <code>\linenumrepR</code> and <code>\sublinenumrepR</code>	41
General: Added <code>\do@lineLhook</code> and <code>\do@lineRhook</code>	70
Added hooks into Leftside environment	58
Reorganize for <code>ledarab</code>	1
v0.3.a.	
<code>\line@marginR</code> : Do not just set <code>\line@marginR</code> in <code>\linenummargin</code>	38
General: Minor <code>\linenummargin</code> fix	1
v0.3.b.	
<code>\Pages</code> : Added <code>\l@dminpagelines</code> calculation for succeeding page pairs	115
General: Improved parallel page balancing	1
v0.3.c.	
General: Compatibilty with Polyglossia	1
v0.4.0.	
General: No more <code>ledparpatch</code> . All patches are now in the main file.	1
v0.5.0.	
General: Corrections about <code>\section</code> and other titles in numbered sections	1
v0.6.0.	
General: Be able to use <code>\chapter</code> in parallel pages.	1
v0.7.0.	
General: Option ‘ <code>shiftedverses</code> ’ which make there is no blank between two parallel verses with inequal length.	1

v0.8.0.	
General: Possibility to have a symbol on each hanging of verses, like in the french typography. Redefine the commande <code>\hangingsymbol</code> to define the character. . . .	1
v0.9.0.	
<code>\ifledRcol</code> : Moved <code>\iflledRcol</code> and <code>\ifnumberingR</code> to <code>eledmac</code>	28
General: Possibility to number <code>\pstart</code>	17
Possibility to number the <code>pstart</code> with the commands <code>\numberpstarttrue</code>	1
v0.9.1.	
General: The numbering of the <code>pstarts</code> restarts on each <code>\beginnumbering</code>	1
v0.9.2.	
General: Debug : with <code>\Columns</code> , the hanging indentation now runs on the left columns and the hanging symbol is shown only when <code>\stanza</code> is used.	1
v0.9.3.	
General: <code>\thepstartL</code> and <code>\thepstartR</code> use now <code>\bfseries</code> and not <code>\bf</code> , which is deprecated and makes conflicts with <code>memoir</code> class.	1
v0.10.0.	
General: <code>\edlabel</code> commands on the right side are now correctly indicated.	1
<code>\edlabel</code> commands which start a paragraph are now put in the right place.	1
v0.11.0.	
<code>\Columns</code> : Line numbering by <code>pstart</code>	100
<code>\affixline@numR</code> : Changed <code>\affixline@numR</code> to allow to disable line numbering (like in <code>eledmac</code> 0.15).	75
<code>\get@nextboxR</code> : Change <code>\get@nextboxL</code> and <code>\get@nextboxR</code> to allow to disable line numbering (like in <code>eledmac</code> 0.15).	123
<code>Pstart</code> number can be printed in side	125
<code>\inserthangingsymbolR</code> : Prevent the column separator for hanging verse from shifting	90
General: Change <code>\do@lineL</code> and <code>\do@lineR</code> to allow line numbering by <code>pstart</code> (like in <code>eledmac</code> 0.15).	67
Lineation can be by <code>pstart</code> (like in <code>eledmac</code> 0.15).	37
New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . .	90
v0.12.0.	
General: New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions.	90
v1.0.0.	
General: Compatibility with <code>eledmac</code> . Change name to <code>eledpar</code>	1
Debug in lineation by <code>pstart</code>	37
v1.0.1.	
General: Correction on <code>\numberonlyfirstinline</code> with lineation by <code>pstart</code> or by page. .	1
v1.1.0.	
<code>\pstartR</code> : Add <code>\labelpstarttrue</code> (from <code>eledmac</code>).	60
General: <code>Shiftedverses</code> becomes <code>shiftedpstarts</code>	1
v1.1.1.	
<code>\pstartR</code> : Correct <code>\pstartR</code> bug introduced by 1.1.	60
v1.1.2.	
<code>\affixside@noteR</code> : Remove spurious space between line number and line content . .	89
v1.2.0.	
General: Support for <code>\led<section></code> commands in parallel texts.	1
v1.2.1.	
<code>\set@sectcountR</code> : For the right section, the counter is defined only once.	35

v1.3.0.	
\edtext: Manage RTL language.	52
v1.3.2.	
General: Debug with some classes.	1
v1.3.3.	
General: Debugging the left notes of the right column.	89
v1.3.4.	
General: Allow use of commands in sidenotes, as introduced by eledmac 1.0.	89
v1.4.0.	
General: Added \do@insidelineLhook and \do@insidelineRhook	70
v1.4.1.	
General: Enable the use of stanzaindentsrepetition within astanza environment.	91
v1.4.3.	
\inserthangingsymbolR: Hanging verse is no longer automatically flush right.	90
\pendL: Spurious spaces in \pendL.	64
\pendR: Spurious spaces in \pendR.	65
\pstartR: Spurious spaces in \pstartL and \pstartR.	60
General: Corrects a false hanging verse when a verse is exactly the length of a line.	1
v1.5.0.	
\sublinenumincrement*: Add starred version of \firstlinenum,	
\linenumincrement, \firstsublinenum, \sublinenumincrement to change	
both Left and Rightside.	40
General: Add, as in eledmac, features to manage page breaks.	1
v1.6.0.	
General: Add tool and documentation for parallel ledgroups	22
v1.7.0.	
General: Add, as in eledmac, features to make crossrefs with pstart numbers.	1
v1.8.0.	
\Columns: Modify \Columns to enable to add section's title.	98
Suppress \l@dchecklang from \Columns.	99
\Pages: Modify \Pages to enable to add section's title.	110
\l@dchecklang: Suppress \l@dchecklang which did not work and was not logical,	
because both columns could have the same language but not the main language of	
the document.	93
\pendL: As in eledmac, \pendL can have an optional argument.	64
\pendR: As in eledmac, \pendR can have an optional argument.	65
\print@columnseparator: Move some code of \Columns to	
\print@columnseparator.	101
\pstartR: As in eledmac, \pendL and \pendR can have an optional argument.	60
\sidenotemarginR: \sidenotemargin is now directly defined in eledmac to be able	
to manage eledpar.	88
Add \sidenotemargin*	88
\theledlanguageR: Correct left/right language setting with polyglossia.	95
General: \beginnumbering is defined only on eledmac, not on eledpar.	33
\l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in eledmac.	89
Add \beforecolumnseparator and \aftercolumnseparator.	9
Add \columnspostion.	9
Add, as in eledmac, new system of sectioning commands.	1
Add, as in eledmac, option to insert something after \pends / verses.	1
Add, as in eledmac, option to insert something between \pstarts / verse.	1

Change <code>\do@lineR</code> and <code>\do@lineR</code> to allow new sectioning commands.	67
Compatibility with <code>musixtex</code>	1
Debug <code>eledmac</code> sectioning command after using <code>\resumenumbering</code>	1
New sectioning commands, as in <code>eledmac</code>	23
Suppress <code>\ifl@dsamelang</code> which did not work and was not logical, because both columns could have the same language but not the main language of the document.	93
v1.8.1.	
<code>\do@lineL</code> : Fix a bug with critical notes at the beginning of a page, (maybe added by v1.8.0) (?).	67
<code>\do@lineR</code> : Fix a bug with critical notes at the beginning of a page, added by v1.8.0 (?).	70
v1.8.2.	
<code>\flag@end</code> : <code>\flag@start</code> and <code>\flag@end</code> are now defined only one time for <code>eledmac</code> and <code>eledpar</code>	52
<code>\lineation*</code> : Add <code>\lineation*</code>	38
<code>\reledpar@error</code> : Errors specific to <code>eledpar</code> send to <code>eledpar</code> handbook	28
General: Debug <code>\eledxxx</code> with some paper sizes	1
Debug left and side note (bugs added by 1.8.0)	1
v1.8.3.	
<code>\Pages</code> : Debug blank pages when using optional argument in the last <code>\pend</code>	110
<code>\doinsidelineRhook</code> : Added <code>\dolineLhook</code> , <code>\dolineRhook</code> , <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code>	70
<code>\pausenumbering@page@numR</code> : Debug <code>\resumenumberingR</code>	35
General: Add <code>\noeledxxx</code> , as in <code>eledmac</code>	1
v1.9.0.	
<code>\ifwidthliketwocolumns</code> : Added <code>widthliketwocolumns</code> option	27
<code>\theledlanguageR</code> : Debug left/right language switching with <code>polyglossia</code> . Do not write in <code>.aux</code> file when setting left/right lines.	95
General: Add <code>\AtBeginPairs</code> macro.	8
Compatibility with <code>\Xnoteswidthliketwocolumns</code> and <code>\notesXwidthliketwocolumns</code>	1
v1.9.1.	
<code>\ifledRcol</code> : Moved <code>\ifl@dpaging</code> to <code>eledmac</code>	28
v1.10.0.	
<code>\Pages</code> : Debug wrong pages splitting when no optional argument is used in last <code>\pend</code> (bug was added in v1.8.3).	110
Debug wrong parallel pages synchronization when an <code>\edtext</code> falls across two pages.	110
General: Compatibility with <code>\AtEveryPstart</code> and <code>\AtEveryPend</code>	1
Restore critical notes in <code>\eledsection</code> in parallel columns (this bug was added in 1.8.2).	1
v1.10.1.	
<code>\next@line@list@stuffR</code> : Revert modification of 1.4.2, which makes bugs with numbering. Leave vertical mode to solve spurious space before <code>minipage</code>	51
v1.11.0.	
<code>\edtext</code> : <code>\critext</code> and <code>\edtext</code> are now defined only in <code>eledmac</code>	52
General: Compatibility of standard footnotes with some <code>biblatex</code> styles.	1
v1.12.0.	
<code>\Columns</code> : Add <code>\l@dprintingcolumnstrue</code>	98
<code>\Pages</code> : Add <code>\l@dprintingpagestrue</code>	110
<code>\edlabel</code> : <code>\edlabel</code> and <code>\edindex</code> works now with <code>hyperref</code> when using <code>eledpar</code>	87

\edlabel is now defined only one time for both eledmac and eledpar	87
\print@eledsectionL: Compatibility with Lua \TeX RTL languages.	69
\print@eledsectionR: Compatibility with Lua \TeX RTL languages.	72
\print@lineL: Compatibility with Lua \TeX RTL languages.	69
General: Compatibility with Lua \TeX RTL languages.	1
v1.12.1.	
\print@eledsectionL: Fixes bug with Lua \TeX RTL \eledsection.	69
v1.13.0.	
\Pages: Prevent false overfull hboxes when using \Pages outside of pages environment.	110
When using shiftedpstarts option, a \l@leftbox with a null height will advance the \pagetotal in any case.	110
\clearl@drighthouse: Use \newpage instead of \clearpage.	118
\ifledRcol: Remove false boolean settings which are not needed.	28
General: Enable the use of optional argument of & in astanza environment.	91
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)	110
\correct@footinsX@box: Call \correct@footinsX@box and \correct@Xfootins@box directly in \print@notesX@forpages and \print@Xnotes@forpages.	82
Correct \correct@footinsX@box and \correct@Xfootins@box	82
v1.14.0.	
General: Fix a bug with line number position when using \eledsection and similar commands for RTL texts with Lua \TeX .	1
The \newifs are not followed by boolean values set to false, because it is the \TeX default setting.	1
v1.15.0.	
\do@actions@nextR: Add actions 1008 and 1009	73
\inserthangingsymbolR: Prevent more efficiently the column separator from shifting when a verse is hanging	90
\lineationR: As \lineation, \lineationR automatically set the \pstartinfootnote.	37
\n@num: \n@num defined only one time for both Eledmac and Eledpar.	47
\skipnumbering: \skipnumbering defined only one time for both Eledmac and Eledpar	52
General: Add \AtEveryPstartCall.	1
Add sameparallelpagenumber option.	13
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.	53
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 <code>nofamiliar / nocritical</code> option of <code>eledmac</code> .	1
New package option <code>sameparallelpagenum</code> to have the same page number for both left and right side.	1
v1.16.1.	
General: Write information about line-list file version in the correct file.	1
v1.16.2.	
General: Fix a bug when adding empty lines before a <code>\pend</code> in combination with some specific penalties setting.	1
v1.17.0.	
General: Add compatibility of optional argument of <code>\pstart/\pend</code> and <code>\AtEveryPstart/\AtEveryPend</code> with two columns mode.	1
v1.21.0.	
General: Add <code>\hidenum</code>	18
v2.0.0.	
<code>\@adv</code> : <code>\@adv</code> defined only in <code>reledmac</code> .	46
<code>\@lab</code> : <code>\@lab</code> defined only in <code>eledmac</code> .	88
<code>\@ref@regR</code> : <code>\@ref</code> defined only in <code>reledmac</code> , code specific to right side moved in <code>\ref@regR</code> .	47
<code>\@set</code> : <code>\@set</code> defined only in <code>reledmac</code> .	46
<code>\advanceline</code> : <code>\advanceline</code> defined only in <code>reledmac</code> .	52
<code>\bbl@set@language</code> : Patch <code>\bbl@set@language</code> instead of redefining it	93
<code>\do@lockonR</code> : <code>\do@lockon</code> defined only in <code>reledmac</code> .	46
<code>\endlock</code> : <code>\startlock</code> and <code>\endlock</code> defined only in <code>reledmac</code> .	52
<code>\endsub</code> : <code>\startsub</code> and <code>\endsub</code> defined only in <code>reledmac</code> .	52
<code>\fix@page</code> : <code>\fix@page</code> is defined only once in <code>reledmac</code>	45
<code>\l@d@set</code> : <code>\l@d@set</code> defined only in <code>reledmac</code> .	46
<code>\line@marginR</code> : <code>\linenummargin</code> now defined only once time in <code>reledmac</code> .	38
<code>\page@action</code> : <code>\page@action</code> defined only in <code>reledmac</code> .	46
<code>\read@linelist</code> : <code>\read@linelist</code> is defined only once time in <code>\reledmac</code> .	44
<code>\set@line</code> : <code>\set@line</code> defined only in <code>reledmac</code> .	52
<code>\set@line@action</code> : <code>\set@line@action</code> defined only in <code>reledmac</code> .	46
<code>\setline</code> : <code>\setline</code> defined only in <code>reledmac</code> .	52
<code>\setlinenum</code> : <code>\setlinenum</code> defined only in <code>reledmac</code> .	52
<code>\skip@lockoff</code> : <code>\do@lockoff</code> defined only in <code>reledmac</code> .	47
<code>\sub@action</code> : <code>\sub@action</code> defined only in <code>reledmac</code> .	46
<code>\sublinenumincrement*</code> : <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code> are now defined only in <code>reledmac</code> .	40
<code>\theledlanguageR</code> : Patch <code>\otherlanguage</code> instead of redefining it.	95
General: <code>\@nl</code> is now defined only in <code>reledmac</code> .	44
<code>\ifbypage@</code> and <code>\ifbypstart@R</code> defined in <code>eledmac</code> .	37
Fix some bugs with ‘ <code>sameparallelpagenum</code> ’ option.	1
Many code refactored and moved to <code>reledmac</code> .	1
Package’s name becomes <code>reledpar</code> .	1
Totally new implementation of ‘ <code>sameparallelpagenum</code> ’ option.	1
<code>chapterinpages</code> : Deleting the old system of managing parallel chapter, keep only the new one with <code>\patchcmd</code> .	58
v2.1.0.	
General: Fix a bug when using <code>\eledsection</code> and related on right pages when page width is short.	1
Fix a bug when using <code>\pagenumbering</code> with memoir (bug added in v2.0.0).	1

Fix a bug with <code>\setparledgroupnotespacing</code> with the <code>shiftedpstarts</code> option.	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code>	1
Options to custom empty right page before <code>\Pages</code>	1
v2.2.0.	
General: <code>astanza</code> environment can take an optional argument, which will be the optional argument of <code>\pstart</code> started by this environment.	1
New tools to number stanza	1
v2.2.1.	
General: Fix a bug with optional argument of last left <code>\pend</code>	1
v2.3.0.	
<code>\Pages</code> : Fix a bug when calling <code>\Columns</code> after a <code>\Pages</code> (bug added in v1.13.0). . . .	110
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.	
<code>\ledstrutR</code> : Deleted <code>\ledstrutL</code> and <code>\ledstrutR</code>	117
Fix a bug with dotted letter	117
General: New way of (not) synchronizing the parallel pages.	1
Option to switch to <code>\mainmatter</code> when calling <code>\Pages</code>	1
v2.5.0.	
General: Disable empty lines as paragraph in <code>astanza</code>	1
Fix a bug introduced in v1.15.0 which made hanging indentation in verse not work anymore.	1
New commands <code>\linenummarginR</code> and <code>\linenummargin*</code>	1
v2.5.1.	
General: Fix spurious space when using optional argument of <code>astanza</code> environment (introduced in v2.5.0).	1
v2.5.2.	
General: Fix a bug introduced in v2.5.0 with <code>\linenummargin</code> , <code>\firstlinenum</code> , <code>\linenumincrement</code> , <code>\firstsublinenum</code> , <code>\sublinenumincrement</code>	1
v2.6.0.	
<code>\l@dmake@labelsR</code> : <code>\@Rlineflag</code> is not stored directly after the line number, but as a fifth argument of <code>\the@labelX</code> . Can be retrieved by <code>\xflagref</code>	87
General: <code>\Xlineflag</code> and <code>\Xendlineflag</code> added	1
<code>\printlinesR</code> deleted	1
Error message when calling <code>\Pages</code> or <code>\Columns</code> without previous <code>pages</code> or <code>pairs</code> environment.	1
Fix 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 <code>pages</code> or <code>columns</code> environments.	1
Fix compatibility with <code>babel</code> (broken in v2.0.0).	1
No error messages about ends of left / right page when using the <code>\syntaxonly</code> command of the <code>syntonly</code> package.	1
v2.6.1.	
General: Fix bug, introduced in v2.6.0, with footnote numbering when using <code>perpage</code> package.	1
v2.6.2.	
<code>\newseries@par</code> : The \TeX counter <code>\footnoteX@reading</code> is defined in <code>reledmac</code> . . .	53

General: Fix (again) bugs with footnote numbering in parallel typesetting while using <code>\edgroup</code> environments (bug added in v2.6.0).	1
Fix a bug (added in v2.6.0) with footnote numbering in parallel typesetting while using <code>\polyglossia</code> with specific numbering systems (like Greek).	1
v2.6.3.	
General: Fix spurious dot when using <code>\linenummargin</code> on right side (introduced in v2.5.0).	1
v2.7.0.	
General: <code>\reledmac</code> cross-referencing can take advantage of <code>xr</code> package.	1
v2.7.1.	
General: Fix a bug added in <code>\reledmac</code> 2.8.2, when typesetting parallel text just after a sectioning command	1
v2.8.0.	
General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> options.	1
Add <code>\linenumberLevenifblanktrue</code> and <code>\linenumberRevenifblank</code> commands	1
Fix a bug when the right line number style is not the same to the left line number style	1
v2.9.0.	
General: Add <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> .	1
More specific error messages.	1
v2.9.1.	
General: Prevent <code>\Xtxtbeforenotes</code> hook from causing notes to go beyond the bottom margin	1
v2.10.0.	
<code>\do@actions@nextR</code> : Add action 1010	73
General: Add new tools to make apparatuses of manuscripts	1
v2.11.0.	
<code>\correct@footinsX@box</code> : Clarification in the handbook about the use of <code>\Xonlyside</code> and <code>onlysideX</code> and error message if misuse.	82
New implementation of <code>\Xonlyside</code> and <code>\onlysideX</code> hooks, prevent trouble with vertical spacing.	82
General: Compatibility with <code>\reledmac</code> 's <code>auxdir</code> option.	1
v2.12.0.	
General: Add <code>\edtextlater</code> and <code>\edtextnow</code> .	1, 85
Fix a bug with <code>\AtBeginPairs</code> .	1
Fix a bug with <code>\lineationR</code> .	1
v2.13.0.	
General: Compatibility with the new features of <code>\reledmac</code> 2.15.0	1
v2.13.1.	
General: Simplification of the code concerning the sectioning command	1
v2.14.0.	
General: Add <code>\linenumberlistR</code> , equivalent to <code>\linenumberlist</code> for the right side.	1
Error message when left / right side are not defined.	1
More explicit error message when the stanza indentation is not defined.	1
New commands to have specific line margins for text running in parallel columns	1
When indexing texts in sidenotes with <code>\edtext</code> , referring to the line number where the sidenote is called.	1

v2.14.1.	
General: Fix a bug when using <code>\markboth</code> on <code>\doinsidelineLhook</code> and <code>\doinsidelineRhook</code>	1
Fix a bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.13.1).	1
v2.14.2.	
General: Fix spurious space with <code>\edtextlater</code>	1
v2.14.3.	
General: Fix a bug in <code>\edtextnow</code>	1
v2.14.4.	
General: Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code>	1
v2.15.0.	
General: Add <code>movecolumnspostiononrightpage</code> option.	1
Fix a bug about space before sectioning commands after a <code>pairs</code> environment (bug added in v. 2.14.4).	1
Fix a bug with page break when using <code>\columns</code> (bug added in v. 2.14.1).	1
New macro <code>\sidenotemarginR</code> , fix default value to right.	1
v2.16.0.	
General: Compatibility with <code>\setmsdataposition</code>	1
v2.16.1.	
General: Restore sidenotes on left side, deleted by mistake 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 <i>latexmk</i>	1
v2.16.3.	
General: Fix a bug when setting <code>\aftercolumnseparator</code>	1
v2.17.0.	
General: Change code for compatibility with <code>reledmac</code> 2.18.0	1
v2.17.1.	
General: Fix a bug with <code>prevpgstyle</code> option when not using <code>prevpgnotnumbered</code>	1
v2.17.2.	
<code>\Pages</code> : Fix a bug when changing <code>\Lcolwidth</code> and <code>\Rcolwidth</code> in <code>pages</code> environment.	110
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 <code>reledmac</code>	1
v2.19.0.	
General: Compatibility with v2.22.0 of <code>reledmac</code>	1
v2.19.1.	
General: Fix a bug with <code>\edtext</code> in sectioning command with <code>shiftedpstarts</code> mode.	1
v2.19.2.	
General: <code>continuousnumberingwithcolumns</code> also works for the <code>pstart</code> number.	1
v2.19.3.	
General: Do not print footnotes at the first run.	1
Fix a bug with <code>nopbinverse</code> when using consecutive <code>\Pages</code>	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 <code>astanza</code> in v. 2.18.0	1
v2.20.0.	
<code>\next@line@list@stuffR</code> : Add <code>\next@line@list@stuffR</code> internal hook.	51
General: Add <code>\linenumOnlyPagesForColumns</code> and	
<code>\linenumOnlyPagesForColumnsR</code> to tell to print the line numbers associated with	
a column only on some specific pages (for example, only on right pages).	1
Add <code>sidenotesmarginpage</code> option to make the sidenotes refer to page margins and	
not column margins.	1
Fix a bug with <code>\continuousnumberingwithcolumns</code> when using multiple series of	
continued numbering text. Now, the line number are correctly reset at each	
<code>\beginnumbering</code> .	1
Fix a bug with <code>\continuousnumberingwithcolumns</code> and inner / outer notes and line	
numbering.	1
Fix a bug with <code>\continuousnumberingwithcolumns</code> when a <code>\beginnumbering</code> is	
called after an other <code>\beginnumbering... \endnumbering</code> .	1
Fix a bug with <code>\widthliketwocolumns</code> when manually setting	
<code>\beforecolumnseparator</code> and <code>\aftercolumnseparator</code> .	1
Fix compatibility between <code>\continuousnumberingwithcolumns</code> and	
<code>\numberpstartfalse</code> .	1
Fix incompatibility between <code>\lineation{page}</code> and	
<code>\continuousnumberingwithcolumns</code> option	1