The pageslts package

H.-Martin Münch
<Martin.Muench at Uni-Bonn.de>

2025-08-14 v2.0g

Abstract

This LaTeX package puts the labels `LastPage` (`\AddToHook{enddocument/afterlastpage}`, formerly `\AtEndDocument`) and `VeryLastPage` (also `\AddToHook{enddocument/afterlastpage}`, but formerly `\AfterLastShipout`) into the `.aux` file, allowing the user to refer to the last page of a document. This might be particularly useful in places like headers or footers. When more than one page numbering scheme is used, these references do not give the total *number* of pages. For this case the label `LastPages` is introduced. Additionally, at the last page of each page numbering scheme a label `pagesLTS.<numbering scheme>` is placed, where `<numbering scheme>` is e. g. arabic, roman, Roman, alph, or Alph. For fnsymbol please use `\lastpageref{pagesLTS.fnsymbol}` instead of `\pageref{pagesLTS.fnsymbol}`. When the same numbering scheme is used twice, the page numbers are either reset to one or continued automatically, depending on the option given when the package is called. The command `\theCurrentPage` prints the current total/absolute page number – in contrast to `\thepage`, which gives only the page *name* in the current page numbering scheme. `\theCurrentPageLocal` gives the current number of pages in the current page numbering scheme. `\thepage` and `\theCurrentPageLocal` are different e. g. when `\addtocounter{page}{...}` or `\setcounter{page}{...}` were used. At the first page of the document a label `pagesLTS.0` is created. This label can be referred to, too. Further labels are provided for special cases.

The alphalph package is supported, i. e. page numbers alph or Alph > 26 and fnsymbol > 9 can be used (with according options set). Even zero and negative page numbers can be used with `arabic`, `alph`, `Alph`, `roman`, `Roman`, and `fnsymbol` page numbering (with alphalph package and according options).

`\pageref*` and `\lastpageref*`, for using hyperref but suppressing links, are supported. – The name of the pageslts package refers to **L**ast, **T**otal, and page numbering **S**chemes pages. pagesLTS was a former name of this package.

Right after `\begin{document}` a `\pagenumbering{...}` should be called – with the appropriate argument out of e. g.
arabic (Arabic numerals: 1, 2, 3, 4,...),
roman (Lowercase Roman numerals: i, ii, iii, iv,...), Roman (Uppercase Roman numerals: I, II, III, IV,...),
alph (Lowercase letters: a, b, c, d,...), Alph (Uppercase letters: A, B, C, D,...),
fnsymbol (Footnote symbols: $*$, †, ‡, §,...),
gobble (Maybe `\thispagestyle{empty}` is more useful?),
silly (from the sillypage package).

This package first started as a revision of the lastpage package of **Jeffrey P. Goldberg** (Thanks!), but then it was deemed necessary to provide a separate, enhanced package. After updates of the kernel and the package, lastpage now does what it is supposed to do.

Disclaimer for web links: The author is not responsible for any contents referred to in this work unless he has full knowledge of illegal contents. If any damage occurs by the use of information presented there, only the author of the respective pages might be liable, not the one who has referred to these pages.

Contents

1 Introduction

This package puts the labels `LastPage` (`\AddToHook{enddocument/afterlastpage}`, formerly `\AtEndDocument`; same as the LastPage package) and `VeryLastPage` (also `\AddToHook{enddocument/afterlastpage}`, but formerly `\AfterLastShipout`) into the `.aux` file, allowing the user to refer to the last page of a document via `\lastpageref{LastPage}` and `\lastpageref{VeryLastPage}`. This might be particularly useful in places like headers or footers. When more than one page numbering scheme is used, these references do not give the total *number* of pages. For this case the label `LastPages` is introduced (similar to the label `TotPages` of the TotPages package, but the label `LastPages` is set later in the document). Additionally, at the last page of each page numbering scheme a label `pagesLTS.`<*numbering scheme*> is placed, where <*numbering scheme*> is e. g. arabic, roman, Roman, alph, or Alph. For fnsymbol please use `\lastpageref{pagesLTS.fnsymbol}` instead of `\pageref{pagesLTS.fnsymbol}`. When the same numbering scheme is used twice, the page numbers are either reset to one or continued automatically, depending on the option given when the package is called. The command `\theCurrentPage` prints the current total/absolute page number – in contrast to `\thepage`, which gives only the page *name* in the current page numbering scheme. `\theCurrentPageLocal` gives the current number of pages in the current page numbering scheme. `\thepage` and `\theCurrentPageLocal` are different e. g. when `\addtocounter{page}{...}` or `\setcounter{page}{...}` were used. (See also LaTeX bug 3421: 3rd page is even (twoside, titlepage, abstract), https://www.latex-project.org/cgi-bin/ltxbugs2html?category=LaTeX&responsible=anyone&state=anything&keyword=pagenumber&pr=latex%2F3421&search=.) At the first page of the document a label `pagesLTS.0` is created. Also to this label it can be referred. Further labels are provided for special cases.

The alphalph package is supported, i. e. page numbers `alph` or `Alph` > 26 and `fnsymbol` > 9 can be used (with the according options set). Even zero or negative page numbers can be used with `arabic`, `alph`, `Alph`, and `fnsymbol` page numbering (with alphalph package and according options), and zero `roman` and `Roman` pages, too.

Right after `\begin{document}` a `\pagenumbering{...}` should be called – with the appropriate argument out of e. g.
arabic (Arabic numerals: 1, 2, 3, 4,...),
roman (Lowercase Roman numerals: i, ii, iii, iv,...), Roman (Uppercase Roman numerals: I, II, III, IV,...),
alph (Lowercase letters: a, b, c, d,...), Alph (Uppercase letters: A, B, C, D,...),
fnsymbol (Footnote symbols: ∗, †, ‡, §,...),
gobble (Maybe `\thispagestyle{empty}` is more useful?),
silly (from the sillypage package).

This package first started as a revision of the lastpage package of Jeffrey P. Goldberg (Thanks!), but then it became obvious that a replacement was needed to accomplish what this package does.

Trademarks appear throughout this documentation without any trademark symbol; they are the property of their respective trademark owner. There is no intention of infringement; the usage is to the benefit of the trademark owner.

`logical page numbers`

Tip: For the display of the `pdf` file use **logical page numbers** together with hyperref!

- In Adobe Acrobat Reader Continuous Release 2024.005.20320 enable:
  Edit > Preferences > Categories: Page Display > Page Content and Information: Use logical page numbers
- Use the hyperref package with option `plainpages=false`.

The display will be e. g. “7 (7 of 9)”, or, in case of Roman instead of arabic numbers, “VII (7 of 9)”, and when different page numbers are used (see below) e. g. arabic after 10 Roman pages: “17 (27 of 30)”. Please try this with the compiled pageslts-example file!

2 Usage

Just load the package by placing

```
\usepackage[<options>]{pageslts}
```

in the preamble of your LaTeX source file and place a `\pagenumbering{...}` with appropriate argument (e. g. arabic, roman, Roman, fnsymbol, alph, or Alph) right behind `\begin{document}` (see subsubsection 2.3.1)!

For example for various draft forms it is desirable to have a page reference to the last page, so that e. g. page footers can contain something like "page N of K", where N is the current page and K is the last page. Once the package is loaded, anywhere in the text references can be made to the labels `LastPage`, `VeryLastPage`, and `LastPages` (most times with `\pageref{...}`, but more save with `\lastpageref{...}`). In particular one can use the fancyhdr (https://ctan.org/pkg/fancyhdr) or nccfancyhdr (https://ctan.org/pkg/nccfancyhdr) package, or redefinitions of the page headings and footings to get a reference to the last page.

`\pageref*` `\lastpageref*` If the hyperref package is used, the references are hyperlinked to their targets. If these hyperlinks shall be suppressed, `\pageref*{...}` and `\lastpageref*{...}` can be used.

2.1 Options

`options` The pageslts package takes the following options:

2.1.1 pagecontinue

`pagecontinue` When option `pagecontinue=false` is **not** given (i. e. `pagecontinue` or `pagecontinue=true` or no `pagecontinue` option at all), at each `\pagenumbering{...}` command the number of the page will be continued with the page number following the last page of the same page numbering scheme. For example, if there are V Roman pages in the frontmatter, some arabic ones in the mainmatter, and then Roman ones again in the backmatter, the last ones will start with VI instead of I again.

If you want to start with I (or i, 1, a, A, *,...) again, set option `pagecontinue=false`. If you want to generally continue the numbers, but for some page numbering scheme do not want this, use `pagecontinue=true` and say `\setcounter{page}{1}` after `\pagenumbering{...}` for that page numbering scheme.

2.1.2 alphMult, AlphMulti, fnsymbolmult

The page number printed in `fnsymbol`[1] must be > 0 and < 10 and those printed in `alph`[2] and `Alph`[3] must be > 0 and < 27. After page Z LaTeX *should* continue with AA, AB, AC,... Some people prefer AA, BB, CC,..., but in hexadecimal it is $AA_{16} = 170_{10}$ and $171_{10} = AB_{16}$, whereas $BB_{16} = 187_{10}$. In any way it should continue at all (maybe even with a user option to choose between the two continuations), but instead only gives an error:

[1] *, †, ‡, §, ¶, ‖, **, ††, ‡‡
[2] a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
[3] A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z

```
LaTeX Error: Counter too large
See the LaTeX manual or LaTeX Companion for explanation.
You've lost some text.  Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.
```

But thanks to the alphalph package these limitation no longer hold. With his `\erroralph` command now even negative or zero page "numbers" are possible.
Nevertheless, some TEX capacity will be exceeded well before the $2\,147\,483\,647^{\text{th}}$ page.

`alphMult` The string option `alphMult` takes three values: `ab`, `bb`, `0`:

ab After page z, the page "numbers" continue with aa, ab, ac, ad,..., and before a with 0, -a, -b,..., -z, -aa, -ab,....

bb After page z, the page "numbers" continue with aa, bb, cc, dd,..., and before a with 0, -a, -b,..., -z, -aa, -bb,....

0 (zero) The pageslts package does nothing, thus the user is free to define the page "numbers" after z and before a.
(But if the user does not do anything at all, the
`LaTeX Error: Counter too large`
will appear again.)

`AlphMulti` The string option `AlphMulti` takes three values: `AB`, `BB`, `0`:

AB After page Z, the page "numbers" continue with AA, AB, AC, AD,..., and before A with 0, -A, -B,..., -Z, -AA, -AB,....

BB After page Z, the page "numbers" continue with AA, BB, CC, DD,..., and before A with 0, -A, -B,..., -Z, -AA, -BB,....

0 (zero) The pageslts package does nothing, thus the user is free to define the page "numbers" after Z and before A.
(But if the user does not do anything at all, the
`LaTeX Error: Counter too large`
will appear again.)

`fnsymbolmult` When option `fnsymbolmult=false` is **not** given (but `fnsymbolmult` or `fnsymbolmult=true` or no `fnsymbolmult` option at all), after 5 (¶) the page "number" is continued with the doubled "number" of the first, second, third,... page (∗∗, ††, ‡‡, §§, ¶¶), and after the tenth page the "number" is tripled (∗ ∗ ∗, † † †,...). Compile the pageslts-example.tex and see the resulting `pdf` file.

Before ∗ (page 1) the page "numbers" are continued with 0, −∗, −†,..., −¶, − ∗ ∗, − † †,...
If this is not wanted, set option `fnsymbolmult=false`, and pageslts will do nothing and allow the user to change the page "number". (But if the user does not do anything at all, the
`LaTeX Error: Counter too large`
will appear again.)

2.1.3 romanMult, RomanMulti

`romanMult` `RomanMulti` The options `romanMult(=true)` and `RomanMulti(=true)` expand the `\roman` and `\Roman` page numbering scheme to values below one (< 1), i. e. 0, -i, -ii, -iii, -iv,... and 0, -I, -II, -III, -IV,..., respectively.
Again the TEX capacity will be exceeded well before ± `MAX` = ±2 147 483 647. If the expansion below 1 is not wanted, set options `romanMult=false` and/or `RomanMulti=false`, and pageslts will do nothing and allow the user to change the page "number". (But if the user does not do anything at all, LATEX will just ignore those values – not even a warning will be issued!)

2.1.4 Arabic page numbers

Arabic page numbers

In LaTeX arabic (page) numbers are already possible between `-MAX...MAX`, where `MAX` = 2 147 483 647 (cf. the alphalph package), without any expansion necessary.

2.2 Labels

pagesLTS.0

At the first page a label `pagesLTS.0` is created. If `\pagenumbering{...}` is used right after `\begin{document}`, this is much easier for the pageslts package (and chances for successful placing of all labels are much higher; cf. subsubsection 2.3.2).

LastPage

`\AddToHook{enddocument/afterlastpage}` (formerly: `\AtEndDocument}` this package defines a label, `LastPage`, which the user can refer to with the `\lastpageref{LastPage}` command. In former times `\pageref{LastPage}` did not work when used together with the hyperref package and the `fnsymbol` page numbering scheme. This should work now.

VeryLastPage

`\AddToHook{enddocument/afterlastpage}` (formerly: `\AfterLastShipout`) the label `VeryLastPage` is defined, which the user can also refer to with the `\lastpageref{VeryLastPage}` command. With the new kernel hooks `LastPage` as well as `VeryLastPage` point to the last page indeed.

LastPages
page number
number of pages

When more than one page numbering scheme is used, neither `LastPage` nor `VeryLastPage` give the total **number** of pages. For example, for a document with VI+36 pages, both give "36" as reference to the last page. While this is correct, the total number of pages is 42, and this is given by the reference to `LastPages`: `\lastpageref{LastPages}` (note the "s" at the end). When the page number was manipulated by `\addtocounter{page}{...}` or `\setcounter{page}{...}`, `LastPages` ignores this. (At a page numbering change the page is automatically reset to one (without option `pagecontinue`). This is done by `\setcounter{page}{1}`, thus this is ignored, too.)

totpages

`\pageref{totpages}` of the totpages package is similar to `\lastpageref{LastPages}`, but while the target for `\pageref{totpages}` is placed `\AtEndDocument`, the target for `\lastpageref{LastPages}` is placed in the hook `enddocument/afterlastpage`, therefore `\lastpageref{LastPages}` is safer to really get the total page number.

\@abspage@last

Nowadays the kernel provides `\@abspage@last`, which contains the number of pages.

\theCurrentPage

`\theCurrentPage` gives the current total/absolute page, in contrast to `\thepage`, which gives only the page *name* in the current page numbering scheme. For example, when there are Roman VII pages in the frontmatter and afterwards in the mainmatter you are at arabic page 9, then `\theCurrentPage` is 16, whereas `\thepage` is 9. When the page "number" (name) is manipulated by `\addtocounter{page}{...}` or `\setcounter{page}{...}`, `\theCurrentPage` ignores this. Because `CurrentPage` is a normal counter, you can also say e. g. `\Roman{CurrentPage}` to get the value in Roman page numbering scheme (e. g. VIII for 8).

\theCurrentPageLocal

`\theCurrentPageLocal` gives the current (arabic) number of pages in the current page numbering scheme. `\thepage` and `\theCurrentPageLocal` are different e. g. when `\addtocounter{page}{...}` or `\setcounter{page}{...}` were used.
`\theCurrentPageLocal` can be printed in other formats, e. g. `\roman{pagesLTS.current.local.roman}`, but probably it only makes sense if page numbering scheme and format are the same, e. g. `\Roman{pagesLTS.current.local.Roman}` or `\Alph{pagesLTS.current.local.Alph}`. `\number\value{pagesLTS.current.local....}` probably make sense even when combined

with another page numbering scheme. And this is exactly what `\theCurrentPageLocal` does: `\def\theCurrentPageLocal{\number\value{pagesLTS.current.local.\pagesLTS@pnc}}`.

pagesLTS. page numbering scheme . number

If you want to refer to the last page of the first, second,... use of a page numbering scheme, you can refer to `pagesLTS.<page numbering scheme>.<number>`, e. g. `\lastpageref{pagesLTS.Roman.1}`, where `<number>` is the occurrence of the page numbering scheme. For details please see page 9.

\lastpageref

In former times for pages with the `fnsymbol` page numbering scheme, `\lastpageref{...}` instead of `\pageref{...}` had to be used. `\lastpageref` can be used for all pages.

\pagenumbering

2.3 \pagenumbering{...}

2.3.1 If \pagenumbering{...} is not used

When the pageslts package is used, but `\pagenumbering{...}` (with an argument like arabic, roman, Roman, fnsymbol, alph, or Alph) is not used, there should be no problem, except that you might need more (!) compiler runs to get all references right, and some references might even be missing (see below). The pageslts package tries to determine the page numbering scheme at the first shipout, but success is not guaranteed. Thus please use `\pagenumbering{...}` at the beginning of your document!
Without `\pagenumbering{<something>}` (`<something>` e. g. = arabic) at the beginning of the document, the page numbers might be given in arabic *by* (class) *default*, but the pageslts package does not know about this without `\pagenumbering{arabic}`. –
The label `pagesLTS.0` is created at the first page even if no `\pagenumbering{...}` command is given. Maybe have a look at the `.aux` file after compiling your document to detect further labels (of other packages, too).

2.3.2 If \pagenumbering{...} is used once

pagesLTS.0

At the first page a label `pagesLTS.0` is created. If `\pagenumbering{...}` is used right after `\begin{document}`, this is much easier for the pageslts package (and chances for successful placing of all labels are much higher).

2.3.3 If \pagenumbering{...} is used more than once

Everything from the preceding subsubsections applies and additionally the following:

When different page numbering schemes are used, e. g. Roman numbers for the frontmatter and arabic numbers for the mainmatter, please use `\pagenumbering{...}` for each of them! Even if you do this, the reference to neither the label `LastPage` nor the label `VeryLastPage` gives the **total** number of pages, but only the number of pages of the last used page numbering scheme (which could be exactly what you want, e. g. if you want to refer to the last page itself and do not want to give the total number of pages).

LastPages

For remediation the label `LastPages` (with "s" at its end) is introduced. Please then refer to this label by `\lastpageref{LastPages}` instead of `LastPage` or `VeryLastPage`.

pagesLTS.arabic
pagesLTS.roman
pagesLTS.Roman
pagesLTS.alph
pagesLTS.Alph
pagesLTS.fnsymbol
pagesLTS.silly
pagesLTS.gobble

Additionally, at the last page of each page numbering scheme a label `pagesLTS.<numbering scheme>` is placed, where `<numbering scheme>` is e. g. arabic, roman, Roman, alph, Alph, fnsymbol, silly, gobble,. . . .
Additional page numbering schemes are unknown for pageslts, see subsection 3.4 Using an unknown page numbering scheme.

2.3.4 If the same \pagenumbering{. . .} scheme is used more than once

Everything from the preceding subsubsections applies and additionally the following:

pagecontinue

If the same page numbering scheme is used twice (or even more often) in one document (e. g. in the frontmatter Roman: I–V, in the mainmatter arabic: 1–20, and in the backmatter again Roman: VI–X), the second time it is used, the page numbering is either continued (option `pagecontinue` or `pagecontinue=true` or no option `pagecontinue`; the default) or reset to one (option `pagecontinue=false`). It is even possible to use a page numbering scheme more than twice.

pagesLTS. page numbering scheme . number

If you want to refer to the last page of the first, second,. . . use of a page numbering scheme, page V in the example above, you can refer to `pagesLTS.<page numbering scheme>.<number>`, e. g. `\lastpageref{pagesLTS.Roman.1}`, where `<number>` is the occurrence of the page numbering scheme.

If you want to refer to the first page of a page numbering scheme, just place a label there, e. g.

```
\pagenumbering{Roman}
\section{Section title\label{RomanSection}}
```

(You know where you use `\pagenumbering{...}` and this is the pageslts package, not the firstpage package).

pagesLTS. page numbering scheme . number .local.cnt

When you want to give the number of pages of each "sector" of the page numbering scheme, you can use `\lastpages{<page numbering scheme>}{<number>}`, where `<page numbering scheme>` is e. g. Roman, arabic,. . . and `<number>` the "sector" number, e. g. `\lastpages{Roman}{2}`. (Internally, the counter has the format `pagesLTS.<page numbering scheme>.<number>.local.cnt`.)
If you used the page numbering scheme Roman for three times, you could say

```
Last Roman page (pagesLTS.Roman): \lastpageref{pagesLTS.Roman}\newline
There are \lastpageref{pagesLTS.Roman.local}~pages with Roman numbers:\newline
\lastpages{Roman}{1}~pages in the first Roman sector
(\pageref{Roman} -- \lastpageref{pagesLTS.Roman.1}),\newline
\lastpages{Roman}{2}~pages in the second Roman sector
(\pageref{Roman2} -- \lastpageref{pagesLTS.Roman.2}), and\newline
\lastpages{Roman}{3}~pages in the third Roman sector
(\pageref{Roman3} -- \lastpageref{pagesLTS.Roman.3}.
```

to get

Last Roman page (pagesLTS.Roman): VIII
There are 8 pages with Roman numbers:
3 pages in the first Roman sector (I – III),
4 pages in the second Roman sector (IV – VII), and
3 pages in the third Roman sector (VIII – X).

(see e. g. the compiled pageslts-example file).

If you want to continue one page numbering scheme, but later on (third use of it, or for another page numbering scheme) want to reset the page number, just say `\setcounter{page}{1}` there.

In your document the code

```
\makeatletter
\renewcommand{\@evenfoot}%
 {\normalsize\slshape DRAFT \today\hfil \upshape page {\thepage} (\theCurrentPage) of\ %
  \lastpageref{pagesLTS.Roman} + \lastpageref{pagesLTS.arabic}\ = \lastpageref{LastPages} pages%
 }
\renewcommand{\@oddfoot}{\@evenfoot}
\makeatother
```

creates footers like

"*DRAFT August 14, 2025* page V (5) of VII + 35 = 42 pages"

or

"*DRAFT August 14, 2025* page 10 (17) of VII + 35 = 42 pages"

in the compiled document (cf. the `pageslts-example` file).

Code like

```
This book has \lastpageref{pagesLTS.Roman}+\lastpageref{pagesLTS.arabic} pages %
(\lastpageref{LastPages} pages in total).
```

produces output like

This book has X+85 pages (95 pages in total).

(when using the hyperref package, the references are even hyperlinked).

If `\addtocounter{page}{...}` or `\setcounter{page}{...}` have been used, the local version of `CurrentPage` can be used, `\theCurrentPageLocal`, see subsection 2.2.

2.4 papermas(s) package

There is a kind of an add-on to this package, the papermas package, which can be used to compute the number of sheets of paper needed to print a document (you can print more than one page of a document on one sheet of paper) as well as the approximate mass of the printout. https://ctan.org/pkg/papermas

3 A few warnings

3.1 Hyperref and repeated page numbers

When two (or more) different page numbering schemes are used, or the page number is reset, or for any other reason there are two pages with the same number (maybe in different format, e. g. 1 and I), and hyperref has not been configured right, this can cause problems. Use hyperref with `plainpages=false` and `pdfpagelabels=true`, and everything should be fine. More details can be found at https://texfaq.org/FAQ-pdfpagelabels.

3.2 showkeys package

When the showkeys package has been loaded in `draft` mode, in the margin for each label a box is displayed with the name of the label. showkeys accomplishes this by redefining `\label`, but pageslts does not always use `\label`, but sometimes writes directly into the `\jobname.aux`-file, and this is generally done after the according page has shipped out, therefore no box can be placed on the preceding page. At least pageslts gives a warning, that showkeys cannot present the respective label.

3.3 lastpage package

This package first started as a revision of the lastpage package of Jeffrey P. Goldberg (jeffrey+news at goldmark dot org), but then I deemed a replacement necessary to accomplish what this package does. For backward compatibility, a label named `LastPage` is provided. Thus `\usepackage{lastpage}` can be replaced by

```
\usepackage[pagecontinue=false,alphMult=0,AlphMulti=0,fnsymbolmult=false,romanMult=false,RomanMulti=false%
  ]{pageslts},
```

if the behaviour of the lastpage package should be simulated. It is also possible to load both packages (if recent versions are used).

3.4 Using an unknown page numbering scheme

While at the time of the latest revision of the pageslts package no other page numbering schemes (e. g. Greek, Hebraic) were known to the maintainer, this package in principle works with every scheme which is recognized by the original `\pagenumbering` command. But the hyperref package only then works with special page names, if the references to those pages are given in a certain way, thus the combination of a new page numbering scheme, the hyperref and the pageslts package might not work. – The pageslts package by itself also works with schemes, which the original `\pagenumbering{...}` does not recognize, but because the original `\pagenumbering{...}` is called by the pageslts package, this might cause an error. Especially if the last page uses this new page numbering scheme, you should check everything double (at least).
And if the number format is unknown to LaTeX, the pages will have no number, and therefore cannot be referenced. You might be able to help yourself by using the hyperref package and manually placing `\phantomsection`s and `\label{...}`s or even `\hypertarget`s and `\href`s.

3.5 Page counter overflow

WITHOUT (!) the use of the alphalph package, the

“ranges of supported counter values are more or less restricted. Only `\arabic` can be used with any counter value TEX supports.

Presentation command	Supported domain	Ignored values	Error message ‘Counter too large’
`\arabic`	`-MAX..MAX`		
`\roman`, `\Roman`	`1..MAX`	`-MAX..0`	
`\alph`, `\Alph`	`1..26`	0	`-MAX..-1, 27..MAX`
`\fnsymbol`	`1..9`	0	`-MAX..-1, 10..MAX`

`MAX` = 2147483647

” (alphalph package manual, 2019/12/09, v2.6, first table, p. 2).
Please see subsubsections 2.1.2 and 2.1.3 for instructions how to overcome these limitations – except ±MAX: When this is exceeded via `\setcounter{<name>}{` something greater than MAX (or smaller then −MAX) `}`, then the error

```
! Number too big.
I can only go up to 2147483647='17777777777="7FFFFFFF,
so I'm using that number instead of yours.
```

will arise. But if the counter has a value of 2 147 483 647 = MAX, and `\addtocounter{<name>}{+1}` is tried, no error is issued, but `\arabic{<name>}` (as well as `\number\value{<name>}`) prints −2147483648, and further `\addtocounter{<name>}{+1}`s give −2147483647, −2147483646 and so on.
For a counter value of −2 147 483 647 = −MAX and `\addtocounter{<name>}{-1}`s after −2147483647 it is printed −2147483648, 2147483647, 2147483646 and so on (without any message in the `log` file about any possible issue).

4 Alternatives

There are similar packages, which do (or do not) similar things. As I neither know what exactly you want to accomplish when using this package (e. g. page number vs. page name, hyperlinks or not), nor what resources your system has (e. g. TEX, LATEX2e, ε-TEX, LATEX-format as recent as 2024-11-01 or newer), here is a list of some possible alternatives:

`LaTeX-kernel` - The number of pages is nowadays available via `\@abspage@last`, `\thetotalpages`, and `\PreviousTotalPages` from the kernel, but when more than one page numbering scheme is used (for example pages I to X and then 1 to 10, thus number of pages "20", but name of the last page "10"), or when or the fnsymbol page numbering scheme is used, or another package has output after this package, or the page numbers exceed a certain range, there might be issues. (Is the total number of pages wanted? Or is the name of the last page sought?)
`\the\ReadonlyShipoutCounter` contains the number of currently shipped out pages, i. e. current page minus one.

`LastPage` - The lastpage package also provides the `LastPage` label (but not `VeryLastPage` or `LastPages`). If you only want this and/or have a quite limited amount of TEX resources, you might want to use that package instead. It includes versions for older LATEX-kernels and even for LATEX2.09.
https://ctan.org/pkg/lastpage

`totpages` - The totpages package provides a `totpages` label similar to `LastPages`, but `\AtEndDocument` instead of hook `enddocument/afterlastpage`. Therefore you should stay with pageslts. The totpages package additionally computes the number of paper sheets needed to (double) print the document (with one, two, three,... pages on one sheet of paper) (which can be achieved also with the papermas package (see subsection 2.4).
https://ctan.org/pkg/totpages

`totalcount` - The totalcount package provides `\totalpages`. If there are only arabic page numbers consecutively running from 1 to the last page, this works. But for example

```
\documentclass{article}
\usepackage[page]{totalcount}
\pagenumbering{Roman}
\begin{document}
\addtocounter{page}{49}
Page \thepage{} of \totalpages
\end{document}
```

prints "Page L of 50", where the number of pages is one (and no hyperlink is provided to the last page even if hyperref is used).
https://ctan.org/pkg/totalcount

`totcount` - The totcount package provides the last value of a counter, thus also the value of the `page` counter. You do not get a hyperlink to the last page, only the numerical value of the last page name is given (i. e. X+72 pages gives 72 instead of 82 as total number of pages), and the number of pages can be changed for example by `\addtocounter`.
https://ctan.org/pkg/totcount

`nofm` - "There is a package nofm.sty available, but some versions of it are defective, and most don't work with fancyhdr because they take over the complete page layout." (Piet van Oostrum: Page layout in LATEX, March 2, 2004, section 16; fancyhdr.pdf)
nofm as of 1991/02/25 (without version number), available at

https://mirror.ctan.org/obsolete/macros/latex209/contrib/misc/nofm.sty,
does not work with e. g. hyperref, redefines `\enddocument` as well as `\@oddhead`, `\@evenhead`, `\@oddfoot`, and `\@evenfoot`.
If you know the (https://CTAN.org) location of a **working** (!) version, please send me an e-mail, thanks!

`count1to` - The count1to package "sets `\count1` to `\count8` with the values of page to subparagraph. `\count9` is used to flag odd pages. ... [T]he code for the TotalPages label" (package manual, 2024-06-13) has been removed from the current package version.
https://ctan.org/pkg/count1to

`zref` - The zref package "implements an extensible referencing system" (package manual, 2023-09-14).
https://ctan.org/pkg/zref

`memoir` - The memoir class provides `\thelastpage` (page number printed on last page) and `\thelastsheet` (number of pages).

(You programmed or found another alternative, which is available at https://CTAN.org?
OK, send an e-mail to me with the name, location at CTAN, and a short notice, and I will probably include it in the list above.)

5 Example

```
⟨*example⟩
\documentclass[british]{article}[2025/01/22]% v1.4n
\usepackage{lipsum}[2021-09-20]% v2.7
\usepackage[draft]{showkeys}[2024/05/23]% v3.21
%%      Use final instead of draft to hide the keys. %%
\usepackage[pdfpagelabels=true,hyperindex=false]{hyperref}[2024-11-05]% v7.01l
\makeatletter
\@ifpackageloaded{hyperref}{% Hypertext links for LaTeX
 \hypersetup{extension=pdf,%
 plainpages=false,%
 pdflang={en},%
 pdftitle={pageslts package example},%
 pdfauthor={H.-Martin Muench},%
 pdfsubject={Example for the pageslts package},%
 pdfkeywords={LaTeX, pageslts},%
 pdfview=Fit,%
 pdfstartview=Fit,%
 pdfpagelayout=SinglePage,%
 bookmarksopen=true%
}}{\usepackage{url}[2013/09/16]}% v3.4
\makeatother
\usepackage[pagecontinue=true,alphMult=ab,AlphMulti=AB,fnsymbolmult=true,%
 romanMult=true,RomanMulti=true]{pageslts}[2025-08-14]% v2.0g
%% These are the default options. %%

\makeatletter
  \renewcommand{\@evenfoot}%
   {Page \thepage\ (\theCurrentPage; local: \theCurrentPageLocal) of %
    \lastpageref{pagesLTS.roman}(\lastpageref{pagesLTS.roman.local}) + %
    \ifpagesLTS@pagecontinue%
      \lastpageref{pagesLTS.Roman}(\lastpageref{pagesLTS.Roman.local}) %
    \else%
      \{\lastpageref{pagesLTS.Roman.1}%
       +\lastpageref{pagesLTS.Roman.2}%
       +\lastpageref{pagesLTS.Roman.3}%
      (\lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
       \the\numexpr%
        \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
       +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
       +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
       \relax)\} %
    \fi + %
    \lastpageref{pagesLTS.arabic}(\lastpageref{pagesLTS.arabic.local}) + %
    \lastpageref{pagesLTS.fnsymbol}(\lastpageref{pagesLTS.fnsymbol.local}) + %
```

```
    \lastpageref{pagesLTS.alph}(\lastpageref{pagesLTS.alph.local}) + %
    \lastpageref{pagesLTS.gobble}(\lastpageref{pagesLTS.gobble.local}) + %
    \lastpageref{pagesLTS.Alph}(\lastpageref{pagesLTS.Alph.local}) = %
    \lastpageref{LastPages} pages.%
   }
  \renewcommand{\@oddfoot}{\@evenfoot}
  \def\pagesLTSexampleArabic{3}
  \def\pagesLTSexamplealph{23}
\makeatother
\setlength{\parindent}{0pt}
\listfiles
\begin{document}
\pagenumbering{roman}
%% Note the first \pagenumbering immediately behind \begin{document}! %%
%%\addtocounter{page}{-2} %%
\section*{Example for pageslts}
\addcontentsline{toc}{section}{Example for pageslts}
\markboth{Example for pageslts}{Example for pageslts}

This example demonstrates the most common uses of package\newline
\textsf{pageslts}, v2.0g as of 2025-08-14 (HMM);\newline
\url{https://ctan.org/pkg/pageslts}.\bigskip

The used options were \texttt{pagecontinue=true},
\texttt{alphMult=ab}, \texttt{AlphMulti=AB}, \linebreak
\texttt{fnsymbolmult=true},
\texttt{romanMult=true}, and \texttt{RomanMulti=true}
(the default\linebreak ones).~-- For more details please see the documentation!\bigskip

\label{keys} To hide the \pageref{keys}{\qquad } use option
\texttt{final} instead of \texttt{draft} with the \textsf{showkeys}
package (or remove the package call from the preamble of
this document).\bigskip

\textbf{Hyperlinks or not:} If the \textsf{hyperref} package is loaded,
the references are also hyperlinked:\bigskip
\smallskip
Page \thepage\ (\theCurrentPage; local: \theCurrentPageLocal) of %
\lastpageref{pagesLTS.roman}(\lastpageref{pagesLTS.roman.local}) + %
\makeatletter%
\ifpagesLTS@pagecontinue%
  \lastpageref{pagesLTS.Roman}(\lastpageref{pagesLTS.Roman.local}) %
\else%
  \{\lastpageref{pagesLTS.Roman.1}%
   +\lastpageref{pagesLTS.Roman.2}%
   +\lastpageref{pagesLTS.Roman.3}%
```

```
  (\lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
   \the\numexpr%
    \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
   \relax)\} %
\fi + %
\makeatother%
\lastpageref{pagesLTS.arabic}(\lastpageref{pagesLTS.arabic.local}) + %
\lastpageref{pagesLTS.fnsymbol}(\lastpageref{pagesLTS.fnsymbol.local}) + %
\lastpageref{pagesLTS.alph}(\lastpageref{pagesLTS.alph.local}) + %
\lastpageref{pagesLTS.Alph}(\lastpageref{pagesLTS.Alph.local}) = %
\lastpageref{LastPages} pages.\bigskip
If the \textsf{hyperref} package is loaded, but the hyperlinks of the
references shall be suppressed, \verb|\pageref*{...}|
and \verb|\lastpageref*{...}| can be used:\\[0.5\baselineskip]
Page \thepage\ (\theCurrentPage; local: \theCurrentPageLocal) of %
\lastpageref*{pagesLTS.roman}(\lastpageref*{pagesLTS.roman.local}) + %
\makeatletter%
\ifpagesLTS@pagecontinue%
  \lastpageref*{pagesLTS.Roman}(\lastpageref*{pagesLTS.Roman.local}) %
\else%
  \{\lastpageref*{pagesLTS.Roman.1}%
   +\lastpageref*{pagesLTS.Roman.2}%
   +\lastpageref*{pagesLTS.Roman.3}%
  (\lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
   \the\numexpr%
    \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
   \relax)\} %
\fi + %
\makeatother%
\lastpageref*{pagesLTS.arabic}(\lastpageref*{pagesLTS.arabic.local}) + %
\lastpageref*{pagesLTS.fnsymbol}(\lastpageref*{pagesLTS.fnsymbol.local}) + %
\lastpageref*{pagesLTS.alph}(\lastpageref*{pagesLTS.alph.local}) + %
\lastpageref*{pagesLTS.Alph}(\lastpageref*{pagesLTS.Alph.local}) = %
\lastpageref*{LastPages} pages.\bigskip

\textbf{Trademarks} appear throughout this example without any
trademark symbol; they are the property of their respective
trademark owner. There is no intention of infringement; the
usage is to the benefit of the trademark owner.\bigskip

\textbf{Tip}: Use \textit{logical page numbers} for the display of the pdf
(in Adobe Acrobat Reader Continuous Release 2024.005.20320:\newline
```

```
Edit $>$ Preferences $>$ Categories: Page Display $>$
Page Content and Information: Use logical page numbers)!\bigskip

You want negative page numbers? Not only arabic, but even roman,
Roman, alph, Alph or fnsymbol ones? No problem, e.\,g.~just give a\newline
\verb|\addtocounter{page}{|-\textit{some number}\verb|}| in the
source code of this example file (or uncomment the prepared line)!
\newpage
\tableofcontents
\newpage%
\makeatletter%
\def\pagesLTS@tmpC{roman}%
\ifx\pagesLTS@pnc\pagesLTS@tmpC%
\else\pagenumbering{roman}%
%% in case the page numbering is changed before,
%% otherwise pagesLTS.current.local.roman on this page would be undefined
\fi%
\makeatother%
\section{roman}

(\texttt{roman} page numbering was started before,
because the page numbering scheme was needed to start at
the first page, of course.)\bigskip

First page (\verb|\lastpageref{pagesLTS.0}|):
\lastpageref{pagesLTS.0}\bigskip

The page (\verb|\thepage|): \thepage \bigskip

Current page (\verb|\theCurrentPage|),
i.\,e.~counted continuously from the first\linebreak page):~\theCurrentPage \bigskip
You can get this also in other formats:
\roman{CurrentPage}, \Roman{CurrentPage}, \arabic{CurrentPage},
\fnsymbol{CurrentPage}, \alph{CurrentPage}, \Alph{CurrentPage}.

CurrentPageLocal (\verb|\theCurrentPageLocal|),
i.\,e.~counted continuously from the first page of the
current page numbering scheme (if option pagecontinue=false: sector)):
\theCurrentPageLocal\newline
You can get also this in other formats, too:
\roman{pagesLTS.current.local.roman}, \Roman{pagesLTS.current.local.roman},
\arabic{pagesLTS.current.local.roman}, \fnsymbol{pagesLTS.current.local.roman},
\alph{pagesLTS.current.local.roman}, \Alph{pagesLTS.current.local.roman},
but probably it only makes sense if page numbering scheme and format are
the same, e.\,g.\newline
\verb|\Roman{pagesLTS.current.local.Roman}|\newline
```

```
or \verb|\Alph{pagesLTS.current.local.Alph}|. \verb|\number\value{...}| could
make sense even if combined with another page numbering scheme.
And this is exactly what \verb|\theCurrentPageLocal| does:\newline
\nolinebreak{\verb|\def\theCurrentPageLocal{\number\value{pagesLTS.current.local.\pagesLTS@pnc}}|.}\bigskip

Last roman page (pagesLTS.roman): \lastpageref{pagesLTS.roman}{\hskip4em }
(There are \lastpageref{pagesLTS.roman.local} pages with roman numbers.)\bigskip

Last Roman page (pagesLTS.Roman): \lastpageref{pagesLTS.Roman}{\hskip3em }%
\makeatletter%
\ifpagesLTS@pagecontinue%
  (There are \lastpageref{pagesLTS.Roman.local}~pages with Roman numbers:\newline
\else%
  (There are %
  \lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
   \the\numexpr%
    \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
   \relax~pages with Roman numbers:\newline
\fi%
\makeatother%
\lastpages{Roman}{1}~pages in the first Roman sector
(\pageref{Roman}{\hskip3em }-\lastpageref{pagesLTS.Roman.1}{\hskip3em }),\newline
\lastpages{Roman}{2}~pages in the second Roman sector
(\pageref{Roman2}{\hskip3em }-\lastpageref{pagesLTS.Roman.2}{\hskip3em }), and\newline
\lastpages{Roman}{3}~pages in the third Roman sector
(\pageref{Roman3}{\hskip3em }-\lastpageref{pagesLTS.Roman.3}{\hskip3em }).)\bigskip

When the option \texttt{pagecontinue=false} is used with the
\textsf{pageslts} package, the \verb|\lastpageref{pagesLTS.Roman}| will
point to the same page as before, but this will have a lower number.
And if that page number was also present in a preceding sector,
the link will go to that page. (Several pages with the same number\ldots)\newline
The \verb|\lastpageref{pagesLTS.Roman.local}| will refer only to the number
of pages in the last sector. To get the total number of Roman pages, %
\begin{verbatim}
\the\numexpr \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
\relax%
\end{verbatim}%
would be necessary.\bigskip

Last arabic page (pagesLTS.arabic): \lastpageref{pagesLTS.arabic}{\hskip5em }
(There are only \lastpageref{pagesLTS.arabic.local} pages with arabic numbers,
```

```
because an \verb|\addtocounter{page}{|\pagesLTSexampleArabic\verb|}| was used.)\bigskip

Last fnsymbol page (pagesLTS.fnsymbol): \lastpageref{pagesLTS.fnsymbol}\newline
(better \verb|\lastpageref{pagesLTS.fnsymbol}| than just\newline
\verb|\pageref{pagesLTS.fnsymbol}|!)\newline
(There are \lastpageref{pagesLTS.fnsymbol.local} pages with fnsymbol numbers.)\bigskip

Last alph page (pagesLTS.alph): \lastpageref{pagesLTS.alph}{\hskip4em }
(There are only \lastpageref{pagesLTS.alph.local} pages with alph numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexamplealph\verb|}| was used.)\bigskip

Last Alph page (pagesLTS.Alph): \lastpageref{pagesLTS.Alph}{\hskip4em }
(There are \lastpageref{pagesLTS.Alph.local} pages with Alph numbers.)\bigskip

Last page's \textit{name} (LastPage): \lastpageref{LastPage}\bigskip

Very last page's \textit{name} (VeryLastPage): \lastpageref{VeryLastPage}\newline
(\texttt{LastPage} and \texttt{VeryLastPage} are now identical, because both are now done
\verb|\AddToHook{enddocument/afterlastpage}| instead of formerly\newline
\verb|\AtEndDocument| and \verb|\AfterLastShipout|, which were different.)\bigskip

Last page's \textit{number} (LastPages): \lastpageref{LastPages}{\hskip3em }
(=~total number of pages)
\newpage
\pagenumbering{Roman}
\section{Roman\label{Roman}}
\subsection{Common Roman page numbering}

First page (\verb|\lastpageref{pagesLTS.0}|):
\lastpageref{pagesLTS.0}\bigskip

The page (\verb|\thepage|): \thepage\bigskip

Current page (\verb|\theCurrentPage|),
i.\,e.~counted continuously from the first page): \theCurrentPage\bigskip

CurrentPageLocal (\verb|\theCurrentPageLocal|),
i.\,e.~counted continuously from the first page of the
current page numbering scheme (if option pagecontinue=false: sector)): \theCurrentPageLocal \bigskip

Last roman page (pagesLTS.roman): \lastpageref{pagesLTS.roman}{\hskip4em }
(There are \lastpageref{pagesLTS.roman.local} pages with roman numbers.)\bigskip

Last Roman page (pagesLTS.Roman): \lastpageref{pagesLTS.Roman}{\hskip3em }%
\makeatletter%
\ifpagesLTS@pagecontinue%
```

```
  (There are \lastpageref{pagesLTS.Roman.local}~pages with Roman numbers:\newline
\else%
  (There are %
  \lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
   \the\numexpr%
    \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
   \relax~pages with Roman numbers:\newline
\fi%
\makeatother%
\lastpages{Roman}{1}~pages in the first Roman sector
(\pageref{Roman}{\hskip3em }-\lastpageref{pagesLTS.Roman.1}{\hskip3em }),\newline
\lastpages{Roman}{2}~pages in the second Roman sector
(\pageref{Roman2}{\hskip3em }-\lastpageref{pagesLTS.Roman.2}{\hskip3em }), and\newline
\lastpages{Roman}{3}~pages in the third Roman sector
(\pageref{Roman3}{\hskip3em }-\lastpageref{pagesLTS.Roman.3}{\hskip3em }).)\bigskip

When the option \texttt{pagecontinue=false} is used with the
\textsf{pageslts} package, the \verb|\lastpageref{pagesLTS.Roman}| will
point to the same page as before, but this will have a lower number.
And if that page number was also present in a preceding sector,
the link will go to that page. (Several pages with the same number\ldots)\newline
The \verb|\lastpageref{pagesLTS.Roman.local}| will refer only to the number
of pages in the last sector. To get the total number of Roman pages, %
\begin{verbatim}
\the\numexpr \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
\relax%
\end{verbatim}%
would be necessary.
\pagebreak

Last arabic page (pagesLTS.arabic): \lastpageref{pagesLTS.arabic}{\hskip5em }
(There are only \lastpageref{pagesLTS.arabic.local} pages with arabic numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexampleArabic\verb|}| was used.)\bigskip

Last fnsymbol page (pagesLTS.fnsymbol): \lastpageref{pagesLTS.fnsymbol}\newline
(better \verb|\lastpageref{pagesLTS.fnsymbol}| than just\newline
\verb|\pageref{pagesLTS.fnsymbol}|!)\bigskip

(There are \lastpageref{pagesLTS.fnsymbol.local} pages with fnsymbol numbers.)\bigskip

Last alph page (pagesLTS.alph): \lastpageref{pagesLTS.alph}{\hskip4em }
(There are only \lastpageref{pagesLTS.alph.local} pages with alph numbers,
```

```
because an \verb|\addtocounter{page}{|\pagesLTSexamplealph\verb|}| was used.)\bigskip

Last Alph page (pagesLTS.Alph): \lastpageref{pagesLTS.Alph}{\hskip4em }
(There are \lastpageref{pagesLTS.Alph.local} pages with Alph numbers.)\bigskip

Last page's \textit{name} (LastPage): \lastpageref{LastPage}\bigskip

Very last page's \textit{name} (VeryLastPage): \lastpageref{VeryLastPage}\newline
(\texttt{LastPage} and \texttt{VeryLastPage} are now identical, because both are now done
\verb|\AddToHook{enddocument/afterlastpage}| instead of formerly\newline
\verb|\AtEndDocument| and \verb|\AfterLastShipout|, which were different.)\bigskip

Last page's \textit{number} (LastPages): \lastpageref{LastPages}{\hskip3em }
(=~total number of pages)

\newpage
\subsection{Last page of first Roman sector}
\verb|\lastpageref{pagesLTS.Roman}| does \textbf{not}
refer to this page (but there: \lastpageref{pagesLTS.Roman}).
When a reference to this page is wanted,\newline
\verb|\lastpageref{pagesLTS.Roman.1}| can be used: \lastpageref{pagesLTS.Roman.1}.\bigskip

There are \lastpages{Roman}{1}~pages (\verb|\lastpages{Roman}{1}|)
in this first Roman sector.\newline
The Roman page numbering scheme is continued later in section~\ref{Roman2})!

\newpage
\pagenumbering{arabic}
\section{arabic}
\subsection{Standard page numbering}

First page (\verb|\lastpageref{pagesLTS.0}|):
\lastpageref{pagesLTS.0}\bigskip

The page (\verb|\thepage|): \thepage \bigskip

Current page (\verb|\theCurrentPage|),
i.\,e.~counted continuously from the first page): \theCurrentPage \bigskip

CurrentPageLocal (\verb|\theCurrentPageLocal|),
i.\,e.~counted continuously from the first page of the
current page numbering scheme (if option pagecontinue=false: sector)): \theCurrentPageLocal \bigskip

Last roman page (pagesLTS.roman): \lastpageref{pagesLTS.roman}{\hskip4em }
(There are \lastpageref{pagesLTS.roman.local} pages with roman numbers.)\bigskip

```

```
Last Roman page (pagesLTS.Roman): \lastpageref{pagesLTS.Roman}{\hskip3em }%
\makeatletter%
\ifpagesLTS@pagecontinue%
  (There are \lastpageref{pagesLTS.Roman.local}~pages with Roman numbers:\newline
\else%
  (There are %
  \lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
   \the\numexpr%
    \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
   \relax~pages with Roman numbers:\newline
\fi%
\makeatother%
\lastpages{Roman}{1}~pages in the first Roman sector
(\pageref{Roman}{\hskip3em }-\lastpageref{pagesLTS.Roman.1}{\hskip3em }),\newline
\lastpages{Roman}{2}~pages in the second Roman sector
(\pageref{Roman2}{\hskip3em }-\lastpageref{pagesLTS.Roman.2}{\hskip3em }), and\newline
\lastpages{Roman}{3}~pages in the third Roman sector
(\pageref{Roman3}{\hskip3em }-\lastpageref{pagesLTS.Roman.3}{\hskip3em }).)\bigskip

When the option \texttt{pagecontinue=false} is used with the
\textsf{pageslts} package, the \verb|\lastpageref{pagesLTS.Roman}| will
point to the same page as before, but this will have a lower number.
And if that page number was also present in a preceding sector,
the link will go to that page. (Several pages with the same number\ldots)\newline
The \verb|\lastpageref{pagesLTS.Roman.local}| will refer only to the number
of pages in the last sector. To get the total number of Roman pages, %
\begin{verbatim}
\the\numexpr \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
\relax%
\end{verbatim}%
would be necessary.
\pagebreak

Last arabic page (pagesLTS.arabic): \lastpageref{pagesLTS.arabic}{\hskip5em }
(There are only \lastpageref{pagesLTS.arabic.local} pages with arabic numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexampleArabic\verb|}| was used.)\bigskip

Last fnsymbol page (pagesLTS.fnsymbol): \lastpageref{pagesLTS.fnsymbol}\newline
(better \verb|\lastpageref{pagesLTS.fnsymbol}| than just\newline
\verb|\pageref{pagesLTS.fnsymbol}|!)\bigskip

(There are \lastpageref{pagesLTS.fnsymbol.local} pages with fnsymbol numbers.)\bigskip
```

```
413
414 Last alph page (pagesLTS.alph): \lastpageref{pagesLTS.alph}{\hskip4em }
415 (There are only \lastpageref{pagesLTS.alph.local} pages with alph numbers,
416 because an \verb|\addtocounter{page}{|\pagesLTSexamplealph\verb|}| was used.)\bigskip
417
418 Last Alph page (pagesLTS.Alph): \lastpageref{pagesLTS.Alph}{\hskip4em }
419 (There are \lastpageref{pagesLTS.Alph.local} pages with Alph numbers.)\bigskip
420
421 Last page's \textit{name} (LastPage): \lastpageref{LastPage}\bigskip
422
423 Very last page's \textit{name} (VeryLastPage): \lastpageref{VeryLastPage}\newline
424 (\texttt{LastPage} and \texttt{VeryLastPage} are now identical, because both are now done
425 \verb|\AddToHook{enddocument/afterlastpage}| instead of formerly\newline
426 \verb|\AtEndDocument| and \verb|\AfterLastShipout|, which were different.)\bigskip
427
428 Last page's \textit{number} (LastPages): \lastpageref{LastPages}{\hskip3em }
429 (=~total number of pages)
430
431 \newpage
432 \thispagestyle{empty}
433 \subsection[Empty page style]{Also an empty page style is no problem %
434 for the current or total page count}
435
436 \bigskip
437
438 First page (\verb|\lastpageref{pagesLTS.0}|):
439 \lastpageref{pagesLTS.0}\bigskip
440
441 The page (\verb|\thepage|): \thepage \bigskip
442
443 Current page (\verb|\theCurrentPage|),
444 i.\,e.~counted continuously from the first page): \theCurrentPage \bigskip
445
446 CurrentPageLocal (\verb|\theCurrentPageLocal|),
447 i.\,e.~counted continuously from the first page of the
448 current page numbering scheme (if option pagecontinue=false: sector)): \theCurrentPageLocal \bigskip
449
450 Last roman page (pagesLTS.roman): \lastpageref{pagesLTS.roman}{\hskip4em }
451 (There are \lastpageref{pagesLTS.roman.local} pages with roman numbers.)\bigskip
452
453 Last Roman page (pagesLTS.Roman): \lastpageref{pagesLTS.Roman}{\hskip3em }%
454 \makeatletter%
455 \ifpagesLTS@pagecontinue%
456   (There are \lastpageref{pagesLTS.Roman.local}~pages with Roman numbers:\newline
457 \else%
458   (There are %
```

```
 \lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
  \the\numexpr%
   \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
  +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
  +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
  \relax~pages with Roman numbers:\newline
\fi%
\makeatother%
\lastpages{Roman}{1}~pages in the first Roman sector
(\pageref{Roman}{\hskip3em }-\lastpageref{pagesLTS.Roman.1}{\hskip3em }),\newline
\lastpages{Roman}{2}~pages in the second Roman sector
(\pageref{Roman2}{\hskip3em }-\lastpageref{pagesLTS.Roman.2}{\hskip3em }), and\newline
\lastpages{Roman}{3}~pages in the third Roman sector
(\pageref{Roman3}{\hskip3em }-\lastpageref{pagesLTS.Roman.3}{\hskip3em }).)\newline

When the option \texttt{pagecontinue=false} is used with the
\textsf{pageslts} package, the \verb|\lastpageref{pagesLTS.Roman}| will
point to the same page as before, but this will have a lower number.
And if that page number was also present in a preceding sector,
the link will go to that page. (Several pages with the same number\ldots)\newline
The \verb|\lastpageref{pagesLTS.Roman.local}| will refer only to the number
of pages in the last sector. To get the total number of Roman pages, %
\begin{verbatim}
\the\numexpr \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
\relax%
\end{verbatim}%
would be necessary.
\pagebreak

Last arabic page (pagesLTS.arabic): \lastpageref{pagesLTS.arabic}{\hskip5em }
(There are only \lastpageref{pagesLTS.arabic.local} pages with arabic numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexampleArabic\verb|}| was used.)\bigskip

Last fnsymbol page (pagesLTS.fnsymbol): \lastpageref{pagesLTS.fnsymbol}\newline
(better \verb|\lastpageref{pagesLTS.fnsymbol}| than just\newline
\verb|\pageref{pagesLTS.fnsymbol}|!)\bigskip

(There are \lastpageref{pagesLTS.fnsymbol.local} pages with fnsymbol numbers.)\bigskip

Last alph page (pagesLTS.alph): \lastpageref{pagesLTS.alph}{\hskip4em }
(There are only \lastpageref{pagesLTS.alph.local} pages with alph numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexamplealph\verb|}| was used.)\bigskip

Last Alph page (pagesLTS.Alph): \lastpageref{pagesLTS.Alph}{\hskip4em }
```

```
(There are \lastpageref{pagesLTS.Alph.local} pages with Alph numbers.)\bigskip

Last page's \textit{name} (LastPage): \lastpageref{LastPage}\bigskip

Very last page's \textit{name} (VeryLastPage): \lastpageref{VeryLastPage}\newline
(\texttt{LastPage} and \texttt{VeryLastPage} are now identical, because both are now done
\verb|\AddToHook{enddocument/afterlastpage}| instead of formerly\newline
\verb|\AtEndDocument| and \verb|\AfterLastShipout|, which were different.)\bigskip

Last page's \textit{number} (LastPages): \lastpageref{LastPages}{\hskip3em }
(=~total number of pages)

\newpage
\subsection[addtocounter, setcounter]{Neither %
\texttt{\textbackslash addtocounter\{page\}} nor %
\texttt{\textbackslash setcounter\{page\}} is a problem for the %
current or total page numbers}

(Here is an \verb|\addtocounter{page}{|\pagesLTSexampleArabic\verb|}|
in the source code.)
\addtocounter{page}{\pagesLTSexampleArabic}\bigskip

The page (from \verb|\thepage| command): \thepage \bigskip

Current page (from \verb|\theCurrentPage|\ %
command), i.\,e.~counted continuously from the first page): \theCurrentPage \bigskip

CurrentPageLocal (from \verb|\theCurrentPageLocal|\ %
command), i.\,e.~counted continuously from the first page of the
current page numbering scheme (if option pagecontinue=false: sector)): \theCurrentPageLocal \bigskip

Last page's number (LastPages): \lastpageref{LastPages}{\hskip3em }
(= total number of pages)

\newpage
\pagenumbering{fnsymbol}
\section{fnsymbol}

Without option \texttt{fnsymbolmult=true} of the \textsf{pageslts} package
(and the help of the \textsf{alphalph} package),
after page~9 (\textquotedblleft \ensuremath {\ddagger \ddagger }\textquotedblright )
(and also for negative page numbers) there would just appear a
\begin{quote}
\begin{verbatim}
LaTeX Error: Counter too large
See the LaTeX manual or LaTeX Companion for explanation.
```

```
You've lost some text.  Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.
\end{verbatim}
\end{quote}
Now the page numbers after 5 (\ensuremath {\mathparagraph }) are
continued  with the doubled \textquotedblleft number\textquotedblright{} of
the first, second, third,\ldots\ page (\ensuremath {**},
\ensuremath {\dagger \dagger }, \ensuremath {\ddagger \ddagger },
\ensuremath {\mathsection \mathsection },
\ensuremath {\mathparagraph \mathparagraph }),
and after the tenth page the \textquotedblleft number\textquotedblright{} is
tripled (\ensuremath {***}, \ensuremath {\dagger \dagger \dagger },\ldots).
Page zero is named 0 and negative pages are just named like the positive ones
with addition of a minus \mbox{sign ($-$).}

\bigskip

For testing purposes, try:

\begin{small}
\begin{verbatim}
\documentclass{article}
\usepackage[plainpages=false,pageanchor=true,pdfpagelabels=true]{hyperref}
\usepackage{pageslts}
\pagenumbering{fnsymbol}
\begin{document}
First page.\newpage
\verb|\lastpageref{pagesLTS.0}                | \lastpageref{pagesLTS.0}                \par
\verb|\lastpageref{pagesLTS.0.local}          | \lastpageref{pagesLTS.0.local}          \par
\verb|\lastpageref{LastPage}                  | \lastpageref{LastPage}                  \par
\verb|\lastpageref{VeryLastPage}              | \lastpageref{VeryLastPage}              \par
\verb|\lastpageref{pagesLTS.fnsymbol.1}       | \lastpageref{pagesLTS.fnsymbol.1}       \par
\verb|\lastpageref{pagesLTS.fnsymbol.1.local}| \lastpageref{pagesLTS.fnsymbol.1.local}\par
\verb|\lastpageref{pagesLTS.fnsymbol}         | \lastpageref{pagesLTS.fnsymbol}         \par
\verb|\lastpageref{pagesLTS.fnsymbol.local}   | \lastpageref{pagesLTS.fnsymbol.local}   \par
\verb|\lastpageref{LastPages}                 | \lastpageref{LastPages}                 \par
\verb|\pagesLTSlastpage                       | \pagesLTSlastpage
\newpage
Last page.
\end{document}
\end{verbatim}
\end{small}
\newpage

First page (\verb|\lastpageref{pagesLTS.0}|):
\lastpageref{pagesLTS.0}\bigskip
```

```
The page (\verb|\thepage|): \thepage \bigskip

Current page (\verb|\theCurrentPage|),
i.\,e.~counted continuously from the first page): \theCurrentPage \bigskip

CurrentPageLocal (\verb|\theCurrentPageLocal|),
i.\,e.~counted continuously from the first page of the
current page numbering scheme (if option pagecontinue=false: sector)): \theCurrentPageLocal \bigskip

Last roman page (pagesLTS.roman): \lastpageref{pagesLTS.roman}{\hskip4em }
(There are \lastpageref{pagesLTS.roman.local} pages with roman numbers.)\bigskip

Last Roman page (pagesLTS.Roman): \lastpageref{pagesLTS.Roman}{\hskip3em }%
\makeatletter%
\ifpagesLTS@pagecontinue%
  (There are \lastpageref{pagesLTS.Roman.local}~pages with Roman numbers:\newline
\else%
  (There are %
  \lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
   \the\numexpr%
    \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
   \relax~pages with Roman numbers:\newline
\fi%
\makeatother%
\lastpages{Roman}{1}~pages in the first Roman sector
(\pageref{Roman}{\hskip3em }-\lastpageref{pagesLTS.Roman.1}{\hskip3em }),\newline
\lastpages{Roman}{2}~pages in the second Roman sector
(\pageref{Roman2}{\hskip3em }-\lastpageref{pagesLTS.Roman.2}{\hskip3em }), and\newline
\lastpages{Roman}{3}~pages in the third Roman sector
(\pageref{Roman3}{\hskip3em }-\lastpageref{pagesLTS.Roman.3}{\hskip3em }).)\bigskip

When the option \texttt{pagecontinue=false} is used with the
\textsf{pageslts} package, the \verb|\lastpageref{pagesLTS.Roman}| will
point to the same page as before, but this will have a lower number.
And if that page number was also present in a preceding sector,
the link will go to that page. (Several pages with the same number\ldots)\newline
The \verb|\lastpageref{pagesLTS.Roman.local}| will refer only to the number
of pages in the last sector. To get the total number of Roman pages, %
\begin{verbatim}
\the\numexpr \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
\relax%
```

```
\end{verbatim}%
would be necessary.\bigskip

Last arabic page (pagesLTS.arabic): \lastpageref{pagesLTS.arabic}{\hskip5em }
(There are only \lastpageref{pagesLTS.arabic.local} pages with arabic numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexampleArabic\verb|}| was used.)\bigskip

Last fnsymbol page (pagesLTS.fnsymbol): \lastpageref{pagesLTS.fnsymbol}\newline
(better \verb|\lastpageref{pagesLTS.fnsymbol}| than just\newline
\verb|\pageref{pagesLTS.fnsymbol}|!)\bigskip

(There are \lastpageref{pagesLTS.fnsymbol.local} pages with fnsymbol numbers.)\bigskip

Last alph page (pagesLTS.alph): \lastpageref{pagesLTS.alph}{\hskip4em }
(There are only \lastpageref{pagesLTS.alph.local} pages with alph numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexamplealph\verb|}| was used.)\bigskip

Last Alph page (pagesLTS.Alph): \lastpageref{pagesLTS.Alph}{\hskip4em }
(There are \lastpageref{pagesLTS.Alph.local} pages with Alph numbers.)\bigskip

Last page's \textit{name} (LastPage): \lastpageref{LastPage}\bigskip

Very last page's \textit{name} (VeryLastPage): \lastpageref{VeryLastPage}\newline
(\texttt{LastPage} and \texttt{VeryLastPage} are now identical, because both are now done
\verb|\AddToHook{enddocument/afterlastpage}| instead of formerly\newline
\verb|\AtEndDocument| and \verb|\AfterLastShipout|, which were different.)\bigskip

Last page's \textit{number} (LastPages): \lastpageref{LastPages}{\hskip3em }
(=~total number of pages)\bigskip

\lipsum[1-58]

\newpage
\pagenumbering{Roman}
\section{Roman -- again!\label{Roman2}}

The page number starts with \textquotedblleft I\textquotedblright{} again for
option just \texttt{pagecontinue=false}, but for option \texttt{pagecontinue=true},
or with option just \texttt{pagecontinue}, or even just
\nolinebreak{with\textbf{out}} option \texttt{pagecontinue=false}
the page numbers are continued.

This package remembered the
$\number\value{pagesLTS.double.Roman}-1$,
i.\,e.~\the\numexpr\value{pagesLTS.double.Roman}-1\relax, pages
already done in Roman output, and therefore continues with page
```

```
\textquotedblleft\Roman{pagesLTS.double.Roman}\textquotedblright{} in
case of option \texttt{pagecontinue}.\newline
If you want to start with \textquotedblleft I\textquotedblright{} all
over again, you will have two pages with the same name,
but nevertheless you can do this by using option \texttt{pagecontinue=false}
or a \verb|\setcounter{page}{1}|\ here (not demonstrated in this example file).\bigskip

There was the question: \textquotedblleft With option \verb|pagecontinue=false|,
how do I get the number of pages in each number block (e.\,g.~in the first Roman section,
second Roman section, third Roman section) with just one command?\textquotedblright
\begin{verbatim}
\makeatletter%
\theCurrentPageLocal~of~\lastpageref*{%
pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local}%
\makeatother%
\end{verbatim}
provides this. Here: %
\makeatletter%
\theCurrentPageLocal~of~%
\lastpageref*{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local}\newline%
\makeatother%
for \verb|pagecontinue=true| is \verb|4 of 8| (including %
\makeatletter%
$\the\numexpr\csname c@pagesLTS.double.Roman\endcsname -1\relax$ %
from the first Roman section,
$\the\csname c@pagesLTS.double.Roman\endcsname\relax
 =\the\numexpr\csname c@pagesLTS.double.Roman\endcsname -1\relax% first section
+1% first page of this second section
$, $\the\numexpr\csname c@pagesLTS.double.Roman\endcsname -1+%
\csname c@pagesLTS.Roman.2.local.cnt\endcsname\relax%
=\the\numexpr\csname c@pagesLTS.double.Roman\endcsname -1\relax+%
\the\numexpr\csname c@pagesLTS.Roman.2.local.cnt\endcsname\relax$)\newline
\makeatother%
and for \verb|pagecontinue=false| is \verb|1 of 5| .\bigskip

\verb|pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local| is:
\begin{description}
\item[-] \verb|pagesLTS| just identifies this counter to belong to the pageslts package.
\item[-] \verb|\pagesLTS@pnc| is the Current Page Numbering scheme, i.\,e.~here it would be \verb|Roman|.
\item[-] \verb|\number\value{pagesLTS.pnc.\pagesLTS@pnc}| is the value of counter\newline
          \verb|pagesLTS.pnc.\pagesLTS@pnc| printed in arabic numbers (1, 2, 3, ...) -- see below.
\item[-] \verb|local|: Local there are 5 pages with Roman numbering in this example, globally there are
          $3+5+4=12$~pages with Roman numbering in this example, and the local numbers are wanted.
\end{description}
The counter \verb|pagesLTS.pnc.\pagesLTS@pnc| gives the number of the\newline
\verb|\pagesLTS@pnc| page numbering scheme currently used, i.\,e.~we have consecutivly\newpage
```

```
\begin{verbatim}
pagesLTS.pnc.roman    = 1
pagesLTS.pnc.Roman    = 1
pagesLTS.pnc.arabic   = 1
pagesLTS.pnc.fnsymbol = 1
pagesLTS.pnc.Roman    = 2
pagesLTS.pnc.alph     = 1
pagesLTS.pnc.Roman    = 3
pagesLTS.pnc.Alph     = 1
\end{verbatim}
So, \verb|pagesLTS.pnc.Roman| changes from $1$ to $2$ automatically
when doing the second \verb|\pagenumbering{Roman}|,
and changes from $2$ to $3$ for the third\newline
\verb|\pagenumbering{Roman}|.\bigskip

First page (\verb|\lastpageref{pagesLTS.0}|):
 \lastpageref{pagesLTS.0}\bigskip

The page (\verb|\thepage|): \thepage \bigskip

Current page (\verb|\theCurrentPage|),
i.\,e.~counted continuously from the first page): \theCurrentPage \bigskip

CurrentPageLocal (\verb|\theCurrentPageLocal|),
i.\,e.~counted continuously from the first page of the
current page numbering scheme (if option pagecontinue=false: sector)): \theCurrentPageLocal \bigskip

Last roman page (pagesLTS.roman): \lastpageref{pagesLTS.roman}{\hskip4em }
(There are \lastpageref{pagesLTS.roman.local} pages with roman numbers.)\bigskip


Last Roman page (pagesLTS.Roman): \lastpageref{pagesLTS.Roman}{\hskip3em }%
\makeatletter%
\ifpagesLTS@pagecontinue%
  (There are \lastpageref{pagesLTS.Roman.local}~pages with Roman numbers:\newline
\else%
  (There are %
  \lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
   \the\numexpr%
    \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
   \relax~pages with Roman numbers:\newline
\fi%
\makeatother%
\lastpages{Roman}{1}~pages in the first Roman sector
```

```
(\pageref{Roman}{\hskip3em }-\lastpageref{pagesLTS.Roman.1}{\hskip3em }),\newline
\lastpages{Roman}{2}~pages in the second Roman sector
(\pageref{Roman2}{\hskip3em }-\lastpageref{pagesLTS.Roman.2}{\hskip3em }), and\newline
\lastpages{Roman}{3}~pages in the third Roman sector
(\pageref{Roman3}{\hskip3em }-\lastpageref{pagesLTS.Roman.3}{\hskip3em }).)\bigskip

When the option \texttt{pagecontinue=false} is used with the
\textsf{pageslts} package, the \verb|\lastpageref{pagesLTS.Roman}| will
point to the same page as before, but this will have a lower number.
And if that page number was also present in a preceding sector,
the link will go to that page. (Several pages with the same number\ldots)\newline
The \verb|\lastpageref{pagesLTS.Roman.local}| will refer only to the number
of pages in the last sector. To get the total number of Roman pages, %
\begin{verbatim}
\the\numexpr \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
\relax%
\end{verbatim}%
would be necessary.\bigskip

Last arabic page (pagesLTS.arabic): \lastpageref{pagesLTS.arabic}{\hskip5em }
(There are only \lastpageref{pagesLTS.arabic.local} pages with arabic numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexampleArabic\verb|}| was used.)\bigskip

Last fnsymbol page (pagesLTS.fnsymbol): \lastpageref{pagesLTS.fnsymbol}\newline
(\verb|\lastpageref{pagesLTS.fnsymbol}| -- never
\verb|\pageref{pagesLTS.fnsymbol}|!)\newline
(There are \lastpageref{pagesLTS.fnsymbol.local} pages with fnsymbol numbers.)\bigskip

Last alph page (pagesLTS.alph): \lastpageref{pagesLTS.alph}{\hskip4em }
(There are only \lastpageref{pagesLTS.alph.local} pages with alph numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexamplealph\verb|}| was used.)\bigskip

Last Alph page (pagesLTS.Alph): \lastpageref{pagesLTS.Alph}{\hskip4em }
(There are \lastpageref{pagesLTS.Alph.local} pages with Alph numbers.)\bigskip

Last page's \textit{name} (LastPage): \lastpageref{LastPage}\bigskip

Very last page's \textit{name} (VeryLastPage): \lastpageref{VeryLastPage}\newline
(\texttt{LastPage} and \texttt{VeryLastPage} are now identical, because both are now done
\verb|\AddToHook{enddocument/afterlastpage}| instead of formerly\newline
\verb|\AtEndDocument| and \verb|\AfterLastShipout|, which were different.)\bigskip

Last page's \textit{number} (LastPages): \lastpageref{LastPages}{\hskip3em }
(=~total number of pages)\bigskip
```

```
\lipsum[1-6]

\newpage

\verb|\lastpageref{pagesLTS.Roman}| does \textbf{not}
refer to this page (but there: \lastpageref{pagesLTS.Roman})
for option \texttt{pagecontinue=true}. When a reference to this page is wanted,\newline
\verb|\lastpageref{pagesLTS.Roman.2}| can be used: \lastpageref{pagesLTS.Roman.2}.\newline
When the option \texttt{pagecontinue=false} is used with the
\textsf{pageslts} package, the \verb|\lastpageref{pagesLTS.Roman}| will
point to the same page as before, but this will have a lower number.
And if that page number was also present in a preceding sector,
the link will go to that page. (Several pages with the same number\ldots)\newline
The \verb|\lastpageref{pagesLTS.Roman.local}| will refer only to the number
of pages in the last sector. To get the total number of Roman pages, %
\begin{verbatim}
\the\numexpr \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
\relax%
\end{verbatim}%
would be necessary.\bigskip

There are \lastpages{Roman}{2}~pages (\verb|\lastpages{Roman}{2}|) in this
second Roman sector.\newline
The Roman page numbering scheme is continued later in section~\ref{Roman3})!

\newpage
\pagenumbering{alph}
\section{alph\label{alph}}

First page (\verb|\lastpageref{pagesLTS.0}|):
 \lastpageref{pagesLTS.0}\bigskip

The page (\verb|\thepage|): \thepage \bigskip

Current page (\verb|\theCurrentPage|),
i.\,e.~counted continuously from the first page): \theCurrentPage \bigskip

CurrentPageLocal (\verb|\theCurrentPageLocal|),
i.\,e.~counted continuously from the first page of the
current page numbering scheme (if option pagecontinue=false: sector)): \theCurrentPageLocal \bigskip

Last roman page (pagesLTS.roman): \lastpageref{pagesLTS.roman}{\hskip4em }
(There are \lastpageref{pagesLTS.roman.local} pages with roman numbers.)\bigskip
```

```
Last Roman page (pagesLTS.Roman): \lastpageref{pagesLTS.Roman}{\hskip3em }%
\makeatletter%
\ifpagesLTS@pagecontinue%
  (There are \lastpageref{pagesLTS.Roman.local}~pages with Roman numbers:\newline
\else%
  (There are %
  \lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
   \the\numexpr%
    \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
   \relax~pages with Roman numbers:\newline
\fi%
\makeatother%
\lastpages{Roman}{1}~pages in the first Roman sector
(\pageref{Roman}{\hskip3em }-\lastpageref{pagesLTS.Roman.1}{\hskip3em }),\newline
\lastpages{Roman}{2}~pages in the second Roman sector
(\pageref{Roman2}{\hskip3em }-\lastpageref{pagesLTS.Roman.2}{\hskip3em }), and\newline
\lastpages{Roman}{3}~pages in the third Roman sector
(\pageref{Roman3}{\hskip3em }-\lastpageref{pagesLTS.Roman.3}{\hskip3em }).)\bigskip

When the option \texttt{pagecontinue=false} is used with the
\textsf{pageslts} package, the \verb|\lastpageref{pagesLTS.Roman}| will
point to the same page as before, but this will have a lower number.
And if that page number was also present in a preceding sector,
the link will go to that page. (Several pages with the same number\ldots)\newline
The \verb|\lastpageref{pagesLTS.Roman.local}| will refer only to the number
of pages in the last sector. To get the total number of Roman pages, %
\begin{verbatim}
\the\numexpr \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
\relax%
\end{verbatim}%
would be necessary.\bigskip

Last arabic page (pagesLTS.arabic): \lastpageref{pagesLTS.arabic}{\hskip5em }
(There are only \lastpageref{pagesLTS.arabic.local} pages with arabic numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexampleArabic\verb|}| was used.)\bigskip

Last fnsymbol page (pagesLTS.fnsymbol): \lastpageref{pagesLTS.fnsymbol}\newline
(better \verb|\lastpageref{pagesLTS.fnsymbol}| than just\newline
\verb|\pageref{pagesLTS.fnsymbol}|!)\bigskip

(There are \lastpageref{pagesLTS.fnsymbol.local} pages with fnsymbol numbers.)\bigskip
```

```
Last alph page (pagesLTS.alph): \lastpageref{pagesLTS.alph}{\hskip4em }
(There are only \lastpageref{pagesLTS.alph.local} pages with alph numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexamplealph\verb|}| was used.)\bigskip

Last Alph page (pagesLTS.Alph): \lastpageref{pagesLTS.Alph}{\hskip4em }
(There are \lastpageref{pagesLTS.Alph.local} pages with Alph numbers.)\bigskip

Last page's \textit{name} (LastPage): \lastpageref{LastPage}\bigskip

Very last page's \textit{name} (VeryLastPage): \lastpageref{VeryLastPage}\newline
(\texttt{LastPage} and \texttt{VeryLastPage} are now identical, because both are now done
\verb|\AddToHook{enddocument/afterlastpage}| instead of formerly\newline
\verb|\AtEndDocument| and \verb|\AfterLastShipout|, which were different.)\bigskip

Last page's \textit{number} (LastPages): \lastpageref{LastPages}{\hskip3em }
(=~total number of pages)

\newpage

Without option \texttt{alphMult=ab} of the \textsf{pageslts} (and the help of the
\textsf{alphalph} package), after page \textquotedblleft z\textquotedblright{} there
would just appear a
\begin{quote}
\begin{verbatim}
LaTeX Error: Counter too large
See the LaTeX manual or LaTeX Companion for explanation.
You've lost some text.  Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.
\end{verbatim}
\end{quote}
Now the page numbers are continued aa, ab, ac,\ldots\ (aa, bb, cc,\ldots\ is
also possible, see the \textsf{pageslts} documentation).\newline
To demonstrate this, we add a\newline
\verb|\addtocounter{page}{|\pagesLTSexamplealph\verb|}|\newline
in the source code here.

\addtocounter{page}{\pagesLTSexamplealph}

\bigskip

\lipsum[1-18]

\newpage
\pagenumbering{Roman}
\section{Roman - third time!\label{Roman3}}
```

```
The page number starts with \textquotedblleft I\textquotedblright{} again for
option just \texttt{pagecontinue=false}, but for option \texttt{pagecontinue=true},
or with option just \texttt{pagecontinue}, or even just
\nolinebreak{with\textbf{out}} option \texttt{pagecontinue=false}
the page numbers are continued.

This package remembered the
$\number\value{pagesLTS.double.Roman}-1$,
i.\,e.~\the\numexpr\value{pagesLTS.double.Roman}-1\relax, pages
already done in%
\begin{description}
\item[-] for \texttt{pagecontinue=false}: the preceding Roman sector,
\item[-] for \texttt{pagecontinue=true}: all Roman sectors,
\end{description}%
and therefore continues with the appropriate page in case of option \texttt{pagecontinue}.\newline
If you want to start with \textquotedblleft I\textquotedblright{} all
over again, you will have two pages with the same name,
but nevertheless you can do this by using option \texttt{pagecontinue=false}
or a \verb|\setcounter{page}{1}|\ here (not demonstrated in this example file).\bigskip

First page (\verb|\lastpageref{pagesLTS.0}|):
 \lastpageref{pagesLTS.0}\bigskip

The page (\verb|\thepage|): \thepage \bigskip

Current page (\verb|\theCurrentPage|),
i.\,e.~counted continuously from the first page): \theCurrentPage \bigskip

CurrentPageLocal (\verb|\theCurrentPageLocal|),
i.\,e.~counted continuously from the first page of the
current page numbering scheme (if option pagecontinue=false: sector)): \theCurrentPageLocal \bigskip

Last roman page (pagesLTS.roman): \lastpageref{pagesLTS.roman}{\hskip4em }
(There are \lastpageref{pagesLTS.roman.local} pages with roman numbers.)\bigskip

Last Roman page (pagesLTS.Roman): \lastpageref{pagesLTS.Roman}{\hskip3em }%
\makeatletter%
\ifpagesLTS@pagecontinue%
  (There are \lastpageref{pagesLTS.Roman.local}~pages with Roman numbers:\newline
\else%
  (There are %
  \lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
   \the\numexpr%
    \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
```

```
   +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
   \relax~pages with Roman numbers:\newline
\fi%
\makeatother%
\lastpages{Roman}{1}~pages in the first Roman sector
(\pageref{Roman}{\hskip3em }-\lastpageref{pagesLTS.Roman.1}{\hskip3em }),\newline
\lastpages{Roman}{2}~pages in the second Roman sector
(\pageref{Roman2}{\hskip3em }-\lastpageref{pagesLTS.Roman.2}{\hskip3em }), and\newline
\lastpages{Roman}{3}~pages in the third Roman sector
(\pageref{Roman3}{\hskip3em }-\lastpageref{pagesLTS.Roman.3}{\hskip3em }).)\bigskip

When the option \texttt{pagecontinue=false} is used with the
\textsf{pageslts} package, the \verb|\lastpageref{pagesLTS.Roman}| will
point to the same page as before, but this will have a lower number.
And if that page number was also present in a preceding sector,
the link will go to that page. (Several pages with the same number\ldots)\newline
The \verb|\lastpageref{pagesLTS.Roman.local}| will refer only to the number
of pages in the last sector. To get the total number of Roman pages, %
\begin{verbatim}
\the\numexpr \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
\relax%
\end{verbatim}%
would be necessary.\bigskip

Last arabic page (pagesLTS.arabic): \lastpageref{pagesLTS.arabic}{\hskip5em }
(There are only \lastpageref{pagesLTS.arabic.local} pages with arabic numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexampleArabic\verb|}| was used.)\bigskip

Last fnsymbol page (pagesLTS.fnsymbol): \lastpageref{pagesLTS.fnsymbol}\newline
(better \verb|\lastpageref{pagesLTS.fnsymbol}| than just\newline
\verb|\pageref{pagesLTS.fnsymbol}|!)\bigskip

(There are \lastpageref{pagesLTS.fnsymbol.local} pages with fnsymbol numbers.)\bigskip

Last alph page (pagesLTS.alph): \lastpageref{pagesLTS.alph}{\hskip4em }
(There are only \lastpageref{pagesLTS.alph.local} pages with alph numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexamplealph\verb|}| was used.)\bigskip

Last Alph page (pagesLTS.Alph): \lastpageref{pagesLTS.Alph}{\hskip4em }
(There are \lastpageref{pagesLTS.Alph.local} pages with Alph numbers.)\bigskip

Last page's \textit{name} (LastPage): \lastpageref{LastPage}\bigskip

Very last page's \textit{name} (VeryLastPage): \lastpageref{VeryLastPage}\newline
```

```
(\texttt{LastPage} and \texttt{VeryLastPage} are now identical, because both are now done
\verb|\AddToHook{enddocument/afterlastpage}| instead of formerly\newline
\verb|\AtEndDocument| and \verb|\AfterLastShipout|, which were different.)\bigskip

Last page's \textit{number} (LastPages): \lastpageref{LastPages}{\hskip3em }
(=~total number of pages)
\newpage
\lipsum[1-3]
\newpage

\lastpageref{pagesLTS.Roman} (\verb|\lastpageref{pagesLTS.Roman}|)
\textbf{does} refers to this page.\newline
Also \verb|\lastpageref{pagesLTS.Roman.3}|
give the right name of the page, but if \verb|pagecontinue=false| was chosen,
it might point to another page with the same name!
\lastpageref{pagesLTS.Roman.3}.\bigskip

\bigskip

There are \lastpages{Roman}{3}~pages (\verb|\lastpages{Roman}{3}|) in this
third Roman sector.

\newpage
\pagenumbering{gobble}
\section{gobble}

Well. You can use \verb|gobble| as page numbering scheme. It gobbles the page numbers,
i.\,e.~there is no page number. Maybe \verb|\thispagestyle{empty}| is more useful?
Do not expect to be able to make working references to pages without page number,
and do expect to get
\begin{verbatim}
pdfTeX warning (ext4): destination with the same identifier
(name{page.}) has been already used, duplicate ignored
<to be read again>
\end{verbatim}

The page (\verb|\thepage|): \thepage{} (There is a nothing between two spaces there.)\bigskip

Current page (\verb|\theCurrentPage|),
i.\,e.~counted continuously from the first page): \theCurrentPage \bigskip

CurrentPageLocal (\verb|\theCurrentPageLocal|),
i.\,e.~counted continuously from the first page of the
current page numbering scheme (if option pagecontinue=false: sector)): \theCurrentPageLocal \bigskip

There are \lastpageref*{pagesLTS.gobble.local} pages with gobbled numbers (i.\,e.~without numbers).
```

```
\bigskip

For testing purposes, try:

\begin{small}
\begin{verbatim}
\documentclass{article}
\usepackage[plainpages=false,pageanchor=true,pdfpagelabels=true]{hyperref}
\usepackage{pageslts}
\pagenumbering{gobble}
\begin{document}
First page.\newpage
\verb|\lastpageref{pagesLTS.0}              | \lastpageref{pagesLTS.0}              \par
\verb|\lastpageref{pagesLTS.0.local}        | \lastpageref{pagesLTS.0.local}        \par
\verb|\lastpageref{LastPage}                | \lastpageref{LastPage}                \par
\verb|\lastpageref{VeryLastPage}            | \lastpageref{VeryLastPage}            \par
\verb|\lastpageref{pagesLTS.gobble.1}       | \lastpageref{pagesLTS.gobble.1}       \par
\verb|\lastpageref{pagesLTS.gobble.1.local} | \lastpageref{pagesLTS.gobble.1.local} \par
\verb|\lastpageref{pagesLTS.gobble}         | \lastpageref{pagesLTS.gobble}         \par
\verb|\lastpageref{pagesLTS.gobble.local}   | \lastpageref{pagesLTS.gobble.local}   \par
\verb|\lastpageref{LastPages}               | \lastpageref{LastPages}               \par
\verb|\pagesLTSlastpage                     | \pagesLTSlastpage
\newpage
Last page.
\end{document}
\end{verbatim}
\end{small}
\newpage

\pagenumbering{Alph}
\section{Alph}

First page (\verb|\lastpageref{pagesLTS.0}|):
 \lastpageref{pagesLTS.0}\bigskip

The page (\verb|\thepage|): \thepage \bigskip

Current page (\verb|\theCurrentPage|),
i.\,e.~counted continuously from the first page): \theCurrentPage \bigskip

CurrentPageLocal (\verb|\theCurrentPageLocal|),
i.\,e.~counted continuously from the first page of the
current page numbering scheme (if option pagecontinue=false: sector)): \theCurrentPageLocal \bigskip

Last roman page (pagesLTS.roman): \lastpageref{pagesLTS.roman}{\hskip4em }
```

```
(There are \lastpageref{pagesLTS.roman.local} pages with roman numbers.)\bigskip

Last Roman page (pagesLTS.Roman): \lastpageref{pagesLTS.Roman}{\hskip3em }%
\makeatletter%
\ifpagesLTS@pagecontinue%
  (There are \lastpageref{pagesLTS.Roman.local}~pages with Roman numbers:\newline
\else%
  (There are %
  \lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
   \the\numexpr%
    \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
   \relax~pages with Roman numbers:\newline
\fi%
\makeatother%
\lastpages{Roman}{1}~pages in the first Roman sector
(\pageref{Roman}{\hskip3em }-\lastpageref{pagesLTS.Roman.1}{\hskip3em }),\newline
\lastpages{Roman}{2}~pages in the second Roman sector
(\pageref{Roman2}{\hskip3em }-\lastpageref{pagesLTS.Roman.2}{\hskip3em }), and\newline
\lastpages{Roman}{3}~pages in the third Roman sector
(\pageref{Roman3}{\hskip3em }-\lastpageref{pagesLTS.Roman.3}{\hskip3em }).)\bigskip

When the option \texttt{pagecontinue=false} is used with the
\textsf{pageslts} package, the \verb|\lastpageref{pagesLTS.Roman}| will
point to the same page as before, but this will have a lower number.
And if that page number was also present in a preceding sector,
the link will go to that page. (Several pages with the same number\ldots)\newline
The \verb|\lastpageref{pagesLTS.Roman.local}| will refer only to the number
of pages in the last sector. To get the total number of Roman pages, %
\begin{verbatim}
\the\numexpr \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
\relax%
\end{verbatim}%
would be necessary.\bigskip

Last arabic page (pagesLTS.arabic): \lastpageref{pagesLTS.arabic}{\hskip5em }
(There are only \lastpageref{pagesLTS.arabic.local} pages with arabic numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexampleArabic\verb|}| was used.)\bigskip

Last fnsymbol page (pagesLTS.fnsymbol): \lastpageref{pagesLTS.fnsymbol}\newline
(better \verb|\lastpageref{pagesLTS.fnsymbol}| than just\newline
\verb|\pageref{pagesLTS.fnsymbol}|!)\newline

```

```
(There are \lastpageref{pagesLTS.fnsymbol.local} pages with fnsymbol numbers.)\bigskip

Last alph page (pagesLTS.alph): \lastpageref{pagesLTS.alph}{\hskip4em }
(There are only \lastpageref{pagesLTS.alph.local} pages with alph numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexamplealph\verb|}| was used.)\bigskip

Last Alph page (pagesLTS.Alph): \lastpageref{pagesLTS.Alph}{\hskip4em }
(There are \lastpageref{pagesLTS.Alph.local} pages with Alph numbers.)\bigskip

Last page's \textit{name} (LastPage): \lastpageref{LastPage}\bigskip

Very last page's \textit{name} (VeryLastPage): \lastpageref{VeryLastPage}\newline
(\texttt{LastPage} and \texttt{VeryLastPage} are now identical, because both are now done
\verb|\AddToHook{enddocument/afterlastpage}| instead of formerly\newline
\verb|\AtEndDocument| and \verb|\AfterLastShipout|, which were different.)\bigskip

Last page's \textit{number} (LastPages): \lastpageref{LastPages}{\hskip3em }
(=~total number of pages)
\newpage

Without option \texttt{alphMulti=AB} of the \textsf{pageslts}
(and the help of the \textsf{alphalph} package), after page
\textquotedblleft Z\textquotedblright{} there would just appear~a
\begin{quote}
\begin{verbatim}
LaTeX Error: Counter too large
See the LaTeX manual or LaTeX Companion for explanation.
You've lost some text.  Try typing  <return>  to proceed.
If that doesn't work, type  X <return>  to quit.
\end{verbatim}
\end{quote}
Now the page numbers are continued AA, AB, AC,\ldots\ (AA, BB, CC,\ldots\ is
also possible, see the \textsf{pageslts} documentation).\newline
This is not demonstrated here, but see section~\ref{alph}.
\newpage

\section{silly}
The sillypage package, \url{https://ctan.org/pkg/sillypage}, provides the
page\-numbering scheme \verb|silly|, which is not demonstrated here
but works with the pageslts package, i.\,e.~the references
all show the right (page) number and link to the right page.\bigskip

This is \textbf{not} demonstrated here (discussing gobbled page numbers
is silly enough), but for testing purposes, try:

\begin{small}
```

```
\begin{verbatim}
\documentclass{article}
\usepackage[plainpages=false,pageanchor=true,pdfpagelabels=true]{hyperref}
\usepackage{pageslts}
\usepackage{sillypage}
\pagenumbering{silly}
\begin{document}
First page.\newpage
\verb|\lastpageref{pagesLTS.0}             | \lastpageref{pagesLTS.0}             \par
\verb|\lastpageref{pagesLTS.0.local}       | \lastpageref{pagesLTS.0.local}       \par
\verb|\lastpageref{LastPage}               | \lastpageref{LastPage}               \par
\verb|\lastpageref{VeryLastPage}           | \lastpageref{VeryLastPage}           \par
\verb|\lastpageref{pagesLTS.silly.1}       | \lastpageref{pagesLTS.silly.1}       \par
\verb|\lastpageref{pagesLTS.silly.1.local} | \lastpageref{pagesLTS.silly.1.local} \par
\verb|\lastpageref{pagesLTS.silly}         | \lastpageref{pagesLTS.silly}         \par
\verb|\lastpageref{pagesLTS.silly.local}   | \lastpageref{pagesLTS.silly.local}   \par
\verb|\lastpageref{LastPages}              | \lastpageref{LastPages}              \par
\verb|\pagesLTSlastpage                    | \pagesLTSlastpage
\newpage
Page \theCurrentPage\newpage
Page \theCurrentPage\newpage
Page \theCurrentPage\newpage
Page \theCurrentPage\newpage
Page \theCurrentPage\newpage
Page \theCurrentPage\newpage
Page \theCurrentPage\newpage
Page \theCurrentPage\newpage
Page \theCurrentPage\newpage
Page \theCurrentPage\newpage
Page \theCurrentPage\newpage
Page \theCurrentPage : Last page.
\end{document}
\end{verbatim}
\end{small}
\newpage

\section{The End}

First page (\verb|\lastpageref{pagesLTS.0}|):
 \lastpageref{pagesLTS.0}\bigskip

The page (\verb|\thepage|): \thepage \bigskip

Current page (\verb|\theCurrentPage|),
i.\,e.~counted continuously from the first page): \theCurrentPage \bigskip
```

```
CurrentPageLocal (\verb|\theCurrentPageLocal|),
i.\,e.~counted continuously from the first page of the
current page numbering scheme (if option pagecontinue=false: sector)): \theCurrentPageLocal \bigskip

Last roman page (pagesLTS.roman): \lastpageref{pagesLTS.roman}{\hskip4em }
(There are \lastpageref{pagesLTS.roman.local} pages with roman numbers.)\bigskip

Last Roman page (pagesLTS.Roman): \lastpageref{pagesLTS.Roman}{\hskip3em }%
\makeatletter%
\ifpagesLTS@pagecontinue%
  (There are \lastpageref{pagesLTS.Roman.local}~pages with Roman numbers:\newline
\else%
  (There are %
  \lastpages{Roman}{1}+\lastpages{Roman}{2}+\lastpages{Roman}{3}=%
   \the\numexpr%
    \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
   +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
   \relax~pages with Roman numbers:\newline
\fi%
\makeatother%
\lastpages{Roman}{1}~pages in the first Roman sector
(\pageref{Roman}{\hskip3em }-\lastpageref{pagesLTS.Roman.1}{\hskip3em }),\newline
\lastpages{Roman}{2}~pages in the second Roman sector
(\pageref{Roman2}{\hskip3em }-\lastpageref{pagesLTS.Roman.2}{\hskip3em }), and\newline
\lastpages{Roman}{3}~pages in the third Roman sector
(\pageref{Roman3}{\hskip3em }-\lastpageref{pagesLTS.Roman.3}{\hskip3em }).)\bigskip

When the option \texttt{pagecontinue=false} is used with the
\textsf{pageslts} package, the \verb|\lastpageref{pagesLTS.Roman}| will
point to the same page as before, but this will have a lower number.
And if that page number was also present in a preceding sector,
the link will go to that page. (Several pages with the same number\ldots)\newline
The \verb|\lastpageref{pagesLTS.Roman.local}| will refer only to the number
of pages in the last sector. To get the total number of Roman pages, %
\begin{verbatim}
\the\numexpr \csname c@pagesLTS.Roman.1.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.2.local.cnt\endcsname%
 +\csname c@pagesLTS.Roman.3.local.cnt\endcsname%
\relax%
\end{verbatim}%
would be necessary.\bigskip

Last arabic page (pagesLTS.arabic): \lastpageref{pagesLTS.arabic}{\hskip5em }
(There are only \lastpageref{pagesLTS.arabic.local} pages with arabic numbers,
```

```
because an \verb|\addtocounter{page}{|\pagesLTSexampleArabic\verb|}| was used.)\bigskip

Last fnsymbol page (pagesLTS.fnsymbol): \lastpageref{pagesLTS.fnsymbol}\newline
(better \verb|\lastpageref{pagesLTS.fnsymbol}| than just\newline
\verb|\pageref{pagesLTS.fnsymbol}|!)\bigskip

(There are \lastpageref{pagesLTS.fnsymbol.local} pages with fnsymbol numbers.)\bigskip

Last alph page (pagesLTS.alph): \lastpageref{pagesLTS.alph}{\hskip4em }
(There are only \lastpageref{pagesLTS.alph.local} pages with alph numbers,
because an \verb|\addtocounter{page}{|\pagesLTSexamplealph\verb|}| was used.)\bigskip

Last gobble page (pagesLTS.gobble: \lastpageref{pagesLTS.gobble}{\hskip4em }
Yes, there is neither page number nor link, because everything got gobbled.
(There is \lastpageref*{pagesLTS.gobble.local} page with gobbled number (i.\,e.~without number).)\bigskip

Last Alph page (pagesLTS.Alph): \lastpageref{pagesLTS.Alph}{\hskip4em }
(There are \lastpageref{pagesLTS.Alph.local} pages with Alph numbers.)\bigskip

Silly page numbers are possible but were not demonstrated in this example.
\newpage

Last page's \textit{name} (LastPage): \lastpageref{LastPage}\bigskip

Very last page's \textit{name} (VeryLastPage): \lastpageref{VeryLastPage}\newline
(\texttt{LastPage} and \texttt{VeryLastPage} are now identical, because both are now done
\verb|\AddToHook{enddocument/afterlastpage}| instead of formerly\newline
\verb|\AtEndDocument| and \verb|\AfterLastShipout|, which were different.)\bigskip

Last page's \textit{number} (LastPages): \lastpageref{LastPages}{\hskip3em }
(=~total number of pages)
\end{document}
```
⟨/example⟩

6 The implementation

(This and the source code of the example file are the reasons for printing the documentation in landscape format instead of portrait.)

We start off by checking that we are loading into LaTeX 2_ε, format version at least 2024-11-01, and announcing the name and version of this package.

⟨*package⟩

```
\NeedsTeXFormat{LaTeX2e}[2024-11-01]
\ProvidesPackage{pageslts}[2025-08-14 v2.0g Refers to special pages' numbers/names (HMM)]

```

A short description of the pageslts package:

```
%% Allows for things like\par
%% |Page \thepage\ (\theCurrentPage; local: \theCurrentPageLocal) of %
%%  \lastpageref{pagesLTS.roman}(\lastpageref{pagesLTS.roman.local}) + %
%%  \lastpageref{pagesLTS.Roman}(\lastpageref{pagesLTS.Roman.local}) + %
%%  \lastpageref{pagesLTS.arabic}(\lastpageref{pagesLTS.arabic.local}) + %
%%  \lastpageref{pagesLTS.fnsymbol}(\lastpageref{pagesLTS.fnsymbol.local}) + %
%%  \lastpageref{pagesLTS.alph}(\lastpageref{pagesLTS.alph.local}) + %
%%  \lastpageref{pagesLTS.Alph}(\lastpageref{pagesLTS.Alph.local}) = %
%%  \lastpageref{LastPages} pages.|\par
%% to get\par
%% 'Page d (57; local: 4) of ii(2) + XX(20) + *(1) + 30(30) + e(5) + C(3) = 61 pages.'.

```

For the handling of the options we use the kernel commands:

```
\DeclareKeys[pagesLTS]
 {alphMult      .store = \pagesLTS@alphMult ,
  alphMult      .usage = preamble ,
  AlphMulti     .store = \pagesLTS@AlphMulti ,
  AlphMulti     .usage = preamble ,
  pagecontinue  .if    = pagesLTS@pagecontinue ,
  pagecontinue  .usage = load ,
  romanMult     .if    = pagesLTS@romanMult ,
  romanMult     .usage = load ,
  RomanMulti    .if    = pagesLTS@RomanMulti ,
  RomanMulti    .usage = load ,
  fnsymbolmult  .if    = pagesLTS@fnsymbolmult ,
  fnsymbolmult  .usage = load
 }

\SetKeys[pagesLTS]{alphMult=ab,AlphMulti=AB,pagecontinue=true,romanMult=true,RomanMulti=true,fnsymbolmult=true}

\DeclareUnknownKeyHandler[pagesLTS]{%
  \PackageError{pageslts}{Unknown option "\CurrentOption"}{%
```

```
    LaTeX has been asked to set an option called\MessageBreak%
    '#1' \if\relax\detokenize{#2}\relax\else with value '#2'\fi\MessageBreak%
    but the package "pageslts" has not created an option with this name.\MessageBreak%
    Try typing  <return>  to proceed.\MessageBreak%
    If that doesn't work, type  X <return>  to quit.%
   }%
  }

\ProcessKeyOptions[pagesLTS]

```

This package really does not work with an old LaTeX format without hooks, or without ε-TeX, or with a pagesLTS.sty file, or with version 2.0 from 1992 of the endfloats package, or hyperref from 2015 and so on. Assuming that nobody would combine those old things with a modern LaTeX format, we request a recent format and do not check for the rest.

```
\@ifl@t@r\fmtversion{2024-11-01}{}{\PackageError{pageslts}{%
  LaTeX format 2024-11-01 or newer needed}{%
  Needed LaTeX format version: 2024-11-01 or newer.\MessageBreak%
  Found\space\space LaTeX format version: \fmtversion.\MessageBreak%
  Please update your TeX distribution!\MessageBreak%
  Loading of pageslts package is aborted.}
  \expandafter\endinput}

```

The undolabl package (see subsection 7.1) is needed to overwrite labels, when the same page numbering scheme is used twice (or even more often).

```
\RequirePackage{undolabl}[2024-12-05]
\IfPackageLoadedTF{undolabl}{\IfPackageAtLeastF{undolabl}{2024-12-05}{%
   \PackageError{pageslts}{undolabl package version too old}{%
     Required version: 2024-12-05 v1.0n or newer,\MessageBreak%
     found version:\space\space\space\space\csname ver@undolabl.sty\endcsname.%
}}}{\PackageError{pageslts}{undolabl package not loaded}{%
     The undolabl package is required by pageslts but was not loaded.}%
   }

```

For comparisons, zero, one, two and three are defined (\z@, \@ne and so on do not work for this) and for the page numbering scheme handling some definitions are made.

```
\def\pagesLTS@zero{0}
\def\pagesLTS@one{1}
\def\pagesLTS@two{2}
\def\pagesLTS@three{3}
\def\pagesLTS@ab{ab}
\def\pagesLTS@bb{bb}
\def\pagesLTS@ABi{AB}
\def\pagesLTS@BBi{BB}

```

The traditional behaviour is a reset of the page number to one, each time the page numbering scheme changes. The option `pagecontinue` changes this to a continuation with the number/name following the last page number/name of the same page numbering scheme. The user is informed accordingly.

```
\ifpagesLTS@pagecontinue
  \PackageNoteNoLine{pageslts}{Option pagecontinue enabled (maybe by default):\MessageBreak%
    The pageslts package will continue the page numbering,\MessageBreak%
    when the same page numbering scheme is used twice.\MessageBreak%
    If you do not want this, call pageslts with option\MessageBreak%
    pagecontinue=false (or use \string\setcounter{page}=1)}
\else
  \PackageNoteNoLine{pageslts}{Option pagecontinue is false:\MessageBreak%
    The pageslts package was used, but the option\MessageBreak%
    pagecontinue was set to false.\MessageBreak%
    If you want the page numbers to be continued,\MessageBreak%
    when the same page numbering scheme is used twice,\MessageBreak%
    please call pageslts with option pagecontinue=true,\MessageBreak%
    otherwise the page number is reset to one each time\MessageBreak%
    the page numbering scheme is changed}
\fi

```

The page number printed in `alph` or in `Alph` page numbering scheme had to be > 0 and < 27. Now the alphalph package allows to extend the numbering scheme (not only for pages). Because some users prefer aa, ab, ac, ad,... and some aa, bb, cc, dd,..., both schemes can be chosen via the options. The `fnsymbol` page numbering scheme was restricted to values > 0 and < 10. The alphalph package allows to extend this page numbering scheme, too. Option `fnsymbolmult` can be chosen with the pageslts package. If no extension is wished (or another extension is wished and implemented manually), pageslts can be called with options set to 0 (zero) and false: `alphMult=0`, `AlphMulti=0`, `fnsymbolmult=false`.

```
\ifx\pagesLTS@alphMult\pagesLTS@ab%
\else
  \ifx\pagesLTS@alphMult\pagesLTS@bb%
  \else
    \ifx\pagesLTS@alphMult\pagesLTS@zero%
      \PackageNoteNoLine{pageslts}{%
        Option alphMult=0 found:\MessageBreak%
        The pageslts package was used, but the option\MessageBreak%
        alphMult was set to 0 (zero).\MessageBreak%
        If you want the page numbers to be extended\MessageBreak%
        after z, you have to organize this yourself now.\MessageBreak%
        For automatic continuation, please use the\MessageBreak%
        alphalph package and call pageslts\MessageBreak%
        with option alphMult=ab (for aa, ab, ac, ad,...) or\MessageBreak%
        with option alphMult=bb (for aa, bb, cc, dd,...)}
    \else
      \PackageError{pageslts}{Unknown option value}{%
        The pageslts package was used with option\MessageBreak%
```

```
        alphMult=\meaning\pagesLTS@alphMult .\MessageBreak%
        Only values 'ab', 'bb', and '0' (zero) are valid.\MessageBreak%
        Now the default 'ab' is being set.}
      \SetKeys[pagesLTS]{alphMult=ab}
    \fi
  \fi
\fi

\ifx\pagesLTS@AlphMulti\pagesLTS@ABi%
\else
  \ifx\pagesLTS@AlphMulti\pagesLTS@BBi%
  \else
    \ifx\pagesLTS@AlphMulti\pagesLTS@zero%
      \PackageNoteNoLine{pageslts}{%
        Option AlphMulti=0 found:\MessageBreak%
        The pageslts package was used, but the option\MessageBreak%
        AlphMulti was set to 0 (zero).\MessageBreak%
        If you want the page numbers to be extended\MessageBreak%
        after Z, you have to organize this yourself now.\MessageBreak%
        For automatic continuation, please use the\MessageBreak%
        alphalph package and call pageslts\MessageBreak%
        with option AlphMulti=AB (for AA, AB, AC, AD,...) or\MessageBreak%
        with option AlphMulti=BB (for AA, BB, CC, DD,...)}
    \else
      \PackageError{pageslts}{Unknown option value}{%
        The pageslts package was used with option\MessageBreak%
        AlphMulti=\meaning\pagesLTS@AlphMulti .\MessageBreak%
        Only values 'AB', 'BB', and '0' (zero) are valid.\MessageBreak%
        The default 'AB' is set.}
      \SetKeys[pagesLTS]{AlphMulti=AB}
    \fi
  \fi
\fi

```

If `alph` or `Alph` or `fnsymbol` page numbers shall be continued, the alphalph package is required.

```
\ifx\pagesLTS@alphMult\pagesLTS@zero\else\RequirePackage{alphalph}[2019/12/09]\fi
\ifx\pagesLTS@AlphMulti\pagesLTS@zero\else\RequirePackage{alphalph}[2019/12/09]\fi
\ifpagesLTS@fnsymbolmult\RequirePackage{alphalph}[2019/12/09]\fi

```

For the roman page numbering scheme, it is just the choice of an extension by pageslts or not.

```
\ifpagesLTS@romanMult
  \PackageNoteNoLine{pageslts}{Option romanMult enabled (maybe by default):\MessageBreak%
    The pageslts package will extend the page numbering\MessageBreak%
    of the roman scheme below i with\MessageBreak%
     0, -i, -ii, -iii, -iv,...\MessageBreak%
```

```
    If you do not want this, call pageslts with option\MessageBreak%
    romanMult=false}
\else
  \PackageNoteNoLine{pageslts}{Option romanMult is set to false:\MessageBreak%
    The pageslts package was used, but the option\MessageBreak%
    romanMult was set to false.\MessageBreak%
    If you want the page numbering of the roman scheme\MessageBreak%
    to be extended below i,\MessageBreak%
    please call pageslts with option romanMult=true,\MessageBreak%
    otherwise zero and negative page numbers of the\MessageBreak%
    roman scheme will need to be defined otherwise}
\fi

```

Same for the Roman page numbering scheme.

```
\ifpagesLTS@RomanMulti
  \PackageNoteNoLine{pageslts}{Option RomanMulti enabled (maybe by default):\MessageBreak%
    The pageslts package will extend the page numbering\MessageBreak%
    of the Roman scheme below I with\MessageBreak%
    0, -I, -II, -III, -IV,...\MessageBreak%
    If you do not want this, call pageslts with option\MessageBreak%
    RomanMulti=false}
\else
  \PackageNoteNoLine{pageslts}{Option RomanMulti is set to false:\MessageBreak%
    The pageslts package was used, but the option\MessageBreak%
    RomanMulti was set to false.\MessageBreak%
    If you want the page numbering of the Roman scheme\MessageBreak%
    to be extended below i,\MessageBreak%
    please call pageslts with option RomanMulti=true,\MessageBreak%
    otherwise zero and negative page numbers of the\MessageBreak%
    Roman scheme will need to be defined otherwise}
\fi

```

For the footnotesymbol page numbering scheme, it is also just the choice of a extension by pageslts or not.

```
\ifpagesLTS@fnsymbolmult
  \PackageNoteNoLine{pageslts}{Option fnsymbolmult enabled (maybe by default):\MessageBreak%
    The pageslts package will extend the page numbering\MessageBreak%
    of the footnotesymbol scheme using the alphalph\MessageBreak%
    package.\MessageBreak%
    If you do not want this, call pageslts with option\MessageBreak%
    fnsymbolmult=false}
\else
  \PackageNoteNoLine{pageslts}{%
    Option fnsymbolmult is set to false:\MessageBreak%
    The pageslts package was used, but the option\MessageBreak%
    fnsymbolmult was set to false.\MessageBreak%
```

```
    If you want the page numbering of the footnotesymbol\MessageBreak%
    scheme to be extended using the alphalph package,\MessageBreak%
    please call pageslts with option fnsymbolmult=true,\MessageBreak%
    otherwise page numbers of the footnotesymbol scheme\MessageBreak%
    greater than nine will need to be defined otherwise}
\fi

```

Now defining some variables, place-holders, and abbreviations:

```
\def\pagesLTS@pnc{0}
\def\pagesLTS@called{0}
\def\pagesLTS@fns{fnsymbol}
\def\pagesLTS@alph{alph}
\def\pagesLTS@Alph{Alph}
\def\pagesLTS@gobble{gobble}
\def\pagesLTS@silly{silly}
\def\pagesLTS@rerun{0}
\def\pagesLTS@esoFs{0}
\def\pagesLTS@esovFs{0}
\def\pagesLTS@esoSy{0}
\def\pagesLTS@esovSy{0}
\def\pagesLTS@esoGbl{0}
\def\pagesLTS@esovGbl{0}
\def\lastpageref{\lastpagereftxt}
\def\pagesLTS@undolable{none}
\def\pncmissing{0}
\def\pagesLTS@messageNPN{%
 The pageslts package was used, but\MessageBreak%
 \string\pagenumbering \MessageBreak%
 was not called at the beginning of the document\MessageBreak%
 (maybe earlier or later).\MessageBreak%
 Please use \string\pagenumbering \MessageBreak%
 (with an argument like arabic, roman, Roman,\MessageBreak%
 fnsymbol, alph, Alph, gobble,\MessageBreak%
 or silly (requiring sillypage package))\MessageBreak%
 at the beginning of your document!\MessageBreak%
 Otherwise your document will probably compile,\MessageBreak%
 but the pageslts package might not be able\MessageBreak%
 to get all labels for the references\MessageBreak%
 to the respective pages right.\MessageBreak%
 }

```

Defining some new counters (and doing related things):

```
\newcounter{CurrentPage}
\setcounter{CurrentPage}{1}
\def\theCurrentPageLocal{\number\value{pagesLTS.current.local.\pagesLTS@pnc}}
```

The counter `pagesLTS.pagenr` is for saving into the `.aux` file the total page number of the last page.

```
\newcounter{pagesLTS.pagenr}
\setcounter{pagesLTS.pagenr}{1}
```

While generally `\pagesLTS@providecounter{pagesLTS.current.local.\pagesLTS@pnc}` is used, for the beginning of the document `pagesLTS.current.local.0` is predefined. (A `\pagesLTS@providecounter{pagesLTS.current.local.\pagesLTS@pnc}` could be used for this, too, but we know that `pagesLTS.current.local.0` was not defined, so we can just do the definition here.) And the first local page gets the number one.

```
\newcounter{pagesLTS.current.local.0}
\setcounter{pagesLTS.current.local.0}{1}
```

And the same again for `pagesLTS.pnc.0`.

```
\newcounter{pagesLTS.pnc.0}
```

Also a scratch counter is defined, `\pagesLTSlastpage` and `\pagesLTSlastpageHy` are predefined (e. g. for computations in the first compilation run), and the obsolete `\pagesLTS.lastpage` points to the new `\pagesLTSlastpage`.

```
\newcounter{pagesLTS.tmpcounter}
\def\pagesLTSlastpage{1}
\def\pagesLTSlastpageHy{1}
\def\pagesLTS.lastpage{\pagesLTSlastpage}
```

`\xroman` When `\roman{...}` is used with a value < 1, LaTeX just ignores this (see subsection 3.5). Here we provide a command `\xroman{...}` (expanded roman), which gives the usual `\roman` numbers (i, ii, iii, iv,...) for positive values, $-|\ldots|$ (i. e. -i, -ii, -iii, -iv,...) for negative values, and 0 for all other values (which should be zero).

```
\newcommand{\xroman}[1]{%
  \ifnum\value{#1}>0%
    \roman{#1}%
  \else%
    \ifnum\value{#1}<0%
```

`\number\value{#1}` gives the arabic number of argument `#1`, which is negative here (for example -7), "`-`" puts another minus sign in front of it (for example $--7$), `\number` removes all unnecessary preceding zeros, plus and minus signs (for example 7), `\romannumeral` turns it into a `roman` number (for example vii), and "`-`" puts the minus sign back in front of it (for example -vii).

```
      -\romannumeral\number-\number\value{#1}%
    \else%
      0%
    \fi%
  \fi%
  }
```

\XRoman

\XRoman does the same for uppercase \Roman numbers. -\uppercase{\romannumeral\number-\number\value{#1}} cannot be used, because the result in the example is -\uppercase{vii} and not -VII.[4] Therefore we have a look at LaTeX's own \@Roman\FOOcounter, \def\@Roman#1{\expandafter\@slowromancap\romannumeral #1@}, and use \@slowromancap, which is a fully expandable macro, to do the trick for this:

> "\def\@slowromancap#1{\ifx @#1% then terminate
> \else
> \if i#1I\else\if v#1V\else\if x#1X\else\if l#1L\else\if
> c#1C\else\if d#1D\else \if m#1M\else#1\fi\fi\fi\fi\fi\fi\fi
> \expandafter\@slowromancap
> \fi
> }" (2021/11/08 Version v1.1n LaTeX Kernel File 21: ltcounts.dtx Counters and Lengths).

```
\newcommand{\XRoman}[1]{%
  \ifnum\value{#1}>0%
    \Roman{#1}%
  \else%
    \ifnum\value{#1}<0%
      -\expandafter\@slowromancap\romannumeral\number-\number\value{#1}@%
    \else%
      0%
    \fi%
  \fi%
  }
```

\pagesLTS@providecounter

We provide a way to create counters like

pagesLTS.pnc. page numbering scheme

- pagesLTS.pnc.<page numbering scheme>, e. g. pagesLTS.pnc.Roman,

pagesLTS.double. page numbering scheme

- pagesLTS.double.<page numbering scheme>, e. g. pagesLTS.double.Roman,

PageCurrentLocal. page numbering scheme

- PageCurrentLocal.<page numbering scheme>, e. g. PageCurrentLocal.Roman,

for all page numbering schemes, even those not supported by the current original \pagenumbering (2020/12/05 Version v1.1a LaTeX Kernel File 33 ltpageno.dtx Page Numbering), which is

```
\countdef\c@page=0 \c@page=1
\def\cl@page{}
\def\pagenumbering#1{%
  \global\c@page \@ne \gdef\thepage{\csname @#1\endcsname
   \c@page}}
```

```
\newcommand{\pagesLTS@providecounter}[1]{\@ifundefined{c@#1}{\newcounter{#1}}{}}
```

[4] This does not matter for the print out, but for the display of the logical page numbers as well as the .aux file.

The command \pagesLTS@ifcounter is obsolete, but if a document was compiled using an older version of the pageslts package, the command will still be in the .aux file. (Following code will be removed in some future version.)

```
\NewCommandCopy{\pagesLTS@ifcounter}{\pagesLTS@providecounter}
\AddToHook{begindocument}{%
```

This hook is executed after the .aux file has been read, so there should be no following \pagesLTS@ifcounter.

```
  \AddToHookWithArguments{cmd/pagesLTS@ifcounter/before}{%
    \PackageError{pageslts}{Old command \string\pagesLTS@ifcounter\space used}{%
      Replace by \string\pagesLTS@providecounter.}%
    }%
  }

```

\lastpages We provide a command to give the number of pages in a sector of a split page numbering scheme (see page 9, pagesLTS.<page numbering scheme>.<number>.local.cnt):

```
\newcommand{\lastpages}[2]{%
  \pagesLTS@providecounter{pagesLTS.#1.#2.local.cnt}%
  \number\value{pagesLTS.#1.#2.local.cnt}%
  }

```

\pagesLTS@writelabel At last defining the writing of a label:

```
\newcommand{\pagesLTS@writelabel}[1]{%
  \addtocounter{page}{+1}%
```

\addtocounter{page}{+1} because \pagesLTS@putlabel includes an \addtocounter{page}{-1}, which is not necessary here.
Into the .aux file something like
\newlabel{pagesLTS.Roman}{{}{VIII}{}{page.VIII}{}}
is written, thus \lastpageref{pagesLTS.Roman} prints VIII and links to page.VIII.

```
  \pagesLTS@putlabel{pagesLTS.#1}{\thepage}{1}%
```

\addtocounter{page}{-1} is skipped, because another \pagesLTS@putlabel follows.

```
  \ifx\pagesLTS@pnc\pagesLTS@zero%
```

i. e. if the current page numbering scheme is "0", i. e. before the first \pagenumbering{...} command, do nothing,

```
  \else%
```

otherwise write into the .aux file something like
\newlabel{pagesLTS.arabic.local}{{}{6}{}{page.9}{}},
thus \lastpageref{pagesLTS.arabic.local} prints 6 and links to page.9.

\addtocounter{page}{+1} is skipped, because we skipped the \addtocounter{page}{-1} above.

```
    \pagesLTS@putlabel{pagesLTS.#1.local}{\theCurrentPageLocal}{1}%
```

\addtocounter{page}{-1} immediately after the \pagesLTS@putlabel is skipped, because we do that outside of the \ifx \else \fi.

```
  \fi%
  \addtocounter{page}{-1}%
  }
```

\erroralphalph

\erroralphalph extends the "numbers" of counters to zero and negative values for representations usually not supporting this:
\alphalph, \AlphAlph, and \fnsymbolmult of the alphalph package.
\alph, \Alph, and \fnsymbol would not support "numbers" below one.
\arabic already supports negative numbers and zero.
\roman and \Roman support neither negative numbers nor zero, but are expanded in this package (\xroman and \XRoman), see page 51.

```
%% The following code is from Heiko Oberdiek [2010/04/18],                      %%
%% expanding his alphalph package as of 2010/04/18, v2.3. (Thanks!)             %%
\newcommand*{\erroralphalph}[2]{%
  \ifnum\value{#2}>0%
    #1{\value{#2}}%
  \else%
    \ifnum\value{#2}<0%
      -#1{\expandafter\@gobble\the\value{#2}}%
    \else%
      0%
    \fi%
  \fi%
  }
%% End of code from Heiko Oberdiek                                             %%
%% Check and Error/Warning messages have been moved to hook "shipout/foreground" %%
%% because messages inside e.g. the \pageref command can cause trouble.        %%
```

\expandPagenumbering Here the \erroralphalph command is called with the appropriate arguments for each page numbering scheme.

```
\newcommand{\expandPagenumbering}[1]{%
  \let\Origthepage\thepage%
  \def\pagesLTS@tmpC{arabic}%
  \ifx\pagesLTS@pnc\pagesLTS@tmpC%
```

\arabic already supports negative numbers and zero (-MAX...MAX, where MAX = 2 147 483 647).

```
  \else%
    \def\pagesLTS@tmpC{roman}%
    \ifx\pagesLTS@pnc\pagesLTS@tmpC%
      \ifpagesLTS@romanMult%
```

\erroralphalph{\roman}{page} cannot be used, because -\roman{\expandafter\@gobble\the\value{page}} does not work. If option romanMult is not false, \xroman (see page 51) expands the usable roman page numbers to values below 1 (i, I, respectively), see subsubsection 2.1.3.

```
        \renewcommand*{\thepage}{\xroman{page}}%
      \fi%
    \else%
      \def\pagesLTS@tmpC{Roman}%
      \ifx\pagesLTS@pnc\pagesLTS@tmpC%
```

The same for \Roman page numbering, expanded by \XRoman (see page 51). When the romanbar package has been loaded, the page format is changed accordingly.

```
        \IfPackageLoadedTF{romanbarpagenumber}{%
          \ifpagesLTS@RomanMulti%
            \renewcommand*{\thepage}{\Roman@bar{\XRoman{page}}}%
          \else%
            \renewcommand*{\thepage}{\Roman@bar{\Roman{page}}}%
          \fi%
        }{\ifpagesLTS@RomanMulti%
            \renewcommand*{\thepage}{\XRoman{page}}%
          \fi%
        }%
      \else%
        \ifx\pagesLTS@pnc\pagesLTS@alph%
```

\alph and \Alph page numberings are expanded to negative and zero values, and to values greater than “z” or “Z” with the alphalph package. – If \pagesLTS@alphMult was zero, nothing is done.

```
          \ifx\pagesLTS@alphMult\pagesLTS@ab%
            \renewcommand*{\thepage}{\erroralphalph{\alphalph}{page}}%
          \else%
            \ifx\pagesLTS@alphMult\pagesLTS@bb%
              \renewcommand*{\thepage}{\erroralphalph{\alphMult}{page}}%
            \fi%
          \fi%
```

```
      \else%
        \ifx\pagesLTS@pnc\pagesLTS@Alph%
          \ifx\pagesLTS@AlphMulti\pagesLTS@ABi%
            \renewcommand*{\thepage}{\erroralphalph{\AlphAlph}{page}}%
          \else%
            \ifx\pagesLTS@AlphMulti\pagesLTS@BBi%
              \renewcommand*{\thepage}{\erroralphalph{\AlphMult}{page}}%
            \fi%
          \fi%
        \else%
          \ifx\pagesLTS@pnc\pagesLTS@fns%
```

Same for `\fnsymbol` page numbers.

```
            \ifpagesLTS@fnsymbolmult%
              \renewcommand*{\thepage}{\erroralphalph{\fnsymbolmult}{page}}%
            \fi%
          \else%
```

For `silly` page numbers (from the sillypage package), `\thepage` is `\csname @silly\endcsname \c@page`, i. e. `\@silly \c@page` – no change needed.

```
            \ifx\pagesLTS@pnc\pagesLTS@silly%
            \else%
```

For `gobble` page numbers, `\thepage` is `\csname @gobble\endcsname \c@page`, i. e. `\@gobble \c@page` and `\@gobble` is `\long\def \@gobble #1{}`, i. e. `\thepage` is `{}`. This is OK, but hyperref does not create links when there is no linktext. Thus we give `\null` as text for the page number, which prints as nothing but makes hyperref happy – although not happy enough yet: `\null` expands to `\hbox {}`, and hyperref complains

```
Package hyperref Warning: Token not allowed in a PDF string (Unicode):
(hyperref)                removing `\hbox' on input line <something>.

Package hyperref Warning: Token not allowed in a PDF string (PDFDocEncoding):
(hyperref)                removing `\hbox' on input line <something>.
```

To make hyperref extra happy, we tell it to not use `\null` as PDF string.

```
              \ifx\pagesLTS@pnc\pagesLTS@gobble%
                \IfPackageLoadedT{hyperref}{%
                  \renewcommand*{\thepage}{\texorpdfstring{\null}{{}}%
                    \csname @gobble\endcsname \c@page}}%
              \else%
```

If the used page numbering scheme has not been recognized by the pageslts package so far, we can do nothing, and problems might result.

```
                \PackageError{pageslts}{unknown page numbering scheme}{%
                  The pageslts package encountered the unknown\MessageBreak%
                  page numbering scheme\MessageBreak%
                  \meaning#1. \MessageBreak%
```

```
                    If this is no typing mistake, it might work\MessageBreak%
                    - or it might not work.\MessageBreak%
                    \@ehc}%
  \fi\fi\fi\fi\fi\fi\fi\fi%
  \let\pagesLTS@tmpC\undefined%
  }
```

`\pagenumbering` To keep the original meaning of `\pagenumbering`:

```
\NewCommandCopy{\OrigPagenumbering}{\pagenumbering}
```

Now for the **new** version of the `\pagenumbering` command:

```
\renewcommand{\pagenumbering}[1]{%
```

If the current page numbering scheme, `\pagesLTS@pnc`, and the requested page numbering scheme, `#1`, is the same one, nothing is done, otherwise the real action begins.

```
  \edef\pagesLTS@tmpA{#1}%
  \ifx\pagesLTS@pnc\pagesLTS@tmpA%
  \else%
```

If the requested page numbering scheme, `#1`, is `\pagesLTS@fns`, i. e. `fnsymbol`, the counter`pagesLTS.fnsymbol.local` is needed. If it does not exists yet, it is created here. Same for the `silly` and `gobble` page numbering schemes.

```
    \ifx\pagesLTS@tmpA\pagesLTS@fns%
      \pagesLTS@providecounter{pagesLTS.fnsymbol.local}%
    \fi%
    \ifx\pagesLTS@tmpA\pagesLTS@silly%
      \IfPackageLoadedF{sillypage}{\PackageError{pageslts}{%
        silly pagenumbering requested, sillypage package not loaded}{%
        When using \string\pagenumbering{silly},\MessageBreak%
        load the sillypage package.}}%
      \pagesLTS@providecounter{pagesLTS.silly.local}%
    \fi%
    \ifx\pagesLTS@tmpA\pagesLTS@gobble%
      \pagesLTS@providecounter{pagesLTS.gobble.local}%
    \fi%
```

The next code is executed, when we are at a page after the first one. This distinction is done for two reasons:
On the one hand, `\pagenumbering` could be called *before* `\begin{document}` (where the current page should not be greater than one), and on the other hand we go one page back to aim all references to that page. Obviously at the first page there is no going backward.

```
    \ifnum\value{CurrentPage}>1%
      \addtocounter{page}{-1}%
      \addtocounter{pagesLTS.current.local.\pagesLTS@pnc}{-1}%
```

For the case that the page numbering scheme is or will be split, like e. g. the Roman one in the `pageslts-example.tex`, a counter like `pagesLTS.Roman.1.local.count` (or `pagesLTS.Roman.2.local.count`, `pagesLTS.Roman.3.local.count`,. . .) is introduced and set to the number of the local page.

```
    \newcounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}%
    \setcounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}{%
      \value{pagesLTS.current.local.\pagesLTS@pnc}}%
```

If the page numbering scheme is `fnsymbol`, and if it *was* used before, from said counter the number of pages of the preceding uses of the same page numbering scheme, `pagesLTS.\pagesLTS@pnc.done`, is subtracted (same as for the other schemes, see below). The same is done for the `silly` and `gobble` page numbers.

```
    \ifx\pagesLTS@pnc\pagesLTS@fns%
      \ifnum\value{pagesLTS.pnc.\pagesLTS@pnc}>1%
        \setcounter{pagesLTS.tmpcounter}{\numexpr\value{pagesLTS.pnc.\pagesLTS@pnc}-1\relax}%
        \addtocounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}{%
          -\value{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.tmpcounter}.local.count}}%
      \fi%
    \else%
      \ifx\pagesLTS@pnc\pagesLTS@silly%
        \ifnum\value{pagesLTS.pnc.\pagesLTS@pnc}>1%
          \setcounter{pagesLTS.tmpcounter}{\numexpr\value{pagesLTS.pnc.\pagesLTS@pnc}-1\relax}%
          \addtocounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}{%
            -\value{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.tmpcounter}.local.count}}%
        \fi%
      \else%
        \ifx\pagesLTS@pnc\pagesLTS@gobble%
          \ifnum\value{pagesLTS.pnc.\pagesLTS@pnc}>1%
            \setcounter{pagesLTS.tmpcounter}{\numexpr\value{pagesLTS.pnc.\pagesLTS@pnc}-1\relax}%
            \addtocounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}{%
              -\value{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.tmpcounter}.local.count}}%
          \fi%
        \else%
```

If the page numbering scheme is *neither* `fnsymbol` *nor* `silly` *nor* `gobble` and if the page numbering scheme was *not* used before, an unnumbered label is also written:

```
          \ifnum\value{pagesLTS.pnc.\pagesLTS@pnc}<2%
            \pagesLTS@writelabel{\pagesLTS@pnc}%
          \fi%
        \fi%
      \fi%
```

For all page numbering schemes: If the page numbering scheme *was* used before, from said counter the number of pages of the preceding uses of the same page numbering scheme, `pagesLTS.\pagesLTS@pnc.done`, is subtracted – except if option `pagenumbering=false` has been chosen, because then the pagenumbers are reset anyway.

```
    \ifnum\value{pagesLTS.pnc.\pagesLTS@pnc}>1%
      \ifpagesLTS@pagecontinue%
        \setcounter{pagesLTS.tmpcounter}{\numexpr\value{pagesLTS.pnc.\pagesLTS@pnc}-1\relax}%
        \pagesLTS@providecounter{pagesLTS.\pagesLTS@pnc.done}%
        \addtocounter{pagesLTS.\pagesLTS@pnc.done}{%
           \value{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.tmpcounter}.local.count}}%
```

```
            \addtocounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}{%
               -\value{pagesLTS.\pagesLTS@pnc.done}}%
          \fi%
        \fi%
```

A numbered label is written:

```
        \pagesLTS@writelabel{\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}}%
```

The values are written to the `.aux` file (if writing is allowed: `\if@filesw`), because they must be available at the beginning of the document:

```
        \if@filesw%
          \immediate\write\@auxout{\string
            \pagesLTS@providecounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.cnt}}%
          \edef\pagesLTS@tmpB{\number\value{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}}%
          \immediate\write\@auxout{\string
            \setcounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.cnt}{\pagesLTS@tmpB}}%
        \fi%
      \fi%
```

For further code for the case of `fnsymbol` please see below (`\lastpagereftext`, page 62).
The last page number is saved, in case the same page numbering scheme is continued later.

```
      \pagesLTS@providecounter{pagesLTS.double.\pagesLTS@pnc}%
      \setcounter{pagesLTS.double.\pagesLTS@pnc}{\value{page}}%
```

We went back one page, so we must go forward again:

```
      \addtocounter{pagesLTS.current.local.\pagesLTS@pnc}{+1}%
      \addtocounter{page}{+1}%
```

The page numbering scheme `\pagesLTS@pnc` is now set to the new one, given by the user as argument with the `\pagenumbering{...}` command:

```
      \xdef\pagesLTS@pnc{#1}%
```

If option `pagecontinue=false` was chosen, the local page count is reset to 1.

```
      \ifpagesLTS@pagecontinue%
      \else%
        \pagesLTS@providecounter{pagesLTS.current.local.\pagesLTS@pnc}%
        \setcounter{pagesLTS.current.local.\pagesLTS@pnc}{1}%
      \fi%
```

The new page numbering scheme is now started for real:

```
      \OrigPagenumbering{#1}%
```

If a page numbering scheme not known by the original `\pagenumbering{...}` command is used, an error will arise here - but maybe without error message.

If page numbering scheme `\alph`, `\Alph`, or `\fnsymbol` is used, pageslts extends the page numbers according to the given options, using the alphalph package. `\arabic` does not need any expansion. `\roman` and `\Roman` at least receive a definition for zero.

```
      \expandPagenumbering{#1}%
```

Counters like `pagesLTS.pnc.Roman` are introduced:

```
\pagesLTS@providecounter{pagesLTS.pnc.\pagesLTS@pnc}%
```

The saved number of times, this page numbering scheme was used, is increased by one:

```
\addtocounter{pagesLTS.pnc.\pagesLTS@pnc}{1}%
```

Now defining the counter `pagesLTS.double.\pagesLTS@pnc`, if it did not exist already, adding 1, because this is the first page of it (or another one, if the scheme is continued):

```
\pagesLTS@providecounter{pagesLTS.double.\pagesLTS@pnc}%
\addtocounter{pagesLTS.double.\pagesLTS@pnc}{1}%
```

The page number is continued, if the option `pagecontinue=false` is **not** set, otherwise it is reset to one. Note that neither the local nor the current counter are reset, as they contain the real *values* and not the *names* of the pages.

```
\ifpagesLTS@pagecontinue%
  \setcounter{page}{\value{pagesLTS.double.\pagesLTS@pnc}}%
\else%
  \setcounter{page}{1}%
\fi%
```

If it does not exist already, the counter `pagesLTS.current.local.\pagesLTS@pnc` (e. g. `pagesLTS.current.local.Roman`) is created.

```
\pagesLTS@providecounter{pagesLTS.current.local.\pagesLTS@pnc}%
```

If `pagesLTS.double.\pagesLTS@pnc` of the current page numbering scheme is equal to one, this is the first page of this page numbering scheme. Then `pagesLTS.current.local.\pagesLTS@pnc` (which was zero) is set to one.

```
\ifnum\value{pagesLTS.double.\pagesLTS@pnc}=1%
  \setcounter{pagesLTS.current.local.\pagesLTS@pnc}{1}%
\fi%
```

Otherwise, i. e. if `\value{CurrentPage}` is not > 1, i. e. before the first page has shipped out:

```
\else%
  \ifx\pagesLTS@pnc\pagesLTS@zero\relax%
  \else%
```

`\pagenumbering{...}` has been called more than once before the end of the first page (probably in the preamble). Let us reset the earlier `\pagenumbering`s, which should not have any effect anyway:

```
    \PackageNoteNoLine{pageslts}{%
      \string\pagenumbering\space used more than once before the end\MessageBreak%
      of the first page, maybe in the preamble.\MessageBreak%
      Resetting all unused page numbering schemes now}%
    \setcounter{pagesLTS.pnc.\pagesLTS@pnc}{0}%
    \setcounter{pagesLTS.double.\pagesLTS@pnc}{0}%
    \setcounter{pagesLTS.current.local.\pagesLTS@pnc}{0}%
  \fi%
```

The current page numbering scheme is defined by the argument of `\pagenumbering{...}`, which the user gave:

```
\xdef\pagesLTS@pnc{#1}%
```

and the page numbering scheme set by the original page numbering command, which resets the page number to one, but at the first page continuation does not make sense.

```
      \OrigPagenumbering{#1}%
```

If a page numbering scheme is used, which is not known by LaTeX, an error might arise here – but maybe without error message.

If page numbering scheme `\alph`, `\Alph`, or `\fnsymbol` is used, pageslts extends the page numbers according to the given options, using the alphalph package. `\arabic` does not need any expansion. `\roman` and `\Roman` at least receive a definition for zero.

```
      \expandPagenumbering{#1}%
```

We are at the first page, so the page counters are set to one:

```
      \pagesLTS@providecounter{pagesLTS.pnc.\pagesLTS@pnc}%
      \setcounter{pagesLTS.pnc.\pagesLTS@pnc}{1}%
      \pagesLTS@providecounter{pagesLTS.double.\pagesLTS@pnc}%
      \setcounter{pagesLTS.double.\pagesLTS@pnc}{1}%
      \pagesLTS@providecounter{pagesLTS.current.local.\pagesLTS@pnc}%
      \setcounter{pagesLTS.current.local.\pagesLTS@pnc}{1}%
    \fi%
```

Whether `\pagenumbering{...}` is called in the preamble, in the `begindocument` hooks, right after `\begin{document}`, or somewhere in the document, we want to remember *whether* it was called at all:

```
    \gdef\pagesLTS@called{1}%
  \fi%
```

We do not need the temporary definitions any more.

```
  \let\pagesLTS@tmpA\undefined%
  \let\pagesLTS@tmpB\undefined%
  }

```

`\lastpageref*` If hyperref is used, but (some) references to some last page shall not be hyperlinked, a command `\lastpageref*` (analogous to `\pageref*`) is needed. Therefore we define (analogous to `\HyPsd@pageref` from the hyperref package

```
%% analogous to \HyPsd@pageref from the hyperref package:
\def\lastpagereftxt#1{\pagesLTS@@pageref#1*\END}

```

Macro `\pagesLTSpageref` checks, whether a star is present (analogous to `\HyPsd@@pageref` again from the hyperref package):

```
\def\pagesLTS@@pageref#1*#2\END{%
  \ifx\\#2\\% no star
    \pagesLTS@@@pageref{#1}%
  \else% star
    \expandafter\pagesLTS@@@pagerefstar%
  \fi%
}

```

```
\def\pagesLTS@@@pageref#1{\lastpagereftext{#1}}
\def\pagesLTS@@@pagerefstar#1{\lastpagereftextstar{#1}}
```

`\lastpagereftext` When `\lastpageref` is used somewhere inside the `txt` (text), i. e. not at the last page, it is defined as `\lastpagereftxt` (see above). When the page numbering scheme is `fnsymbol`, and the hyperref package has been loaded, a hyperref instead of a label is used for the reference to `pagesLTS.fnsymbol.local.\number\value{pagesLTS.fnsymbol.local}`. (And if the `pagesLTS.fnsymbol.local` counter did not exist yet, it is created here.)

```
\newcommand{\lastpagereftext}[1]{%
  \def\pagesLTS@tmpA{#1}%
  \def\pagesLTS@tmpB{pagesLTS.fnsymbol.local}%
  \ifx\pagesLTS@tmpA\pagesLTS@tmpB%
    \pagesLTS@providecounter{pagesLTS.fnsymbol.local}%
    \IfPackageLoadedTF{hyperref}{%
      \href{\#pagesLTS.fnsymbol.local.\number\value{pagesLTS.fnsymbol.local}}%
        {\number\value{pagesLTS.fnsymbol.local}}%
```

When the page numbering scheme is `fnsymbol`, but the hyperref package has *not* been loaded, just the arabic number of the `pagesLTS.fnsymbol.local` counter is given (because there will be no hyperlink anyway).

```
     }{\number\value{pagesLTS.fnsymbol.local}}%
  \else%
```

Same for `silly` pagenumbering:

```
    \def\pagesLTS@tmpB{pagesLTS.silly.local}%
    \ifx\pagesLTS@tmpA\pagesLTS@tmpB%
      \pagesLTS@providecounter{pagesLTS.silly.local}%
      \IfPackageLoadedTF{hyperref}{%
        \href{\#pagesLTS.silly.local.\number\value{pagesLTS.silly.local}}%
          {\number\value{pagesLTS.silly.local}}%
       }{\number\value{pagesLTS.silly.local}}%
```

and for `gobble` pagenumbering:

```
    \else%
      \def\pagesLTS@tmpB{pagesLTS.gobble.local}%
      \ifx\pagesLTS@tmpA\pagesLTS@tmpB%
        \pagesLTS@providecounter{pagesLTS.gobble.local}%
        \IfPackageLoadedTF{hyperref}{%
          \href{\#pagesLTS.gobble.local.\number\value{pagesLTS.gobble.local}}%
            {\number\value{pagesLTS.gobble.local}}%
         }{\number\value{pagesLTS.gobble.local}}%
```

Otherwise just the common `\pageref` is applied:

```
      \else%
        \pageref{#1}%
      \fi%
```

```
    \fi%
  \fi%
```

We do not need the temporary definitions any more.

```
  \let\pagesLTS@tmpA\undefined%
  \let\pagesLTS@tmpB\undefined%
  }

```

`\lastpagereftextstar` And the same for the starred version, where no hyperlink is generated:

```
\newcommand{\lastpagereftextstar}[1]{%
  \def\pagesLTS@tmpA{#1}%
  \def\pagesLTS@tmpB{pagesLTS.fnsymbol.local}%
  \ifx\pagesLTS@tmpA\pagesLTS@tmpB%
     \pagesLTS@providecounter{pagesLTS.fnsymbol.local}%
     \number\value{pagesLTS.fnsymbol.local}%
  \else%
    \def\pagesLTS@tmpB{pagesLTS.silly.local}%
    \ifx\pagesLTS@tmpA\pagesLTS@tmpB%
       \pagesLTS@providecounter{pagesLTS.silly.local}%
       \number\value{pagesLTS.silly.local}%
    \else%
      \def\pagesLTS@tmpB{pagesLTS.gobble.local}%
      \ifx\pagesLTS@tmpA\pagesLTS@tmpB%
         \pagesLTS@providecounter{pagesLTS.gobble.local}%
         \number\value{pagesLTS.gobble.local}%
```

There is no `\pageref*` without hyperref.

```
      \else%
        \IfPackageLoadedTF{hyperref}{\pageref*{#1}}{\pageref{#1}}%
      \fi%
    \fi%
  \fi%
  \let\pagesLTS@tmpA\undefined%
  \let\pagesLTS@tmpB\undefined%
  }

```

`\overrideLTSlabel` `\overridelabel` from the undolabl package just `\undonewlabel`s a label and places a new `\label{#1}`, but we need to place a `\pagesLTS@putlabel{#1}{#2}`, therefore we need another command instead of (but somewhat similar to) `\overridelabel`:

```
\newcommand\overrideLTSlabel[2]{%
  \@bsphack%
  \if@filesw%
    \edef\pagesLTS@tmpC{#1}%
    \edef\pagesLTS@tmpD{pagesLTS.\pagesLTS@pnc.local}%
    \ifx\pagesLTS@tmpC\pagesLTS@tmpD%
```

```
    \ifnum\value{pagesLTS.pnc.\pagesLTS@pnc}>1%
      \immediate\write\@auxout{\string\undonewlabel{#1}}\@overriddenmessage s{#1}%
    \else%
      \edef\pagesLTS@tmpD{fnsymbol}%
      \ifx\pagesLTS@pnc\pagesLTS@tmpD%
        \immediate\write\@auxout{\string\undonewlabel{#1}}\@overriddenmessage s{#1}%
      \fi%
      \edef\pagesLTS@tmpD{gobble}%
      \ifx\pagesLTS@pnc\pagesLTS@tmpD%
        \immediate\write\@auxout{\string\undonewlabel{#1}}\@overriddenmessage s{#1}%
      \fi%
      \edef\pagesLTS@tmpD{silly}%
      \ifx\pagesLTS@pnc\pagesLTS@tmpD%
        \immediate\write\@auxout{\string\undonewlabel{#1}}\@overriddenmessage s{#1}%
      \fi%
    \fi%
  \fi%
  \pagesLTS@putlabel{#1}{#2}{0}%
\fi%
\@esphack%
}

```

`\pagesLTS@EveryShipout` Because we cannot make references to pages with `fnsymbol` page "numbers" manually with hyperref, we use `\phantomsection`s and refer to one of those. But because we do not know how many `\phantomsection`s and `\section*`s are introduced by the user (or other packages; cf. LaTeX bug 2298: knowing level of `section*`, https://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/2298&introduction=yes&state=open), we cannot refer to the last one as we did with the pages.

Therefore each page with `fnsymbol` page "number" receives a `\phantomsection` and a label, which includes a number increased by one for each page. This is done for `pagesLTS.fnsymbol.local.\number\value{pagesLTS.fnsymbol.cont}` as well as `pagesLTS.fnsymbol`, `pagesLTS.\pagesLTS@pnc`, and `pagesLTS.\pagesLTS@pnc.local`.
In case an older label already existed, it is overwritten by an `\overridelabel` command.

```
\newcommand{\pagesLTS@EveryShipout}{%
  \begingroup%
    \ifx\pagesLTS@pnc\pagesLTS@fns%
      \pagesLTS@providecounter{pagesLTS.fnsymbol.cont}%
      \addtocounter{pagesLTS.fnsymbol.cont}{1}%
      \IfPackageLoadedT{hyperref}{\phantomsection%
        \hypertarget{pagesLTS.fnsymbol.local.\number\value{pagesLTS.fnsymbol.cont}}{}%
        }%
      \ifnum\pagesLTS@esovFs=\pagesLTS@zero%
        \if@filesw\label{pagesLTS.fnsymbol}\fi%
        \pagesLTS@putlabel{pagesLTS.fnsymbol.local}{\theCurrentPageLocal}{0}%
      \else%
        \overridelabel{pagesLTS.fnsymbol}%
```

```
      \overrideLTSlabel{pagesLTS.fnsymbol.local}{\theCurrentPageLocal}%
    \fi%
  \else%
```

Same with `silly` page “numbers”:

```
    \ifx\pagesLTS@pnc\pagesLTS@silly%
      \pagesLTS@providecounter{pagesLTS.silly.cont}%
      \addtocounter{pagesLTS.silly.cont}{1}%
      \IfPackageLoadedT{hyperref}{\phantomsection%
        \hypertarget{pagesLTS.silly.local.\number\value{pagesLTS.silly.cont}}{}%
        }%
      \ifnum\pagesLTS@esovSy=\pagesLTS@zero%
        \if@filesw\label{pagesLTS.silly}\fi%
        \pagesLTS@putlabel{pagesLTS.silly.local}{\theCurrentPageLocal}{0}%
      \else%
        \overridelabel{pagesLTS.silly}%
        \overrideLTSlabel{pagesLTS.silly.local}{\theCurrentPageLocal}%
      \fi%
    \else%
```

and with `gobble` page “numbers”:

```
      \ifx\pagesLTS@pnc\pagesLTS@gobble%
        \pagesLTS@providecounter{pagesLTS.gobble.cont}%
        \addtocounter{pagesLTS.gobble.cont}{1}%
        \IfPackageLoadedT{hyperref}{\phantomsection%
          \hypertarget{pagesLTS.gobble.local.\number\value{pagesLTS.gobble.cont}}{}%
          }%
        \ifnum\pagesLTS@esovGbl=\pagesLTS@zero%
          \if@filesw\label{pagesLTS.gobble}\fi%
          \pagesLTS@putlabel{pagesLTS.gobble.local}{\theCurrentPageLocal}{0}%
        \else%
          \overridelabel{pagesLTS.gobble}%
          \overrideLTSlabel{pagesLTS.gobble.local}{\theCurrentPageLocal}%
        \fi%
      \else%
        \IfPackageLoadedT{hyperref}{\phantomsection}%
        \if@filesw%
          \overridelabel{pagesLTS.\pagesLTS@pnc}%
          \overrideLTSlabel{pagesLTS.\pagesLTS@pnc.local}{\theCurrentPageLocal}%
  \fi\fi\fi\fi%
\endgroup%
}
```

ddToHookshipout/foreground At the end of each shipout, the following commands are executed:

```
\AddToHook{shipout/foreground}{%
  \ifnum\value{page}>0%
  \else%
    \ifnum\value{page}=0%
      \PackageWarning{pageslts}{%
        Counter ‘page’ is zero!\MessageBreak%
        If the page numbering scheme is not arabic\MessageBreak%
        and further not extended\MessageBreak%
        (see Page counter overflow in the pageslts\MessageBreak%
        documentation), without other measures\MessageBreak%
        this will lead to a counter overflow.\MessageBreak%
        Code}%
    \else%
      \ifnum\value{page}<0%
        \PackageWarning{pageslts}{%
          Counter ‘page’ is negative: ‘\the\value{page}’!\MessageBreak%
          If the page numbering scheme is not arabic\MessageBreak%
          and further not extended\MessageBreak%
          (see Page counter overflow in the pageslts\MessageBreak%
          documentation), without other measures\MessageBreak%
          this will lead to a counter overflow.\MessageBreak%
          Code}%
      \else%
        \PackageError{pageslts}{Counter page has no recognized value}{%
          Counter ‘page’ does not have a recognized value:\MessageBreak%
          ‘\the\value{page}’\MessageBreak%
          \@ehd\MessageBreak%
          }%
      \fi%
    \fi%
  \fi%
```

If the `CurrentPage` is equal to one, this is the first shipout.

```
  \ifnum\value{CurrentPage}=1%
```

We check whether some page numbering scheme was defined by `\pagenumbering{...}` (**as it should be!**):

```
    \ifx\pagesLTS@called\pagesLTS@zero%
```

If it was not defined (i. e. `\pagesLTS@called` is zero), the user should be informed, that a `\pagenumbering{...}` is missing behind `\begin{document}`. Of course, it is possible that some package did some pages of output using the `begindocument/end` hook. In that case, one `\pagenumbering{...}` before `\begin{document}` and one `\pagenumbering{...}` (with the same argument, of course!) behind `\begin{document}` could help somewhat. When `\PackageError` was used here, the error message was not written to the screen and the `.log`-file, but into the document. Therefore we just make a note to give the error message later (`\AddToHook{enddocument/afterlastpage}`). At that time unfortunately most of the document has already been compiled (or did not compile due to this error), but I do not know how to change that.

```
      \gdef\pncmissing{1}%
```

We save the current value of the page,

```
      \setcounter{pagesLTS.tmpcounter}{\value{page}}%
```

determine the current page numbering scheme,

```
%% Code from Andres L\"{o}h, Universiteit Utrecht (NL) %%
      \def\extract#1{\expandafter\extract@ #1\END}%
      \def\extract@#1\csname @#2\endcsname#3\END{#2}%
      \edef\pagesLTS@tmpQ{\extract\thepage}%
%% End of code from Andres L\"{o}h                      %%
```

set the current page numbering scheme to 0 (because before the beginning of the document it should be 0),

```
      \def\pagesLTS@pnc{0}%
```

and then issue a `\pagenumbering` command with the determined page numbering scheme as argument:

```
      \expandafter\pagenumbering\expandafter{\pagesLTS@tmpQ}%
```

This resets the page to one (if option `pagecontinue=false` was chosen), but because we do not start a new page numbering scheme here but manifest a page numbering scheme, which the user forgot to define, the page number should not have been reset to one. (This is the first page, but maybe the user wants it to have page number 2024?) Therefore we revert this here and set the page number to its value, which was saved before the `\pagenumbering` command.

```
      \setcounter{page}{\value{pagesLTS.tmpcounter}}%
    \fi%
```

We are at the first page, so we put the label here.

```
    \pagesLTS@writelabel{0}%
  \fi%
```

If the current page numbering scheme `\pagesLTS@pnc` is `\pagesLTS@fns` (which is defined as `fnsymbol`), the label is set by `\pagesLTS@EveryShipout` (see just above), and `\pagesLTS@esovFs` is set to the (real) number (not the name) of this page numbering scheme, `\number\value{pagesLTS.fnsymbol.cont}`.
When no more pages with `fnsymbol` page "number" are shipped out, the value remains fixed and we have our reference to the last page of the `fnsymbol` page numbering range. (At least we will have that reference after some more work, see below).

```
  \ifx\pagesLTS@pnc\pagesLTS@fns%
    \pagesLTS@EveryShipout%
    \xdef\pagesLTS@esovFs{\number\value{pagesLTS.fnsymbol.cont}}%
  \else%
```

Same for `silly` page "number":

```
    \ifx\pagesLTS@pnc\pagesLTS@silly%
      \pagesLTS@EveryShipout%
      \xdef\pagesLTS@esovSy{\number\value{pagesLTS.silly.cont}}%
    \else%
```

and for gobble page “number”:

```
    \ifx\pagesLTS@pnc\pagesLTS@gobble%
      \pagesLTS@EveryShipout%
      \xdef\pagesLTS@esovGbl{\number\value{pagesLTS.gobble.cont}}%
```

When another page numbering scheme was reused (in the example file Roman), we also need to apply \pagesLTS@EveryShipout, because otherwise we would get multiply defined labels.

```
    \else%
      \ifnum\value{pagesLTS.pnc.\pagesLTS@pnc}>1%
        \pagesLTS@EveryShipout%
\fi\fi\fi\fi%
```

The CurrentPage as well as the pagesLTS.current.local.\pagesLTS@pnc are advanced by one (because one page was shipped out and the next is about to begin).

```
\addtocounter{CurrentPage}{1}%
\addtocounter{pagesLTS.current.local.\pagesLTS@pnc}{1}%
}

```

We use some code from the lastpage package:

\protected@iwrite We need an \immediate\protected@write. Just \immediate\write had led to errors, for example when packages like babel-greek re-defined \roman.

```
%% Code provided by Prof. Enrico Gregorio at https://tex.stackexchange.com/a/542425
\long\def\protected@iwrite#1#2#3{%
  \begingroup%
  #2%
  \let\protect\@unexpandable@protect%
  \edef\reserved@a{\immediate\write#1{#3}}%
  \reserved@a%
  \endgroup%
  \if@nobreak\ifvmode\nobreak\fi\fi%
  }

```

\pagesLTS@makeHy Just once we need the page from \@currentHpage without any “page.” (\pagesLTS@makeHy is identical to \lastpage@makeHy except of the used namings – well, being identical depends on the versions of the packages):

```
\newcommand{\pagesLTS@makeHy}{%
```

\gdef\pagesLTS@Hy{}, but that was already done before this command.

```
  \xdef\pagesLTS@Hptest{Doc-Start}%
  \ifx\pagesLTS@Hptest\@currentHpage\relax%
    \gdef\pagesLTS@Hy{\@currentHpage}%
  \else%
    \edef\pagesLTS@Hptest{\@currentHpage}%
    \ifx\pagesLTS@Hptest\empty\relax%
```

then \gdef\pagesLTS@Hy{}, but that was already done before this command.

```
    \else%
      \edef\pagesLTS@Hptest{page.}%
      \ifx\@currentHpage\pagesLTS@Hptest\relax%
        \ifx\pagesLTS@pnc\pagesLTS@gobble%
          \PackageNoteNoLine{pageslts}{%
            With gobbled page numbers "\string\@currentHpage "\MessageBreak%
            is just "page." without number.\MessageBreak%
            "\string\pagesLTSlastpageHy " is now let empty}%
        \else%
          \PackageWarningNoLine{pageslts}{%
            \string\@currentHpage\space is just "page." without number.\MessageBreak%
            \string\pagesLTSlastpageHy\space is now let empty}%
        \fi%
```

then in both cases \gdef\pagesLTS@Hy{}, but that was already done before this command.

```
      \else%
```

\@currentHpage should be page.<some number>, \pagesLTS@rmpage removes the "page.".
Next compilation run, \pagesLTSlastpageHy gets defined via the aux file. If we arrived at this place, but the definition is still empty, then \@currentHpage has some unexpected content.

```
        \gdef\pagesLTS@Hy{\pagesLTS@rmpage{\@currentHpage}}%
        \ifx\pagesLTSlastpageHy\empty\relax%
          \PackageWarningNoLine{pageslts}{%
            \string\@currentHpage\space is\MessageBreak%
            \meaning\@currentHpage\MessageBreak%
            not beginning with "page.",\MessageBreak%
            \string\pagesLTSlastpageHy\space is now let empty}%
  \fi\fi\fi\fi%
  }%

```

\pagesLTS@rmpage \pagesLTS@rmpage removes the "page.".

```
%% Code provided by David Carlisle at https://tex.stackexchange.com/a/721877
\def\pagesLTS@rmpage#1{%
  \expandafter\xpagesLTS@rmpage\expanded{#1}\xpagesLTS@rmpage page.%
  \xpagesLTS@rmpage\xxpagesLTS@rmpage{#1}}
\def\xpagesLTS@rmpage #1page.#2\xpagesLTS@rmpage#3\xxpagesLTS@rmpage#4{%
  \if$\detokenize{#1}$#2%\else#4
  \fi}

```

\else#4 means, that it did not start with page., and whatever it is, we cannot use this #4 for \pagesLTS@Hy.

\pagesLTS@putlabel Since the page has been put out, we are on the page after that page. We therefore subtract one from the page counter.

```
\newcommand{\pagesLTS@putlabel}[3]{%
  \addtocounter{page}{-1}%
```

When the showkeys package has been loaded in `draft` mode, in the margin for each label a box is displayed with the name of the label. showkeys accomplishes this by redefining `\label`, but pageslts does not use `\label`, but writes directly to the `\jobname.aux`-file, and this is generally done after the according page has shipped out, therefore no box can be placed on the preceding page. At least pageslts gives a warning, that showkeys cannot present the respective label.

```
  \IfPackageLoadedT{showkeys}{%
    \IfPackageLoadedWithOptionsF{showkeys}{final}{%
      \ifnum\value{pagesLTS.pagenr}<1%
        \PackageWarningNoLine{pageslts}{%
          Package showkeys without option final loaded,\MessageBreak%
          but label #1 on page \thepage\space(about \theCurrentPage)\MessageBreak%
          cannot be shown, because pageslts does not use \string\label,\MessageBreak%
          but writes directly to the \jobname.aux file}%
      \else%
        \PackageWarningNoLine{pageslts}{%
          Package showkeys without option final loaded,\MessageBreak%
          but label #1 on page \thepage\space (about \theCurrentPage\space of %
          \number\value{pagesLTS.pagenr})\MessageBreak%
          cannot be shown, because pageslts does not use \string\label,\MessageBreak%
          but writes directly to the \jobname.aux file}%
      \fi%
    }%
  }%
  \IfPackageLoadedTF{hyperref}{%
```

Here the labels are set, if the hyperref package was loaded. Simply using `\label` would not work, because labels wait for the output routines to work, and there may not be any more invocations of the output routines. To force the write out we need to do an `\immediate` write (`\protected@iwrite`).

```
    \ifHy@pageanchor\else%
```

If the hyperref package is used, but pageanchors are disabled, the hyperlinking will not work.

```
      \PackageWarningNoLine{pagesLTS}{%
        The \string\lastpageref{#1} link does not work\MessageBreak%
        using hyperref with disabled option 'pageanchor'.\MessageBreak%
        Better enable 'pageanchor' or use\MessageBreak%
        \string\lastpageref*{...} (not generating a link)%
      }%
    \fi%
```

If use of the `.aux`-file is allowed, the label for `LastPage` is written into that file, the page reference depending on the options, which where set for the hyperref package.

```
    \if@filesw%
      \begingroup%
```

`\@currentlabelname` should have been sanitized, but sometimes it is not. Therefore we get rid of any possible `\label`, `\index` and `\glossary` contained here. This code is inside a `\begingroup`...`\endgroup`, thus there is no need to save and later restore the original meaning of those commands.

```
        \let\label\@gobble@om%
        \let\index\@gobble@som%
        \let\glossary\@gobble@om%
        \protected@iwrite\@auxout{}{\string\newlabel{#1}{%
          {\@currentlabel}{#2}{\@currentlabelname}%
          {\ifx\pagesLTS@pnc\pagesLTS@fns%
             \@currentHref%
           \else\ifx\pagesLTS@pnc\pagesLTS@silly%
             \@currentHref%
             \else\ifx\pagesLTS@pnc\pagesLTS@gobble%
               \@currentHref%
               \else\ifHy@pageanchor\@currentHpage\fi%
           \fi\fi\fi%
          }%
          {\@kernel@reserved@label@data}}}%
      \endgroup%
    \fi%
   }{%
```

Also when hyperref has not been loaded, the same procedure for `\@currentlabelname` is done:

```
     \begingroup%
       \let\label\@gobble@om%
       \let\index\@gobble@som%
       \let\glossary\@gobble@om%
```

If the hyperref package is not used, there will be no hyperlinks, and the label is written in the way of the lastpage package. But we must remember to undo the label first, if it already exists.

```
      \if@filesw%
        \ifnum\value{pagesLTS.pnc.\pagesLTS@pnc}<2%
            \protected@iwrite\@auxout{}{\string\newlabel{#1}{%
              {\@currentlabel}{#2}{\@currentlabelname}%
              {\@currentHref}{\@kernel@reserved@label@data}}%
            }%
        \else%
          \edef\pagesLTS@tmpC{#1}%
          \edef\pagesLTS@tmpD{pagesLTS.\pagesLTS@pnc.local}%
          \ifx\pagesLTS@tmpC\pagesLTS@tmpD%
            \edef\pagesLTS@tmpC{#3}%
```

```
            \ifx\pagesLTS@tmpC\pagesLTS@one%
```

Only when the third argument of `\pagesLTS@putlabel` is 1, we do need to undo the label. Otherwise there is no label to undo, and the undolabl package would give an error.

```
              \immediate\write\@auxout{\string\undonewlabel{#1}}%
            \fi%
          \fi%
          \protected@iwrite\@auxout{}{\string\newlabel{#1}{%
            {\@currentlabel}{#2}{\@currentlabelname}%
            {\@currentHref}{\@kernel@reserved@label@data}}%
          }%
        \fi%
      \fi%
    \endgroup%
   }%
```

After the writeout we restore the page number again, since there might be other things still to be done.

```
  \addtocounter{page}{+1}%
  }
```

`\pagesLTS@putlabels` `\pagesLTS@putlabels` writes labels and calls `\pagesLTS@putlabel{LastPages}{\theCurrentPage}{1}`.

```
\newcommand{\pagesLTS@putlabels}{%
  \IfFormatAtLeastF{2025-06-01}{\addtocounter{page}{-1}}%
  \addtocounter{CurrentPage}{-1}%
  \addtocounter{pagesLTS.current.local.\pagesLTS@pnc}{-1}%
```

If `\pagenumbering{...}` has not been used, `\pagesLTS@pnc` is still zero (0, `\pagesLTS@zero`), and the according warning message is given.

```
  \ifx\pagesLTS@pnc\pagesLTS@zero%
   \PackageWarningNoLine{pagesLTS}{No page numbering scheme found:\MessageBreak%
     \pagesLTS@messageNPN}%
```

otherwise the numbered label is written, and if the page numbering scheme was not used before, also the unnumbered label is written.

```
  \else%
    \pagesLTS@writelabel{\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}}%
    \ifnum\value{pagesLTS.pnc.\pagesLTS@pnc}<2%
      \ifx\pagesLTS@pnc\pagesLTS@fns%
      \else\ifx\pagesLTS@pnc\pagesLTS@silly%
        \else\ifx\pagesLTS@pnc\pagesLTS@gobble%
          \else\pagesLTS@writelabel{\pagesLTS@pnc}%
  \fi\fi\fi\fi\fi%
```

Before the label for the `LastPages` can be put, we must advance one page again (if we changed it), because `\pagesLTS@putlabel` itself goes back one page (and at its end forward again).

```
  \IfFormatAtLeastF{2025-06-01}{\addtocounter{page}{+1}}%
  \pagesLTS@putlabel{LastPages}{\theCurrentPage}{1}%
```

Here should follow a
`\addtocounter{page}{-1}`,
but we have to remember to increase the page counters again, which were decreased at the start of this `\pagesLTS@putlabels` command, and that would include
`\addtocounter{page}{+1}`,
therefore these two lines cancel each other and therefore they just can be skipped. But the other counters have to be increased again:

```
  \addtocounter{CurrentPage}{+1}%
  \addtocounter{pagesLTS.current.local.\pagesLTS@pnc}{+1}%
  }
```

dToHook{begindocument/end} `\AddToHook{begindocument/end}{...}`:

```
\AddToHook{begindocument/end}{%
```

Checking, whether lastpage and pageslts have both been loaded.

```
  \IfPackageLoadedT{lastpage}{%
    \PackageNoteNoLine{pageslts}{Packages lastpage and pageslts used:\MessageBreak%
      Are you sure that you want to load both packages?}%
    }%
```

It is checked whether the hyperref package, if loaded, is recent enough:

```
  \IfPackageLoadedTF{hyperref}{%
     \ProvideTextCommand{\textasteriskcentered}{PD1}{*}%
     \IfPackageAtLeastF{hyperref}{2024-11-05}{%
       \PackageError{pageslts}{hyperref package version too old}{%
         Required version: 2024-11-05 v7.01l or newer,\MessageBreak%
         found version:\space\space\space\space\csname ver@hyperref.sty\endcsname.}}%
   }{\PackageNoteNoLine{pageslts}{Package hyperref not loaded.\MessageBreak%
       pageslts would support hyperref.\MessageBreak%
       The page references will NOT be hyperlinked}%
    }%
```

Further it is checked whether the alphalph package has been loaded. If that is the case, the commands are defined accordingly.

```
  \IfPackageLoadedT{alphalph}{%
    \newalphalph{\AlphMult}[mult]{\@Alph}{26}%
    \newalphalph{\alphMult}[mult]{\@alph}{26}%
    \newalphalph{\fnsymbolmult}[mult]{\@fnsymbol}{5}%
    }%
```

It is checked whether writing to files is allowed. The pageslts package cannot be used without that! Some packages (e. g. tikz and selectp) sometimes prevent the output to the aux file. In that case a warning or an error message is issued. This is no problem as long as there is/was another compilation run where the labels can/could be processed via the aux file.

```
\if@filesw%
\else%
  \IfPackageLoadedTF{tikz}{%
    \PackageWarningNoLine{pageslts}%
      {It was not allowed to write to an\MessageBreak%
       .aux file. This package does not work without access\MessageBreak%
       to an .aux file.\MessageBreak%
       It is OK if the .aux file was already updated\MessageBreak%
       by a previous compiler run\MessageBreak%
       and would not have changed anyway%
      }%
  }{\PackageError{pageslts}{No writing to auxiliary file allowed}%
      {It was not allowed to write to an .aux file.\MessageBreak%
       This package does not work without access to an .aux file.\MessageBreak%
       Press Ctrl+Z to exit.\MessageBreak%
       But it is OK if the .aux file was already updated\MessageBreak%
       by previous compiler runs\MessageBreak%
       and would not have changed anyway.\MessageBreak%
       (In that case just press Enter or Return\MessageBreak%
       to continue the compilation.)%
      }%
   }%
\fi%
}

```

We do check whether showkeys has been loaded (before pageslts!) in a mode to show the labels:

```
\def\pagesLTS@skld{0}
\IfPackageLoadedT{showkeys}{\IfPackageLoadedWithOptionsF{showkeys}{final}{%
  \def\pagesLTS@skld{1}}%
  }%

```

enddocument/afterlastpage} \AddToHook{enddocument/afterlastpage} it is checked whether the showkeys package was loaded after the check performed before, in which case an error message is issued.

```
\AddToHook{enddocument/afterlastpage}{%
  \IfPackageLoadedT{showkeys}{\IfPackageLoadedWithOptionsF{showkeys}{final}{%
    \ifx\pagesLTS@skld\pagesLTS@one\relax%
    \else\PackageError{pageslts}{Package showkeys loaded after pageslts}{%
      The showkeys package with option "final" has been loaded\MessageBreak%
      after the pageslts package.\MessageBreak%
```

```
    Please first load showkeys and then pageslts.}%
  \fi}%
}%
```

We also give the error message about the missing (i. e. not found) page numbering scheme, which could not be given in hook `shipout/foreground`.

```
\ifx\pncmissing\pagesLTS@one\relax%
  \PackageError{pageslts}{pagenumbering scheme missing}{\pagesLTS@messageNPN}%
\fi%
```

Then we put in an `info` to show, in what order things (which were called) are done.

```
\PackageInfo{pageslts}{enddocument/afterlastpage (AED): pageslts setting LastPage}%
```

After this we remember the page number (if `fnsymbol` or `silly` or `gobble`), and after that we place the `LastPage` label.

```
\ifx\pagesLTS@pnc\pagesLTS@fns%
  \def\pagesLTS@tmpA{\number\value{pagesLTS.fnsymbol.local}}%
  \ifnum\pagesLTS@esoFs=\pagesLTS@tmpA%
    \gdef\pagesLTS@rerun{0}%
  \else%
    \gdef\pagesLTS@rerun{1}%
  \fi%
  \if@filesw%
    \immediate\write\@auxout{\string\gdef\string\pagesLTS@esoFs{\pagesLTS@tmpA}}%
  \fi%
\else%
  \ifx\pagesLTS@pnc\pagesLTS@silly%
    \def\pagesLTS@tmpA{\number\value{pagesLTS.silly.local}}%
    \ifnum\pagesLTS@esoSy=\pagesLTS@tmpA%
      \gdef\pagesLTS@rerun{0}%
    \else%
      \gdef\pagesLTS@rerun{1}%
    \fi%
    \if@filesw%
      \immediate\write\@auxout{\string\gdef\string\pagesLTS@esoSy{\pagesLTS@tmpA}}%
    \fi%
  \else%
    \ifx\pagesLTS@pnc\pagesLTS@gobble%
      \def\pagesLTS@tmpA{\number\value{pagesLTS.gobble.local}}%
      \ifnum\pagesLTS@esoGbl=\pagesLTS@tmpA%
        \gdef\pagesLTS@rerun{0}%
      \else%
        \gdef\pagesLTS@rerun{1}%
      \fi%
      \if@filesw%
        \immediate\write\@auxout{\string\gdef\string\pagesLTS@esoGbl{\pagesLTS@tmpA}}%
\fi\fi\fi\fi%
```

```
\IfFormatAtLeastT{2025-06-01}{\addtocounter{page}{+1}}%
\pagesLTS@putlabel{LastPage}{\thepage}{1}%
\IfFormatAtLeastT{2025-06-01}{\addtocounter{page}{-1}}%
```

We do not need the temporary definition any more.

```
\let\pagesLTS@tmpA\undefined%
```

The number of pages with the `fnsymbol` page numbering scheme, `\pagesLTS@esovFs`, and the number of pages with the `silly` page numbering scheme, `\pagesLTS@esovSy`, and the number of pages with the `gobble` page numbering scheme, `\pagesLTS@esovGbl`, are saved via the `.aux` file (if it is not zero, respectively):

```
\if@filesw%
  \ifx\pagesLTS@esovFs\pagesLTS@zero\relax%
  \else%
    \immediate\write\@auxout{\string\pagesLTS@providecounter{pagesLTS.fnsymbol.local}}%
    \immediate\write\@auxout{\string\setcounter{pagesLTS.fnsymbol.local}{\pagesLTS@esovFs}}%
  \fi%
  \ifx\pagesLTS@esovSy\pagesLTS@zero\relax%
  \else%
    \immediate\write\@auxout{\string\pagesLTS@providecounter{pagesLTS.silly.local}}%
    \immediate\write\@auxout{\string\setcounter{pagesLTS.silly.local}{\pagesLTS@esovSy}}%
  \fi%
  \ifx\pagesLTS@esovGbl\pagesLTS@zero\relax%
  \else%
    \immediate\write\@auxout{\string\pagesLTS@providecounter{pagesLTS.gobble.local}}%
    \immediate\write\@auxout{\string\setcounter{pagesLTS.gobble.local}{\pagesLTS@esovGbl}}%
  \fi%
\fi%
```

At the call of a `\pagenumbering{...}` command, everything for a split page numbering scheme is organized. For the last page numbering scheme, there is no `\pagenumbering{...}` command at the end, so we need to handle this here:

```
\pagesLTS@providecounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}%
\setcounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}{%
  \value{pagesLTS.current.local.\pagesLTS@pnc}}%
```

And we are one page after the last one (hook `enddocument/afterlastpage`!), so we go back one page.

```
\addtocounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}{-1}%
\ifnum\value{pagesLTS.pnc.\pagesLTS@pnc}>1%
  \setcounter{pagesLTS.tmpcounter}{\value{pagesLTS.pnc.\pagesLTS@pnc}}%
  \@tempcnta=\value{pagesLTS.tmpcounter}\relax%
  \loop%
    \ifnum\@tempcnta>1\relax%
      \addtocounter{pagesLTS.tmpcounter}{-1}%
      \addtocounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}{%
        -\value{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.tmpcounter}.local.count}}%
      \@tempcnta=\value{pagesLTS.tmpcounter}\relax%
  \repeat%
```

```
\fi%
\if@filesw%
  \immediate\write\@auxout{\string
    \pagesLTS@providecounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.cnt}}%
  \edef\pagesLTS@tmpA{\number\value{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.count}}%
  \immediate\write\@auxout{\string
    \setcounter{pagesLTS.\pagesLTS@pnc.\number\value{pagesLTS.pnc.\pagesLTS@pnc}.local.cnt}{\pagesLTS@tmpA}}%
\fi%
```

We need to save (via the `.aux` file) the page name `\thepage` and the page number `\number\value{CurrentPage}` of the last page, in case the last page has `fnsymbol` page numbering scheme.

```
\IfFormatAtLeastF{2025-06-01}{\addtocounter{page}{-1}}%
\def\pagesLTS@tmpA{\thepage}%
\if@filesw%
  \protected@iwrite\@auxout{}{\string\gdef\string\pagesLTSlastpage{\pagesLTS@tmpA}}%
  \gdef\pagesLTS@Hy{}%
  \IfPackageLoadedT{hyperref}{\ifHy@pageanchor\pagesLTS@makeHy\fi}%
  \protected@iwrite\@auxout{}{\string\gdef\string\pagesLTSlastpageHy{\pagesLTS@Hy}}%
\fi%
\IfFormatAtLeastF{2025-06-01}{\addtocounter{page}{-1}}%
\addtocounter{CurrentPage}{-1}%
\edef\pagesLTS@tmpB{\number\value{CurrentPage}}%
\if@filesw%
  \immediate\write\@auxout{\string\setcounter{pagesLTS.pagenr}{\pagesLTS@tmpB}}%
\fi%
\addtocounter{CurrentPage}{+1}%
```

The `VeryLastPage` label is set here. Before the introduction of the hook mechanism in the kernel, `\lastpageref{VeryLastPage}` could point to a later page then `\lastpageref{LastPage}`. `LastPage` and `VeryLastPage` should now be identical, but for backward compatibility we keep both.

```
\PackageInfo{pageslts}{enddocument/afterlastpage (AED): pageslts setting VeryLastPage}%
\IfFormatAtLeastT{2025-06-01}{\addtocounter{page}{+1}}%
\pagesLTS@putlabel{VeryLastPage}{\thepage}{1}%
\IfFormatAtLeastT{2025-06-01}{\addtocounter{page}{-1}}%
```

The `LastPages` label is set here, and `\lastpageref{LastPages}` gives the total number of pages and points to the last page.

```
\PackageInfo{pageslts}{enddocument/afterlastpage (AED): pageslts setting LastPages}%
\pagesLTS@putlabels%
\ifodd\pagesLTS@tmpB%
  \PackageNoteNoLine{pageslts}{Total number of pages is odd}%
\else%
  \PackageNoteNoLine{pageslts}{Total number of pages is even}%
\fi%
}
```

\frontmatter \frontmatter often contains \pagenumbering, but for some unknown reason there are problems when another \pagenumbering with different page numbering scheme has been used before on the same page. (This would not make any sense anyway, because one page can only have one page numbering scheme.) This problem does not occur when two \pagenumbering commands are used inside normal text. Thus we need to check whether \frontmatter has been defined, whether it changes the page numbering scheme, and whether the page numbering scheme before \frontmatter was initiated at the same page. With the hook mechanism it does not matter whether the command to be patched is defined.

```
\AddToHook{cmd/frontmatter/before}{%
  \xdef\pagesLTS@FMBpncn{\pagesLTS@pnc}%
  \xdef\pagesLTS@FMBpncp{\number\value{pagesLTS.current.local.\pagesLTS@pnc}}%
  }

```

At the beginning of the \frontmatter (FMB), we remember the current (c) page numbering (pn) scheme: its name (n) and page number (p). \pagesLTS@FMBpncp would be 1 if the according \pagenumbering command was used on the same page at the \frontmatter.
At the end of the \frontmatter the named checks are performed and in case of the named problem an error message is given.

```
\AddToHook{cmd/frontmatter/after}{%
  \xdef\pagesLTS@FMEpncn{\pagesLTS@pnc}%
  \ifx\pagesLTS@FMBpncn\pagesLTS@FMEpncn%
  \else%
    \ifx\pagesLTS@FMBpncn\pagesLTS@zero%
    \else%
      \ifx\pagesLTS@FMBpncp\pagesLTS@one%
        \PackageWarning{pageslts}{\string\pagenumbering\space before \string\frontmatter:\MessageBreak%
          Do not use \string\pagenumbering{\pagesLTS@FMBpncn} before \string\frontmatter\MessageBreak%
          on the same page!\MessageBreak%
          \string\frontmatter\space (re)defines the page numbering scheme to \pagesLTS@FMEpncn ,\MessageBreak%
          thus earlier use of \string\pagenumbering{\pagesLTS@FMBpncn} %
          on the same page is useless anyway\MessageBreak}%
      \fi%
    \fi%
  \fi%
  }

```

ddToHook{enddocument/info} The hook `enddocument/info` is even later:

> "This hook is meant to receive code that write final information messages to the terminal. It follows immediately after the previous hook"

(Frank Mittelbach (2024-11-25): LaTeX's hook management, lthooks-code.pdf, p. 28) `enddocument/afteraux`. Here it is used for a rerun hint.
For example if the page numbering scheme of the last page of the pageslts-example.tex file is changed to `fnsymbol` and two runs of pdfLaTeX are done, pdfLaTeX will be happy and will not complain about changed labels. But indeed, a third run is necessary and indicated by the warning message below.

```
\AddToHook{enddocument/info}{%
 \ifx\pagesLTS@rerun\pagesLTS@one%
   \PackageWarningNoLine{pageslts}{Label(s) may have changed.\MessageBreak%
     Rerun to get cross-references right}%
 \fi%
 }

⟨/package⟩
```

7 Installation

7.1 Downloads

Everything is available at https://CTAN.org, but may need additional packages themselves.

`pageslts.dtx` For unpacking the `pageslts.dtx` file and constructing the documentation it is required:

- TEXFormat LaTeX 2$_\varepsilon$ 2025-06-01: https://CTAN.org
- document class ltxdoc, 2024/02/08, v2.1j, https://ctan.org/pkg/ltxdoc
- package holtxdoc, 2019/12/09, v0.30, https://ctan.org/pkg/holtxdoc
- package hypdoc, 2023-10-26, v1.19, https://ctan.org/pkg/hypdoc
- package geometry, 2020/01/02, v5.9, https://ctan.org/pkg/geometry
- package ulem, 2019-11-18, no version number given, https://ctan.org/pkg/ulem

`pageslts.sty` The `pageslts.sty` for LaTeX 2$_\varepsilon$ $\geq$ 2025-06-01 (i. e. all documents using the pageslts package) requires:

- TEX Format LaTeX 2$_\varepsilon$ 2025-06-01, https://CTAN.org
- package undolabl, 2025-01-28, v1.0o, https://ctan.org/pkg/undolabl
- if any of the options `alphMult`, `AlphMulti`, or `fnsymbolmult` is used: package alphalph, 2019/12/09, v2.6, https://ctan.org/pkg/alphalph
- if `\pagenumbering{silly}` shall be used, the sillypage package must be loaded, 2024-07-23, v1.7, https://www.ctan.org/pkg/sillypage

`pageslts-example.tex` The `pageslts-example.tex` requires the same files as all documents using the pageslts package, and additionally:

- class article, 2025/01/22, v1.4n, from classes.dtx: https://ctan.org/pkg/classes
- package alphalph, 2019/12/09, v2.6, https://ctan.org/pkg/alphalph
- package lipsum, 2021-09-20, v2.7, https://ctan.org/pkg/lipsum
- package showkeys, 2024/05/23, v3.21, https://ctan.org/pkg/showkeys
- package hyperref, 2025-07-12, v7.01o, https://ctan.org/pkg/hyperref
- package pageslts, 2025-08-14, v2.0g, https://ctan.org/pkg/pageslts
  (Well, it is the example file for this package, and because you are reading the documentation for the pageslts package, it can be assumed that you already have some version of it – is it the current one?)

`Münch` A hyperlinked list of my (other) packages can be found at https://ctan.org/author/muench-hm.

7.2 Package, unpacking TDS

Package. This package is available on https://CTAN.org.

https://mirror.ctan.org/macros/latex/contrib/pageslts/pageslts.dtx
: The source file.

https://mirror.ctan.org/macros/latex/contrib/pageslts/pageslts.pdf
: The documentation.

https://mirror.ctan.org/macros/latex/contrib/pageslts/pageslts-example.pdf
: The compiled example file, as it should look like.

https://mirror.ctan.org/macros/latex/contrib/pageslts/README
: The README file.

There is also a `pageslts.tds.zip` available:

https://mirror.ctan.org/install/macros/latex/contrib/pageslts.tds.zip
: Everything in TDS compliant, compiled format.

which additionally contains

pageslts.ins	The installation file.
pageslts.drv	The driver to generate the documentation.
pageslts.sty	The style file.
pageslts-example.tex	The example file.

For required other packages please see the preceding subsection.

Unpacking. The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `..dtx` through plain TEX:

```
tex pageslts.dtx
```

About generating the documentation see paragraph 7.4 below.

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
pageslts.sty          → tex/latex/pageslts.sty
pageslts.pdf          → doc/latex/pageslts.pdf
pageslts-example.tex → doc/latex/pageslts-example.tex
pageslts-example.pdf → doc/latex/pageslts-example.pdf
pageslts.dtx          → source/latex/pageslts.dtx
```

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

7.3 Refresh file name databases

If your TEX distribution (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

7.4 Some details for the interested

Unpacking with LATEX. The `.dtx` chooses its action depending on the format:

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{pageslts.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by a configuration file `ltxdoc.cfg`. For instance, put the following line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfLATEX:

```
pdflatex pageslts.dtx
makeindex -s gind.ist pageslts.idx
pdflatex pageslts.dtx
makeindex -s gind.ist pageslts.idx
pdflatex pageslts.dtx
```

7.5 Compiling the example

The example file, pageslts-example.tex, can be compiled via
`latex pageslts-example.tex`
or (recommended)
`pdflatex pageslts-example.tex`
and will need *at least* (!) three compiler runs to get all references right.

8 Acknowledgements

I (H.-Martin Münch) would like to thank Jeffrey P. Goldberg (jeffrey+news at goldmark dot org) for inventing the lastpage package. This pageslts package first started as a revision of the lastpage package, but a replacement was deemed necessary to accomplish what this package does. Further I would like to thank Heiko Oberdiek for providing the `\erroralphalph` command as well as a lot (!) of useful packages (from which I also got everything I know about creating a file in `dtx` format, OK, say it: copying), Ulrich Diez for his code for the undolabl package, which allows overwriting of labels, and Andres Löh for the code to determine the current page numbering scheme. For bug reports I thank Michał Herman, kwikwi, Joshua Ellis, and Dr. Clea F. Rees. For telling me how to fix a bug (and for all his shared whisdom at https://tex.stackexchange.com) thanks go to Prof. Enrico Gregorio.

9 History

[1994/06/17, lastpage]

- lastpage v0.99a: First shot by Jeffrey P. Goldberg.

[1994/06/25, lastpage]

- lastpage v0.1b: Last version number created by Jeffrey P. Goldberg.

[1994/07/20, lastpage]

- lastpage v0.1b (again): Documentation updated by Jeffrey P. Goldberg.
  The main source code of the lastpage package 1994/07/20 v0.1b was:

```
\NeedsTeXFormat{LaTeX2e}[1994/06/01]
\ProvidesPackage{lastpage}[1994/07/20 v0.1b
  LaTeX2e package for refs to last page number (JPG)]
\def\lastpage@putlabel{\addtocounter{page}{-1}%
  \immediate\write\@auxout{\string
  \newlabel{LastPage}{{}{\thepage}}}%
  \addtocounter{page}{1}}
\AtEndDocument{%
  \message{AED: lastpage setting LastPage}%
  \clearpage\lastpage@putlabel}%
\endinput
```

  and then hyperref and revtex even redefine `\lastpage@putlabel`.

[2010/02/18, lastpage]

- lastpage v1.1: Proposed `LastPages` label by H.-Martin Münch on news:comp.text.tex, see e. g. https://groups.google.com/group/comp.text.tex/msg/4407493da9c747f0?dmode=source; now available in this pageslts package.

[2010/05/15 v1.0 pagesLTS]

- pagesLTS Complete rewriting of the package, so as to work with **more than one page numbering scheme**; using `\AtVeryEnd` for `VeryLastPage`; upgrade from fancyheadings to fancyhdr package, then removed the need for a fancyhdr package at all.
- Rewriting of the package, so as to work with the `fnsymbol` page numbering scheme (even on the last page).
- Introduction of kvoptions into this package.
- Check for incompatible endfloat package.
- lastpage209.sty for LaTeX209 .
- Replacement of `\filedate`, `-version`, `-name`,... because of LaTeX bug 2705:
  Synopsis: Possible problem with `\fileversion` and `\filedate`
  https://www.latex-project.org/cgi-bin/ltxbugs2html?category=LaTeX&responsible=anyone&state=anything&keyword=lastpage&pr=latex%2F2705&search=
- alphalph support included.
- Page numbering extension `\erroralph` by Heiko Oberdiek included.
- (Page-)Numbering extension for `roman` and `Roman` numbers included.
- Incompatible, old lastpage package "killed".
- Example `pagesLTS-example.tex` added.
- Alternatives listing (section 4).
- Listing of TeX sources (subsection 7.1).
- Complete rewriting of the documentation.
- Everything in DTX framework.
- New package name: pagesLTS for Last, Total, and page numbering Schemes pages.

[2010/06/01 v1.1(a) pagesLTS]

- Abstract changed: Negative `roman` and `Roman` page numbers are now possible.
- Several typing mistakes have been corrected - both in the `style` file as well as in this documentation.

[2010/06/03 v1.1b **pagesLTS**]

- Corrected a bug in `\XXRoman`, where `\roman` instead of `\Roman` had been used.
- New papermas package mentioned.

[2010/06/24 v1.1c **pagesLTS**]

- holtxdoc warning in `drv` updated.
- Removed CRLF line endings from the `dtx` file.
- Corrected the location of the package at CTAN. (In this version `TDS` was still missing due to a packaging error.)
- Corrected Message format in pagesLTS.`ins`.

[2010/07/15 v1.1d **pagesLTS**]

- Added the `\@ifclassloaded{revtex4}` code for `\lastpage@putlabel` (changed to `\pagesLTS@putlabelhyper`) from the hyperref package **as comment**.
- In the documentation added the explanation of the occurrence of multiply definitions of the `LastPage` label with lastpage, pagesLTS, hyperref package (in that order).
- Updated to (then) new version of undolabl package [2010/07/15] v1.0d, which uses `\undonewlabel` with only one instead of two arguments.
- Added a warning message, if hyperref *and* pdfpages were *both* used.

[2010/07/29 v1.1e **pagesLTS**]

- Removed lastpage209.sty, because it is now contained in the lastpage.`dtx` file, v $\geq$ 1.2a.
- Removed the `\@ifclassloaded{revtex4}` code for `\lastpage@putlabel`.
- Handling of lastpage package adapted to updated version 1.2(a).
- Corrected error in lastpage code [1994/07/20 v0.1b].
- Version handling for undolabl package updated.
- Included a `\CheckSum`.

[2010/08/08 v1.1f **pagesLTS**]

- Version 1.1e had a bug: AlphAlph was replaced by alphalph (because that package is named like this), but this was done also in commands and definitions - now reverted.

[2010/08/12 v1.1g pagesLTS]

- Now the rerun warning is given *after* e. g. the `\listfiles`, increasing the chance of the user to read it.

[2010/08/23 v1.1h pagesLTS]

- Renamed `\XXRoman` to `\XRoman`.
- Reduced the number of needed counters.
- Removed wrong % from the driver file.
- Changed the `\unit` definition (got rid of an old `\rm`).
- Without use of the hyperref package, labels of type `pagesLTS.`⟨*page numbering scheme*⟩`.local` became multiply defined. Now an `\undolabl` was inserted.

[2010/08/25 v1.1i pagesLTS]

- Bug fix: tcilatex defines the `\hyperref` command, therefore for hyperref package detection this had to be changed to `\Hy@Warning`.

[2010/09/12 v1.1j pagesLTS]

- Bug fix: LaTeX issued a "`Label(s) may have changed. Rerun to get cross-references right.`"-warning, even if labels had not changed but were overwritten.
- Starred version of `\lastpageref` for suppressing hyperlinks introduced.
- A lot of details.

[2010/09/22 v1.1k pagesLTS]

- When no `fnsymbol` pagenumbering scheme is used, the respective counters are not defined, saving three counters.
- Moved the package from `.../latex/muench/pagesLTS/...` to `.../latex/pagesLTS/...`.

[2010/09/27 v1.1l pagesLTS]

- Bug fix: `\PackageError{pagesLTS}{pagenumbering missing}{\pagesLTS@messageNPN }` had to be moved to the outside of `\EveryShipout`, because it wrote its message into the document instead to the screen and the `.log`-file.
- Updated to version 2010/04/24 v0.19 of the holtxdoc package.

[2011/02/01 v1.1m pagesLTS]

- Added a new warning subsection about hyperref and repeated page numbers.
- Bug fix: Missing % after `-\romannumeral\number-\arabic{#1}` added.
- The (then) new version v2.4i of the endfloat package was then even older than 15 years.
- Put a warning in the documentation as well as in the `log`-file and at the screen during compilation about the showkeys package. (The labels of the pagesLTS package cannot be shown by the showkeys package.)
- Bug fix: In some situations a rerun warning was given even if no rerun was necessary.
- The option `alphMult` is now set to `ab` by default.
- The option `AlphMulti` is now set to `AB` by default.

[2011/03/16 v1.1n pagesLTS]

- Bug fix: Handling of option `pagecontinue=false` changed. When `pagecontinue=false` was used, but also a `alphMult`, `AlphMulti`, `fnsymbolmult`, `romanMult` or `RomanMulti` option other than `0` or `false`, respectively, was used, the page numbering *was* continued/extended. Now a warning is issued in case of such option clash and `pagecontinue=false` is heeded, disabling all continuation.
- Bug fix: `\ProvidesPackage{pagesLTS}` contained an older date (2010/09/27 of v1.1l instead of 2011/02/01 of v1.1m).
- Bug fix: The ulem package is needed to generate the documentation from the `pagesLTS.dtx` file, but was not listed as necessary package.
- Bug fix: One reference to an outdated version of undolabl package, replaced by the (then) recent version.
- Some minor details.

[2011/03/17 v1.1o pagesLTS]

- Documentation and ReadMe bug fix: This pagesLTS package is located at https://ctan.org/pkg/pageslts instead of .../pagesLTS/.
- There is a new (possible) alternative package, totcount, see section 4.
- Bug fix: There was a reference to lastpage 1994/07/20, v0.1b, instead of the current version.

[2011/08/08 v1.2a]

- **Renamed the package from pagesLTS to pageslts (keeping family, prefix, internal commands,... as pagesLTS).** Added checking against double loading as pagesLTS and pageslts.
- The holtxdoc package was fixed (recent: 2011/02/04, v0.21), therefore the warning in `drv` could be removed.
- `\AtEndAfterFileList` from the (then) new version of the atveryend package, 2011/04/23, v1.7, by Heiko Oberdiek, is now used for the rerun hint instead of appending to `\@dofilelist`.
- Now defining 2: `\def\pagesLTS@two{2}` and 3: `\def\pagesLTS@three{3}`.
- Replaced `\texttt{\textbackslash...}` by `|\...|` in the `dtx` and by `\verb|\...|` in the example (where possible).
- When the alphalph package is needed, it is loaded via `\RequirePackage` instead of crashing with an error message.
- A lot of details (also in the documentation).

[2013/01/28 v1.2b]

- Updated to TeX live 2012 (for compiling the documentation and example) and installed the available updates. Therefore I can no longer test whether pageslts works with earlier versions of LaTeX.
- Replaced `\let` by `\LetLtxMacro`.
- The nameref package redefines `\label` to have five arguments instead of two, therefore `\newlabel{LastPage}{{}{\thepage}{}{}{}}` instead of `\newlabel{LastPage}{{}{\thepage}}` must be used. (Bug reported at https://tex.stackexchange.com/q/95541/6865, thanks to Michał Herman!) Fixed. [Later in the kernel *all* labels got defined to have five arguments.]
- Updates to a lot of details, also in the documentation.

[2014/01/19 v1.2c]

- Bug: missing loop, fix: inserted.
- Bug: when option `pagecontinue=false` was set, the extension of the page numbering schemes was disabled by mistake, fixed.
- Now using `\ltx@ifpackageloaded` from the ltxcmds package for checking (even after `\AtBeginDocument`) whether a package has been loaded.
- Bug: incompatibility with lineno because of a mistake in the redefined `\pagenumbering`, fixed. (Bug reported by kwikwi, thanks!)
- Bug: When `\pagenumbering` preceded `\frontmatter` on the same page but with different argument than the `\pagenumbering`, which was inside `\frontmatter`, then some labelling got mixed up. (Also this bug reported by kwikwi, thanks!) While this is not fixed automatically, now an appropriate error message is given. (Two different page numbering schemes on the same page make no sense anyway.)

[2015/08/02 v1.2d]

- Updated to TeX Live 2015 (for compiling the documentation and example) and installed the available updates. Therefore I can no longer test whether pageslts works with earlier versions of LaTeX.
- A `ifundefinedorrelax` similar to the one from scrbase.sty of the KOMA script bundle as 2013/12/19 v3.12 is used now, without the need for ε-TeX.
- New versions of Adobe Reader and of some packages have become available.
- Changed the message (type) to be displayed if writing to files is disallowed (as pointed out by Joshua Ellis, thanks!).
- Updates to several details, also in the documentation.

[2015/08/17 v1.2e]

- Bug fix, see https://tex.stackexchange.com/q/261445/6865, thank you to Prof. Enrico Gregorio for providing the fix and to Dr. Clea F. Rees for bringing this to my attention.

[2015/12/21 v1.2f]

- Replaced `\next` with `\@pageslts@currname`.
- Bug fix, see https://tex.stackexchange.com/q/140235/6865. If the bug had been actually reported (instead of waiting until I see the question), I would have fixed it earlier, of course.
- Changed minor details like fixing urls in the manual.

[2024-11-20 v2.0a]

- Kernel-updates broke the package, the package was repaired and updated to the new kernel-code; ports from lastpage.
- Removed `\unit`.
- Removed `\XXRoman` (deprecated at least since 2015).
- Removed code from/need for packages prelim2e, ltxcmds, atveryend, etoolbox, everyshi, letltxmacro, and rerunfilecheck.
- Removed checks for available ε-TeX, endfloats package from 1992, pagesLTS package from 2011, hyperref from 2015 etc.
- Renamed `\pagesLTS@ifcounter` to `\pagesLTS@providecounter`.
- Renamed `\pagesLTS.lastpage` to `\pagesLTSlastpage`.

[2024-12-07 v2.0b]

- Bug fix: `pagecontinue=false` did not reset everything always fully correct.
- Instead of requiring the kvoptions package, kernel commands are used.
- Now compatible with the romanbarpagenumber package, even independent of loading order.
- When more than one `\pagenumbering` command is given before the end of the first page (maybe in the preamble), all but the last `\pagenumbering` are ignored (as also LaTeX would naturally do).
- The saved `\OrigPagenumbering` is used in both places now instead of the plain definition from the kernel, so any change before the pageslts package is taken into account.
- Changed the section 4: Alternatives, mirroring the one from the current lastpage package.
- For cases when `\@currentHpage` is just `page.`, the warning text has been corrected.
- Page numbering scheme `silly` (from sillypage package) now works.
- Page numbering scheme `gobble` now works. (Is that silly?!)
- Added `\ProvideTextCommand{\textasteriskcentered}{PD1}{*}`.
- Missing `\if@filesw \fi` added.
- Recent undolabl package (≥ 2024-12-05 v1.0n) required!

[2024-12-23 v2.0c]

- Bug fix: Error handling for missing `\pagenumbering` now needs two `\expandafter`s.

[2025-01-10 2.0d]

- Bug fix: One `\Roman@bar{page}` must be `\Roman@bar{\Roman{page}}`.
- Several typing mistakes have been corrected, used format and package versions have been updated.

[2025-06-05 v2.0e]

- Since LaTeX-format 2025-06-01 after shipping the last page all further `\write`s are made `\immediate`, requiring some adjusted `\addtocounter{page}{-1}` and `\addtocounter{page}{+1}` here.

[2025-08-09 v2.0f]

- polyglossia with `arabic` changes the meaning of `\arabic`, therefore `\number\value` is needed to internally get compilable numbers.

[2025-08-14 v2.0g]

- Compatibility fix for package titlesec ported from lastpage.

When you find a mistake or have a suggestion for an improvement of this package, please send an e-mail to the maintainer, thanks! (Please see BUG REPORTS in the README.)

10 Index

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

R

T

U

V

W

X

Z