The lhelp Package*

Volker Kuhlmann†

2004/07/13

Abstract

This LaTeX2e package defines macros which are useful for many documents. It is a large collection of simple "little helpers" which do not really warrant a separate package on their own.

Included are, among other things, definitions of common units with preceeding thinspaces and optionally following space, framed boxes where both width and height can be specified, starting new odd or even pages, draft markers, notes, conditional includes, including EPS files, and versions of enumerate and itemize which allow the horizontal and vertical spacing to be changed.

Contents

*This file has version number v2.1, last modified 2004/07/13.

†Email: VolkerKuhlmann@GMX.de. For a postal address refer to the license section.

1 License

Volker Kuhlmann, c/o University of Canterbury, ELEC Dept, Creyke Road, Christchurch, New Zealand
E-Mail: `VolkerKuhlmann@GMX.de`

2 Introduction

This package is a collection of helpful short macros which are not related to each other.

Most of these macros are generally useful, but some will be useful to only a few people, if at all. They are combined here because any document I write uses at least some of them.

The first version of this package dates back to 1994, and combined bits and pieces from older LaTeX2.09 packages into this one. This package was made specifically for LaTeX 2_ε.

3 User Manual

3.1 Units

A selection of units is defined. Because the names are short and could easily clash with something else, units are only defined when the respective package option is used. All defined units have the thinspace between the number and the unit already included, so '`123\cm`' gives '123 cm'.

The units defined by unitBasic redefine existing control sequences.

A degree symbol and commands for degree celsius and fahrenheit are provided. Because `\degree` clashes with many thesis packages, lhelp will define both `\degree` and `\Degree` to be the degree symbol (superscript circle), if they are not defined already by the time lhelp is loaded. Either way, `\celsius` and `\fahren` will work.

`\muunit` `\muunit` produces just the micro (μ) part of the unit and includes the thinspace. If `\textmu` is defined (by e.g. textcomp), it will be used instead of `$\mu$`.

`unitxspace` With package option unitxspace, all defined units are followed by an `\xspace`, which automatically insert white space if the unit is not followed by punctuation. E.g. '`is 23\m long`' becomes 'is 23 m long' instead of 'is 23 mlong'. See the `xspace` package for details.

\celsius
\fahren

Package option unitbasic:

\g	g
\kg	kg
\mm	mm
\mum	μm
\cm	cm
\m	m
\ml, \mL	mL
\ns	ns
\mus	μs
\ms	ms
\s	s
\h	h
\muunit	μ

Package option unitBasic:

\l	L
\L	L
\min	min

Package option unittemp:

\degree	° (see text)
\Degree	° (see text)
\celsius	°C
\fahren	°F

Package option unitelec:

\muA	μA
\muH	μH
\muV	μV
\muW	μW
\ohm	Ω
\kohm	kΩ
\Mohm	MΩ
\ac	$_{\text{ac}}$
\dc	$_{\text{dc}}$
\rms	$_{\text{rms}}$
\Vac	V_{ac}
\Vdc	V_{dc}
\VLL	V_{LL}
\kVLL	V_{LL}

Package option units: all of the above.

3.2 Paragraph and page layout

page
emptypage
vmargin

Package option page sets a zero \parindent and a non-zero \parskip. Option emptypage also selects pagestyle empty. If package vmargin was loaded before package lhelp, both options also set the paper size to A4 and the margins to some more useful values.

3.3 Draft markers

draftmark

Package option draftmark selects pagestyle plain and changes the page footer to include the current date and the text "DRAFT". This works with the standard pagestyles empty (which is not empty then), plain, and headings. It obviously does not work with any custom pagestyles. This option does not affect the page header.

\draftname

\draftname produces the "DRAFT" text.

\draftfont

\draftfont switches to the font with which the draft text is printed. It takes one argument: the text to typeset in draft font.

draftmarkps
\putdraftmarkps

Same as option draftmark, but also writes a big "DRAFT" in gray across the page. Required are the color package, and either the graphics or graphicx package. Packages color and graphics are loaded if necessary. To load these packages with options, load them before lhelp. It works with page styles empty, plain, and headings. The page header is changed. This uses a PostScript font (font family phv), so may require a PostScript output device. The actual mark is placed with \putdraftmarkps, which may have to be redefined to accomodate a different \draftname. Its original definition can be found via the index in the code section of this manual.

draftmarkpsonly

Same as option draftmarkps, but only writes a big "DRAFT" in gray across the page. I.e., it doesn't change the footer.

3.4 Shortcuts and Symbols

refshortcuts

Package option refshortcuts defines several shortcuts for cross-referencing chapter, tables, etc. This is currently only really useful for English. Usage is the same as for \ref, except that "chapter" etc. is also printed. Those shortcuts starting with an uppercase letter print a word with an uppercase first letter.

For chapter, section, appendix, figure, table: \cref, \Cref, \sref, \Sref, \aref, \Aref, \fref, \Fref, \tref, \Tref

For pages: `\pgref`, `\Pgref`

For photos[1]: `\phref`, `\Phref`

Examples:

`\sref{s:mylabel}` produces `section~\ref{s:mylabel}`,
`\pgref{somepage}` produces `page~\pageref{somepage}`,
`\phref{p:photo1}` produces `photo~\ref{p:photo1}`,
`\Aref{s:a1}` produces `Appendix~\ref{s:a1}`.

`\lineout` `\ul` `\ulbf` Text with attributes. The text is taken as argument by these macros, and can't be broken across lines. If it is necessary to break text into lines, try Donald Arseneau's package ulem.

`\ulbf{`⟨*text*⟩`}`	**underlined bold**
`\ul{`⟨*text*⟩`}`	underlined
`\lineout{`⟨*text*⟩`}`	~~lined out~~

`\textsubscript` More shortcuts. LaTeXprovides a `\textsuperscript`, but no `\textsubscript`, so we make one here. `\textsubscript` is robust, and only defined by lhelp if it is not already defined, in case LaTeX does provide it in a future version.

`\bs`, `\texttt{\bs}`	\, \
`\careof`	℅ (the following space is included)
`\textsubscript{`⟨*subscript*⟩`}`	$\text{Text}_{\text{subscript}}$

These symbols produce the same result in text and math mode:

`\larr`	$\leftarrow$
`\rarr`	$\rightarrow$
`\PP`	$+$
`\MM`	$-$
`\PM`	$\pm$
`\about`	$\approx$

`\eg` `\ie` `\etc` These macros produce the English abbreviations "e.g.", "i.e.", and "etc.", including the following normal space. This is for those (like me) who always forget the backslash after the second period and thus get a sentence-ending space with `\frenchspacing`.

`\ca` `\resp` `\diameter` This is a bad excuse for a diameter symbol. When the AMS symbols are available (package amssymb), their `\varnothing` symbol will be used instead, which is an acceptable solution.

`\diameter`	ϕ
`\diameter`	$\varnothing$ (with package amssymb; else nothing shown here)

`\Discuss` `\Edit` `\Mark` Put some obvious markers into the text, as a reminder that changes to the text are necessary before publication.

`\Discuss`	*discuss*
`\Edit`	*edit*
`\Mark`	♣

3.5 Boxes and rules

`\fparbox` A framed `\parbox`. The text material is set in paragraph mode, not LR-mode. Unlike `\fbox{\parbox{\columnwidth}{...}}`, the width of the box generated by `\fparbox` including the frame is ⟨*width*⟩. The default for ⟨*width*⟩ is `\hsize`. `\fboxrule` and `\fboxsep` are used as for `\fbox`.

`\fparbox[`⟨*width*⟩`](`⟨*parbox-optargs*⟩`){`⟨*text*⟩`}`

⟨*parbox-optargs*⟩ are the optional arguments to `\parbox`, namely `[POS][HEIGHT][INNER-POS]`.

Examples:

```
\fparbox{text...}
\fparbox(){text...}
\fparbox([t]){text...}
\fparbox[50mm]([t][80mm][s]){text...}
```

[1] (e.g. with the photo package, also by Volker Kuhlmann, available from CTAN)

This box is produced with:
```
\fparbox{This box is produced with:\\\hspace*{7mm}\texttt{...}}
```

\xyfparbox Draws a framed box with ⟨*text*⟩ in it. ⟨*text*⟩ is set in a \parbox. The outside of the frame has a size of ⟨*width*⟩ times ⟨*height*⟩.

\xyfparbox[⟨*pos*⟩]{⟨*width*⟩}{⟨*height*⟩}{⟨*text*⟩}

The optional argument ⟨*pos*⟩ becomes the first optional argument to \parbox and defaults to c. \parbox's ⟨*inner-pos*⟩ is set to c. \fboxrule and \fboxsep are used as for \fbox.

Example:

```
\xyfparbox{60mm}{12mm}{...}
```

\tlap \blap Similar to plain's \llap and \rlap, the following overlapping boxes are provided. All of these boxes have a size of zero. Their content overlaps into adjacent page areas. Horizontal overlap uses \hbox, vertical overlap uses \vbox.

\lrlap	left and right (i.e. horizontally centred)
\tlap	top
\blap	bottom
\tblap	top and bottom (i.e. vertically centred)
\rtlap	right and top
\rblap	right and bottom

The following macros are useful for creating fill-in type forms.

\vnull In analogy to \null: an empty \vbox.

\vnul An empty \vbox right at the top edge of the page. This uses \offinterlineskip (and its own level of grouping).

\hrulenull An \hrule with zero dimensions.

3.6 Notes

printnotes Package option printnotes: print all notes in the document. The default is not to.

\notes{⟨*anytext*⟩}
\nnotes{⟨*anytext*⟩}

\notes \nnotes \notes prints ⟨*anytext*⟩ if package option printnotes was given, and discards ⟨*anytext*⟩ otherwise. \nnotes always discards ⟨*anytext*⟩; instead of deleting any notes you never want printed, rename \notes to \nnotes.

\notesfont If any notes are printed, they are printed in \notesfont.

\bnotemark \enotemark Printed notes are surrounded by a begin and an end mark which are printed into the margin. These marks are generated by \bnotemark and \enotemark.

\ifprintnotes The conditional used to decide whether notes are printed is \ifprintnotes. It can be used like \ifprintnotes ... \fi.

3.7 Conditional includes

The conditional including macros provided here use an "include level" to decide whether something is to be included or not. This is more powerful than a binary include switch, and for example allows all xfig figures to be included while no graphical images are. Graphical images are a common cause for exploding file sizes and much increased processing times. For larger documents the time saved while preparing can be considerable. For the final document this is then simply turned off.

excludelevel \includelower The counter excludelevel is the document's threshold for whether something is included or not. The default is 5.

\includelower{⟨*includelevel*⟩}{⟨*yes*⟩}{⟨*no*⟩}

`\includelower` expands to ⟨*yes*⟩ if ⟨*includelevel*⟩ is lower than the counter `excludelevel`, otherwise it expands to ⟨*no*⟩.

Example:

```
\documentclass ...
\usepackage{lhelp}
\setcounter{excludelevel}{2}
\begin{document}
\includelower{1}{\input{some-xfig-figure.latex}}{}
\includelower{2}{\includegraphics{some-image.eps}}{}
```

Will include the xfig figure but not the eps image.

Set `excludelevel` to a number larger than the largest ⟨*includelevel*⟩ used to include everything.

`\ifinclude` An alternative to `\includelower` is

`\ifinclude{`⟨*includelevel*⟩`}` ... `\else` ... `\fi`

`\ifinclude{`⟨*includelevel*⟩`}` expands to `\iftrue` if ⟨*includelevel*⟩ is lower than the counter `excludelevel`, and `\iffalse` otherwise.

3.8 Including EPS files

`\placeEPS` This macro helps with inserting EPS files into the document, reducing the amount of typing required for the common case. In draft mode (see package option epsdraft below) only an outline of the EPS is drawn. Missing files will not produce an error, only a warning is written on the screen.

`\placeEPS[`⟨*moreargs*⟩`]{`⟨*width*⟩`}{`⟨*height*⟩`}{`⟨*filename*⟩`}`

Inserts encapsulated postsript file ⟨*filename*⟩ into the text with width ⟨*width*⟩ and height ⟨*height*⟩. ⟨*width*⟩ and/or ⟨*height*⟩ may be empty; default is natural size. ⟨*moreargs*⟩ is anything that is allowed as argument to `\includegraphics` (graphicx version), e.g. bounding boxes—see the graphicx documentation.

The standard graphicx package is required, and must be loaded by the user (lhelp does not load it). Unfortunately, graphics is not sufficient, so that a document using `\placeEPS` must use the graphicx package.

If ⟨*filename*⟩ does not exist, an empty framed box with the given dimensions is drawn with the name of the file in it; if any of ⟨*width*⟩ and ⟨*height*⟩ are empty the default for ⟨*width*⟩ is `.8\hsize`, and for ⟨*height*⟩ is `40mm`.

In short, the advantages of `\placeEPS` over `\includegraphics` are:

- ...

Example:

```
\placeEPS[bb=20 20 500 500,width=6cm]{}{5cm}{figure1}
```

sets the bounding box to 20 20 500 500, and scales figure1.eps to a width of 6 cm and a height of 5 cm. The same could have been achieved with

```
\placeEPS[bb=20 20 500 500]{6cm}{5cm}{figure1}
```

`\addEPSopt`

`\addEPSopt{`⟨*options*⟩`}`

`\addEPSopt` adds the given comma-separated ⟨*options*⟩ to the internal option list for `\placeEPS`, so that these options do not need to be repeated with each `\placeEPS`. `\placeEPS` always passes the internal option list on to `\includegraphics`.

`epsdraft` With package option epsdraft, a draft keyword is inserted into every `\placeEPS` to get only an outline-box of the eps (same as package option draft for the graphics package).

`epspdf` `\EPSfileext` The package option epspdf sets `\EPSfileext` to `.eps`, or `.pdf` when running under pdflatex. `\EPSfileext` is appended to the filename by `\placeEPS`. This makes it possible to use `\placeEPS` in documents which are compiled with either latex or pdflatex. The default for `\EPSfileext` is empty, for compatibility with existing documents.

3.9 List environments and aides

The following macros extend the functionality of the existing list environments list, itemize, and enumerate.

First a few macros which set the spacing of a list environment. They are only useful inside the second argument of the list environment, which is used for setting various spacing-related variables.

`\listlabelleft` Sets the horizontal spacing, and the label raggedright.

`\listlabelleft{`⟨*labelindent*⟩`}{`⟨*labelwidth*⟩`}{`⟨*labelsep*⟩`}{`⟨*rightmargin*⟩`}`

⟨*labelindent*⟩ is the distance from the left edge of the surrounding text to the left edge of the label, ⟨*labelwidth*⟩ the width of the label, ⟨*labelsep*⟩ the distance between the label and the left edge of the list-text, and ⟨*rightmargin*⟩ the distance between the right edge of the list-text and the right edge of the surrounding text.

⟨*labelindent*⟩, ⟨*labelwidth*⟩, ⟨*labelsep*⟩, the width of the list-text and ⟨*rightmargin*⟩ added together make up `\columnwidth`. I find this much more user-friendly than the way LaTeX measures the horizontal dimensions in the list environment. `\listparindent` is set to zero. Currently, `\itemindent` is unchanged but LaTeX's default is zero.

`\listlabelleftindent` Some hard-coded values:

`\listlabelleftindent{`⟨*labelwidth*⟩`}`

Short for `\listlabelleft{1.5em}{`⟨*labelwidth*⟩`}{1.5em}{4.5em}`.

`\listshort` `\listshort` sets all vertical spacing to zero: `\topsep`, `\partopsep`, `\itemsep` and `\parsep`. Useful for lists which are not intended to stand out as prominently as LaTeX's default.

`Eenumerate` `Eitemize` Extend the enumerate and itemize environments such that the list spacing can be user-controlled.

`\begin{Eenumerate}{`⟨*formatting for list*⟩`}`
`\begin{Eitemize}{`⟨*formatting for list*⟩`}`

When enumerate and itemize call the list environment, ⟨*formatting for list*⟩ is inserted into the second argument of the list environment to allow changing any of the spacing. These can be mixed with enumerate and itemize.

`enumerateshort` `itemizeshort` As the enumerate and itemize environments, but with reduced vertical spacing.

`\begin{enumerateshort}`
`\begin{itemizeshort}`

These can be mixed with enumerate and itemize.

3.10 Starting new pages

This macros start new pages in a variety of ways.

`\newoddpage` `\newevenpage` `\newoddpage` starts a new odd page and `\newevenpage` starts a new even page, creating a blank page if necessary. The behaviour is always the same, regardless whether the document is double-sided (i.e. `\twoside` is in effect) or not.

`\clearoddpage` `\clearevenpage` `\clearoddpage` and `\clearevenpage` are the same as `\newoddpage` and `\newevenpage`, except they call `\clearpage` first, causing all unprocessed floats to be written out.

`\newoddpage*` `\newevenpage*` The star-form of these commands will always advance to the next odd/even page, creating blank pages if necessary, even if the current page is odd/even and empty. This means that they can be used repeatedly in immediate succession: `\newoddpage*\newoddpage*` starts an odd page and leaves another 2 blank pages.

`\clearthispage` `\clearthispage` writes out all unprocessed floats, and starts a new page regardless of whether the current page is empty.

Currently none of these macros allow to change the pagestyle of any blank pages which are generated (for example to empty when the document's pagestyle is plain).

There could be problems with page number references after `\new...`. It might pay to check `\pageref`'s after any of `\new...`.

3.11 One and two columns

\ensureonecolumn
\ensuretwocolumn
\ensurecolumnend

Ensure one or two columns for a part of a document, regardless whether the rest of the document is in one or two columns. Every `\ensureonecolumn`, or `\ensuretwocolumn` must be ended by `\ensurecolumnend`.

```
\ensureonecolumn
\ensuretwocolumn
\ensurecolumnend
```

If nesting is required, the environment form must be used:

```
... text in one or two columns ...
\begin{ensureonecolumn}
   ... text in one column ...
   \begin{ensuretwocolumn}
   ... text in two columns ...
   \end{ensuretwocolumn}
   ... text back in one column ...
\end{ensureonecolumn}
... text in one or two columns again ...
```

3.12 Hanging indentation

\hanghere

The macro `\hanghere` causes a hanging indentation for the rest of the paragraph, from the actual horizontal position of the `\hanghere`.

NOTE: The code for `\absval` and `\hangindent` is not covered by the copyright of the lhelp package, and, as published by its author, remains without copyright.

Here is the documentation of `\hanghere`, as published on the newsgroup `comp.text.tex`:

```
-- Donald Arseneau (1993)  (Not copyright, not supported)
   asnd@reg.triumf.ca

This is a style file that can be used in both LaTeX and plain TeX.
To use, put \hanghere in the middle of a paragraph and the rest of
            the paragraph will be indented to the spot so indicated.
            Spaces are retained on both sides of \hanghere, but if
                                                 you want a space
                                                 afterwards, you
                                                 should type \hanghere\
or \hanghere{} .  There is a length parameter, \minlinelen, that sets
   a minimum length for the lines of text.  If the position of \hanghere
would cause the line length to be too small, the following lines will
start at the left margin, as illustrated twice just above. Multiple
uses of \hanghere in a paragraph are cumulative, until the line length
        gets too short and reverts to the full text width.  There are
        no parboxes used so line spacing and page breaking is normal.
        There is one problem: if some text on the first line following
        the \hanghere is very tall, it can overlap the text above
            instead of doing the line spacing properly.  Other instances
            of tall text work fine.
```

\absval

The `\absval` macro is used by `\hanghere`, but is generally useful. It returns the "absolute value of a number or a dimension (if in a dimension register)".

\labelhangindent

Hanging indentation with the width of the printed label.

`\labelhangindent{`⟨*label*⟩`}`

This prints ⟨*label*⟩ and starts a hanging indentation. The hanging indentation remains for all paragraphs in the current group. Because this does not insert any horizontal space after the label, it is probably a good idea to insert a `\␣` as part of the label. `\labelhangindent` uses `\everypar`. So far, `\labelhangindent` is not cumulative, i.e. only one can be used in a paragraph.

3.13 Misc

\gobble
\gobbletwo

These macros simply discard their argument(s).

```
\gobble{⟨anything⟩}
\gobbletwo{⟨anything⟩}{⟨anythingelse⟩}
```

ddmonthyyyy
\ddmonthyyyy

Package option ddmonthyyyy switches \today's date format to a more user-friendly (and non-American) "dd Month yyyy". Regardless of what \today happens to be defined as, \ddmonthyyyy gives a format of 15 July 2004.

yyyymmdd
\yyyymmdd

Package option yyyymmdd switches \today's date format to "yyyy/mm/dd". Regardless of what \today happens to be defined as, \yyyymmdd gives a format of 2004/7/15.

hh:mm
\timehhmm
\todayaddtime

The current time is put into \timehhmm formatted as HH:MM with leading 0. Package option hh:mm appends \timehhmm to whatever \today is defined as, at the beginning of the document. Order of options is important—yyyymmdd for example redefines \today, so an appended time would be lost. \todayaddtime appends the time to the definition of \today, which is useful if something else has redefined \today.

morefontsizes

Package option morefontsizes defines the additional font sizes \HUGE, \veryhuge, \veryHuge, and \veryHUGE, which is sometimes useful for posters, or very large headings. It should look ok with PostScript fonts, and perhaps computer modern fonts. It is recommended to use this with the type-1 version of the computer modern fonts, or metafont might create some humungous bitmap fonts.

verbose

Package option verbose causes some commands to print some output which might be useful sometimes. Currently only \placeEPS makes use of it.

shorttoc

For documents with a zero \parindent and a non-zero \parskip, \tableofcontents generates fairly useless output. Package option shorttoc restores the previous behaviour for the table of contents.

countryselect
\selectD
\selectNZ
\selectUK
\selectUSA

These macros are meant to provide a standardised way for selecting country-specific settings, i.e. hyphenation patterns and specific language definitions. Theoretically, the babel package should offer this, but it doesn't—the name for a particular country depends on the name entered into babel's configuration file, for which there is no standard. Package option countryselect defines known names for selecting settings of a specific country. Because using the hyphenation patterns of a language is desirable even without the language-specific macro definitions, these commands now select the hyphenation patterns when available, even if babel is not used by the document. If babel is used, a \selectlanguage command is issued.

Currently only babel is supported as underlaying language-switching mechanism. Clearly this also calls for an lhelp configuration file, matching the entries in babel's.

So far, \selectD, \selectNZ, \selectUK, \selectUSA are defined for Germany, New Zealand, the UK, and the USA.

narrowpars

The narrowpars environment temporarily narrows the width of the text body, respectively increases the left and right margins.

```
\begin{narrowpars}{⟨indentation⟩}
```

Paragraphs are narrowed by ⟨*indentation*⟩ on the left and ⟨*indentation*⟩ on the right. It uses \everypar, \hangindent and \hangafter, and changes \columnwidth and \hsize.

It would have been possible to achieve the same effect by using \leftskip and \rightskip, but either will fail in some cases. Perhaps a narrowpars∗ environment should use these?

\thinthinspace

Approximately half a thinspace. A quarterspace?

\setTBstruts
\T
\B

In tables created with tabular and array which use horizontal lines, there is often too little space between the text of a line and the adjacent horizontal lines. \setTBstruts defines two macros, \T and \B, which fine-tune the vertical spacing on these lines. Example:

```
text ...\par
\setTBstruts
\begin{tabular}{ll}
    \hline\T
        cell$^2$ & cell\\
        cell$_3$ & cell \B\\
    \hline\T
        cell & cell \B\\
    \hline
\end{tabular}\par
text ...
```

With \T and \B:

text ...	
cell^2	cell
cell_3	cell
cell	cell
text ...	

Without:

text ...	
cell^2	cell
cell_3	cell
cell	cell
text ...	

(Taken from "TeX and TUG NEWS", Vol. 2, No. 3, 1993, p. 10.)

\placepos This macro can place anything at an arbitrary position onto the page, without shifting the current position (as much that is possible).

\placepos{⟨*right*⟩}{⟨*down*⟩}{⟨*text*⟩}

Places ⟨*text*⟩ (in LR mode) a distance of ⟨*right*⟩ to the right and ⟨*down*⟩ down from the current position. ⟨*right*⟩, ⟨*down*⟩ can be negative.

\placepos{..} has zero size and is equivalent to \hbox{}, which is the same as \mbox{} without the \leavevmode. This means that \placepos, unlike \mbox, will never start a paragraph. Although \placepos has zero size, placing an \hbox can have effects on spacing in some circumstances. If \placepos is at the beginning of a paragraph of text, it might be necessary to use a \leavevmode before the first \placepos.

To include paragraph material in ⟨*text*⟩, use \parbox{⟨*width*⟩}{⟨*text*⟩} as ⟨*text*⟩.

Spaces following \placepos are ignored.

\PSadjust Some older versions of the PostScript fonts had such a tight horizontal spacing that they could essentially not hyphenate a word. As a temporary solution it was suggested to change some line breaking parameters. Putting \PSadjust into the preamble will do this. Current versions of the PostScript fonts (from CTAN) don't need this.

4 To Do and Bugs

To do:

- \sref should become \aref when used after \appendix
- coloured boxes of specified sizes, e.g. \fcolorparbox
- Provide an upright Greek micro for units in roman text.

Any bugs? Please notify the author if you find any, or if the documentation is unclear. These ones were already found:

- draftmarkps causes two DRAFT warnings to be printed and logged to .log, ever since draftmarkpsonly was added

5 Implementation

1 ⟨*package⟩

5.1 Package options

5.1.1 Units

\lunitbox With option unitxspace, suffix all units with \xspace. Micro-units have a slightly reduced kerning with the math-mode \mu, which is used if \textmu is undefined.

```
2 \DeclareOption{unitxspace}{%
3   \def\lhelpxspace{\xspace}%
4 }
5 \newcommand\lunitbox[1]{\mbox{\,#1}\lhelpxspace}
6 \newcommand\luunitbox[1]{\mbox{\muunit #1}\lhelpxspace}
7 \newcommand\muunit{\@ifundefined{textmu}{\kern.05em\ensuremath{\mu}}{\,\textmu}}
```

```
\let\lhelpxspace\relax
```

\g \kg \mm \mum \cm \m \ml \mL \ns \mus \ms \s \h

Base units, plus a few more:

```
\DeclareOption{unitbasic}{%
  \newcommand\g{\,g\lhelpxspace}
  \newcommand\kg{\,kg\lhelpxspace}
  \newcommand\mm{\,mm\lhelpxspace}
  \newcommand\mum{\luunitbox{m}}% micro-metre
  \newcommand\cm{\,cm\lhelpxspace}
  \newcommand\m{\,m\lhelpxspace}
  \newcommand\ml{\mL\lhelpxspace}
  \newcommand\mL{\,mL\lhelpxspace}
  \newcommand\ns{\,ns\lhelpxspace}
  \newcommand\mus{\luunitbox{s}}% micro-seconds
  \newcommand\ms{\,ms\lhelpxspace}
  \newcommand\s{\,s\lhelpxspace}
  \newcommand\h{\,h\lhelpxspace}
}
```

These control sequences are already in use, but can be overridden with this option to give these units instead.

```
\DeclareOption{unitBasic}{%
  %\ExecuteOptions{unitbasic}
  \renewcommand\l{\,L\lhelpxspace}
  \renewcommand\L{\,L\lhelpxspace}
  \renewcommand\min{\,min\lhelpxspace}
}
```

\celsius \fahren

Temperature:

```
\DeclareOption{unittemp}{%
  \newcommand\degree@temperature{\ensuremath{^\circ}}
  \providecommand\degree{\degree@temperature\lhelpxspace}
  \providecommand\Degree{\degree@temperature\lhelpxspace}
  \newcommand\celsius{\mbox{\degree@temperature\kern-.05em C}\lhelpxspace}
  \newcommand\fahren{\mbox{\degree@temperature F}\lhelpxspace}
}
```

Electrical engineering: Use `\providecommand` for `\ohm` to allow precedence of other packages which also provide `\ohm`. Make `\kohm` and `\Mohm` fall back on `\ohm`.

```
\DeclareOption{unitelec}{%
  \newcommand\muA{\luunitbox{A}}
  \newcommand\muH{\luunitbox{H}}
  \newcommand\muV{\luunitbox{V}}
  \newcommand\muW{\luunitbox{W}}
  \providecommand\ohm{\lunitbox{$\Omega$}}
  \newcommand\kohm{\mbox{k}\ohm}
  \newcommand\Mohm{\mbox{M}\ohm}
  \newcommand\ac{\textsubscript{ac}\lhelpxspace}
  \newcommand\dc{\textsubscript{dc}\lhelpxspace}
  \newcommand\rms{\textsubscript{rms}\lhelpxspace}
  \newcommand\Vac{\lunitbox{V\kern-.16em\ac}}
  \newcommand\Vdc{\lunitbox{V\kern-.16em\dc}}
  \newcommand\VLL{\lunitbox{V\kern-.16em\textsubscript{LL}}}
  \newcommand\kVLL{\lunitbox{V\kern-.16em\textsubscript{LL}}}
}
```

All of the above units:

```
\DeclareOption{units}{%
  \ExecuteOptions{unitbasic,unitBasic,unittemp,unitelec}%
}
```

5.1.2 Paragraph layout (and page layout if vmargin is loaded)

```
\DeclareOption{page}{%
  \ifx\setpapersize\undefined\else
    \lhelp@paper
    \setmarginsrb{30mm}{20mm}{25mm}{10mm}{0pt}{0mm}{}{10mm}
  \fi
  \lhelp@pars
}
\DeclareOption{emptypage}{%
  \ifx\setpapersize\undefined
    \pagestyle{empty}%
  \else
    \lhelp@paper
    \setmargnohfrb{30mm}{20mm}{25mm}{10mm}
  \fi
  \lhelp@pars
}
\newcommand\lhelp@paper{%
  \typeout{Package lhelp: setting paper size and margins.}
  \setpapersize{A4}
}
\newcommand\lhelp@pars{%
  \columnsep 8mm
  \parskip 2ex \@plus0.5ex \@minus0.5ex
  \parindent \z@
}
```

5.1.3 Draft marks

Activate definitions and print notice to screen.

```
\newcommand\draftmark@select{%
  \PackageWarning{lhelp}{DRAFT mark selected}%
  \pagestyle{plain}%
}
```

```
\DeclareOption{draftmark}{%
  \newcommand\@draft@odd{\llap{\hbox{\draftfont{\today\ \ \draftname}}}}
  \newcommand\@draft@even{\rlap{\hbox{\draftfont{\draftname\ \ \today}}}}
  \newcommand\@psdraft@empty{%
    \def\@oddfoot{\reset@font\hfil\hfil\@draft@odd}%
    \def\@evenfoot{\reset@font\@draft@even\hfil\hfil}}
  \newcommand\@psdraft@plain{%
    \def\@oddfoot{\reset@font\hfil\thepage\hfil\@draft@odd}%
    \def\@evenfoot{\reset@font\@draft@even\hfil\thepage\hfil}}
%
  \add@toks\ps@empty\@psdraft@empty
  \add@toks\ps@plain\@psdraft@plain
  %\add@toks\ps@headings\@psdraft@plain
  %(should be odd foot of pagestyle empty, even foot of pagestyle plain)
```

`\DeclareOption` of LaTeX2e <1995/12/01> pl 2 does not handle macros which in themselves define new macros with options. Confusion about the number of "#" in the nested declarations. (Funnily, it works if option draftmark is called from within option draftmarkps.) Use alternative:

```
  \let\lhelp@o@ps@headings\ps@headings
  \def\ps@headings{\lhelp@o@ps@headings\@psdraft@plain}%
%
  \draftmark@select
}
```

`\add@toks` Add some tokens to the end of an existing control sequence. This could be good for other things too.

`\add@toks{`⟨*control sequence*⟩`}{`⟨*tokens to add*⟩`}`

```
\newtoks\lhelp@toks
\newcommand\add@toks[2]{%
  \expandafter\lhelp@toks\expandafter{#1}%
  \expandafter\def\expandafter#1\expandafter{\the\lhelp@toks #2}}
```

`\draftname` The "draft" text

```
\providecommand\draftname{DRAFT}
```

`\draftfont` The font with which the "DRAFT" text is printed

```
\newcommand\draftfont{\textsf}
```

`\draftmark@watermark` `\putdraftmarkps` PostScript draft mark. Changes the page header; can't use footer because this mark must be printed before (underneath) the text.

```
\newcommand\draftmark@watermark{%
  \newcommand\putdraftmarkps{%
    \placepos{.2\textwidth}{.8\textheight}{\rotatebox{65}{%
      \fontfamily{phv}\fontsize{.2\textheight}\z@\selectfont
      \color{draftgray}\draftname}}}
  \@ifundefined{rotatebox}{\AtEndOfPackage{\RequirePackage{graphics}}}{}
  \@ifundefined{color}{\AtEndOfPackage{\RequirePackage{color}}}{}
  \AtEndOfPackage{\definecolor{draftgray}{gray}{0.9}}  %0.955, 0.93
  \def\ps@plain{\let\@mkboth\@gobbletwo
    \def\@oddhead{\putdraftmarkps\hfil}\let\@evenhead\@oddhead}
  \let\ps@empty\ps@plain
  \let\ps@h@old\ps@headings \def\ps@headings{\ps@h@old
    \def\@oddhead{\putdraftmarkps{\slshape\rightmark}\hfil\thepage}%
    \def\@evenhead{\putdraftmarkps\thepage\hfil\slshape\leftmark}}%
  \let\draftmark@watermark\empty
%
  \draftmark@select
}
```

Package options for printing the postscript draft mark with and without the non-postscript mark.

```
\DeclareOption{draftmarkps}{%
  \draftmark@watermark
  \csname ds@draftmark\endcsname  % = \ExecuteOptions{draftmark}
}
\DeclareOption{draftmarkpsonly}{%
  \draftmark@watermark
}
```

`\EPSopt` Print only outlines for EPS included with `\placeEPS`.

```
\DeclareOption{epsdraft}{\def\EPSopt{draft}}
```

`\EPSfileext` Set `\EPSfileext` to `.eps`, or `.pdf` when running under pdflatex. `\EPSfileext` is appended to the filename by `\placeEPS`.

```
\DeclareOption{epspdf}{%
  \@ifundefined{pdfpagewidth}{\def\EPSfileext{.eps}}{\def\EPSfileext{.pdf}}}
```

5.1.4 Cross-referencing

For the lazy typist. `\phref` and `\Phref` may already have been defined by the photo package, therefore use `\providecommand`.

```
\DeclareOption{refshortcuts}{%
  \newcommand\cref{chapter~\ref}
  \newcommand\Cref{Chapter~\ref}
  \newcommand\sref{section~\ref}
  \newcommand\Sref{Section~\ref}
```

```
  \newcommand\aref{appendix~\ref}
  \newcommand\Aref{Appendix~\ref}
  \newcommand\fref{figure~\ref}
  \newcommand\Fref{Figure~\ref}
  \newcommand\tref{table~\ref}
  \newcommand\Tref{Table~\ref}
  \newcommand\pgref{page~\pageref}
  \newcommand\Pgref{Page~\pageref}
  \providecommand\phref{photo~\ref}
  \providecommand\Phref{Photo~\ref}
}
```

5.1.5 Other

`\ifprintnotes` A simple conditional whether any notes are printed or not. Notes should be surrounded by `\ifprintnotes ... \fi`.

```
\DeclareOption{printnotes}{%
  \printnotestrue
}
\newif\ifprintnotes
```

`\ddmonthyyyy` `\month@english` Date format "dd Month yyyy".

```
\DeclareOption{ddmonthyyyy}{%
  \providecommand\month@english{%
    \ifcase \month \or January\or February\or March\or
    April\or May\or June\or July\or August\or September\or
    October\or November\or December\fi}
  \newcommand\ddmonthyyyy{\number\day\space
    \month@english\space \number\year}%
  \AtBeginDocument{\let\today=\ddmonthyyyy}
}
```

`\yyyymmdd` Date format "yyyy/mm/dd"

```
\DeclareOption{yyyymmdd}{%
  \AtBeginDocument{\let\today=\yyyymmdd}
}
\newcommand\yyyymmdd{\number\year/\number\month/\number\day}%
```

`\hour` `\minute` `\timehhmm` Current time of day: hours and minutes

```
\DeclareOption{hh:mm}{%
  \AtBeginDocument{\todayaddtime}%
}
\newcount\hour
\newcount\minute
\hour\time \divide\hour 60
\minute-\hour \multiply\minute 60\advance\minute\time
\edef\timehhmm{\ifnum\hour<10 0\fi\the\hour
  :\ifnum\minute<10 0\fi\the\minute}
\newcommand\todayaddtime{\edef\today{\today\,~\timehhmm}}
```

`\HUGE` `\veryhuge` `\veryHuge` `\veryHUGE` Additional larger font sizes. This is mainly for PostScript fonts, or perhaps the type1 versions of the computer modern fonts.

```
\DeclareOption{morefontsizes}{%
  \newcommand\HUGE    {\@setfontsize\HUGE{29.86}{36}}
  \newcommand\veryhuge{\@setfontsize\veryhuge{35.83}{43}}
  \newcommand\veryHuge{\@setfontsize\veryHuge{43}{52}}
  \newcommand\veryHUGE{\@setfontsize\veryHUGE{51.6}{62}}
}
```

Print progress/debugging info in some places (default no)

```
\DeclareOption{verbose}{\lh@verbosetrue}
\newif\iflh@verbose
```

Shorten the toc (for when `\parskip` is non-zero)

```
\DeclareOption{shorttoc}{%
  \let\old@tableofcontents=\tableofcontents
  \def\tableofcontents{{%
    \parskip \z@ \@plus 1pt \parindent \z@
    \let\old@addvspace=\addvspace
    \def\addvspace##1{\skip0=##1\relax\old@addvspace{.5\skip0}}%
    \old@tableofcontents\par
  }}%
}
```

`\selectD` `\selectNZ` `\selectUK` `\selectUSA`

Selecting country specifics in a standard way. The definition of these macros depends on the underlying format, resp. particular package used to select languages (e.g. babel with file language.dat, german). (Perhaps this should go in lhelp.cfg?) If the hyphenation patterns for these languages are loaded, they will be selected even is babel is otherwise unused.

```
\DeclareOption{countryselect}{%
  \newcommand\selectD{\lhelp@lang@sel{german}}
  \newcommand\selectNZ{\lhelp@lang@sel{UKenglish}}
  \newcommand\selectUK{\lhelp@lang@sel{UKenglish}}
  \newcommand\selectUSA{\lhelp@lang@sel{USenglish}}
}
\newcommand\lhelp@lang@sel[1]{
  \expandafter\ifx\csname date#1\endcsname\relax
    \PackageWarning{lhelp}{Language '#1' not loaded,
      selecting hyphenation only}%
    \@ifundefined{l@#1}%
      {\PackageWarning{lhelp}{Hyphenation patterns for '#1' unavailable}}%
      {\language=\csname l@#1\endcsname}%
  \else
     \selectlanguage{#1}%
  \fi
}
```

5.1.6 lhelp extension package

Does not exist yet but hey, we plan ahead...

```
\DeclareOption{X}{\AtEndOfFile{\RequirePackage{lhelpx}}}
```

5.1.7 Process options

```
\ProcessOptions
```

5.2 Shortcuts and Symbols

`\textsubscript`

Provide a `\textsubscript` in case none is already defined.

```
\@ifundefined{textsubscript}{%
  \DeclareRobustCommand\textsubscript[1]{%
    \m@th\ensuremath{_{\mbox{\fontsize\sf@size\z@\selectfont #1}}}%
  }%
}{}
```

`\ulbf` `\ul` `\lineout`

Quick and dirty underline, underline bold, and lineout. For longer text use ulem.sty.

```
\newcommand\ul{\underline}
\newcommand\ulbf[1]{\underline{\textbf{#1}}}
\newcommand\lineout[1]{{\setbox0\hbox{#1}\rlap{\raise.4ex\hbox{%
  \vrule \@height.15ex \@width\wd0 \@depth 0pt}}\box0}}
```

\larr A left and a right arrow.
\rarr

```
\newcommand\larr{\mbox{$\leftarrow$}} % leftarrow
\newcommand\rarr{\mbox{$\rightarrow$}} % rightarrow
```

\bs A backslash. (by Donald Arseneau)

```
\newcommand\bs{\ifdim\fontdimen3\font=0pt\char`\\% (tt font)
  \else\ensuremath{\backslash}\fi}
```

\PP A math mode plus, minus, and plusminus.
\MM
\PM

```
\newcommand\PP{\ensuremath{+}}
\newcommand\MM{\ensuremath{-}}
\newcommand\PM{\ensuremath{\pm}}
```

\about An approximate sign which also works in text mode.

```
\newcommand\about{\mbox{$\approx$}}
```

\eg Abbreviations "e.g.", "i.e.", "etc.", "ca.", and "resp." with the following space included.
\ie Using \providecommand allows a previous differing definition.
\etc
\ca
\resp

```
\providecommand\eg{e.g.\ }
\providecommand\ie{i.e.\ }
\providecommand\etc{etc.\ }
\providecommand\ca{ca.\ }
\providecommand\resp{resp.\ }
```

\Discuss Place a visible sign that a text passage needs to be further discussed or edited.
\Edit

```
\newcommand\@edtext[1]{%
  {\slshape\footnotesize\fboxrule.4mm\fboxsep.4mm\fbox{#1}}}
\newcommand\Discuss{\protect\@edtext{discuss}}
\newcommand\Edit{\protect\@edtext{edit}}
```

\Mark Place a visible mark in the text to mark something which is not yet finished.

```
\newcommand\@showmark{\fbox{\small$\clubsuit$}}
\newcommand\Mark{\protect\@showmark}
```

\diameter A diameter symbol. This is a bad cludge without AMS symbols.

```
\newcommand\diameter{\@ifundefined{varnothing}%
  {\mbox{\raise.15ex\hbox{o}\kern-.5em/}}%
  {\,\kern-0.07em\ensuremath{\varnothing}}}
```

\careof A careof symbol. The following space is already included.

```
\providecommand\careof{\mbox{\raise.5ex\hbox{c}\kern-.2em/\kern-.2emo~}}
```

5.3 Framed parboxes, other boxes, and rules

\fparbox Framed paragraph text.

```
\newcommand\fparbox[1][\hsize]{\@ifnextchar({\@fparbox{#1}}{\@fparbox{#1}()}}
\long\def\@fparbox#1(#2)#3{\mbox{\fbox
  {\dimen0=#1\advance\dimen0-2\fboxsep\advance\dimen0-2\fboxrule
  \parbox#2{\dimen0}{#3}}}}
```

\xyfparbox A framed box with both width and height specified.

```
\newcommand\xyfparbox[4][c]{\mbox{\fbox{%
  \dimen0=2\fboxrule\advance\dimen0 2\fboxsep
  \dimen1=#2\advance\dimen1 -\dimen0
  \dimen2=#3\advance\dimen2 -\dimen0
  \parbox[#1][\dimen2][c]{\dimen1}{#4}}}}
```

\lrlap Overlapping boxes. Similar to plain's \llap, \rlap. Left+right (i.e. centre), top,
\tlap bottom, top+bottom, right+top, right+bottom.
\blap
\tblap
\rtlap
\rblap

```
\newcommand\lrlap[1]{\hb@xt@\z@{\hss#1\hss}}
\newcommand\tlap[1]{\vbox to\z@{\vss#1}}
\newcommand\blap[1]{\vbox to\z@{#1\vss}}
\newcommand\tblap[1]{\vbox to\z@{\vss#1\vss}}
\newcommand\rtlap[1]{\rlap{\tlap{#1}}}
\newcommand\rblap[1]{\rlap{\blap{#1}}}
```

Empty boxes and rules:

\vnull Like \null, but with \vbox.

```
\providecommand\vnull{\vbox{}}
```

\vnul An empty \vbox right at the top edge of the paper.

```
\providecommand\vnul{{\offinterlineskip\vnull}}
```

\hrulenull An \hrule with zero dimensions.

```
\providecommand\hrulenull{\hrule\@width\z@\@height\z@\@depth\z@}
```

5.4 Notes and remarks

This must all be \long\def!

\notes Take one argument, and print it depending on a switch, or always discard it.
\nnotes

```
\newcommand\notes[1]{\ifprintnotes
  \begingroup\reset@font\notesfont
  \bnotemark
  \ignorespaces #1\relax
  \enotemark
  \endgroup
  \fi
}
\newcommand\nnotes[1]{}
```

\bnotemark The marks printed into the margin at the beginning and end of a note, and the font
\enotemark with which notes are printed.
\notesfont

```
\newcommand\bnotemark{%
  \leavevmode
  \vadjust{\vbox to\z@{\vss\llap
    {\raise1ex\hbox{\footnotesize$\bigtriangledown$}\ \ }}}%
}
\newcommand\enotemark{%
  \leavevmode
  \vadjust{\vbox to\z@{\vss\llap
    {\hbox{\footnotesize$\bigtriangleup$}\ \ }\vss}}%
}
\newcommand\notesfont{\footnotesize\sffamily}
```

5.5 Including figures, EPS files, etc

\includelower Similar to LaTeX's \@ifundefined{NAME}{TRUE}{FALSE}.

\ifinclude \ifinclude{NUMBER} translates into \iftrue or \iffalse.

```
\newcommand\includelower[3]{\ifinclude{#1}\def\@tempa{#2}\else
  \def\@tempa{#3}\fi\@tempa}
\newcommand\ifinclude[1]{\ifnum#1<\value{excludelevel}}
\newcounter{excludelevel}
\setcounter{excludelevel}{5}
```

\placeEPS Place an EPS file into the document.

```
\providecommand\EPSfileext{}
\newcommand\placeEPS[4][]{%
  \begingroup
  \edef\w{#2}\ifx\w\empty\def\w{\the\z@}\fi
  \edef\h{#3}\ifx\h\empty\def\h{\the\z@}\fi
  \edef\eps@fn{#4\EPSfileext}%
  \normalsize
  \IfFileExists{\eps@fn}{%
    %\epsfig{file={\eps@fn}\@more,width=\w,height=\h}%
    %  (obsolete - hangs if \w, \h are 0pt)
    \edef\opt{\EPSopt,#1}%
    \ifdim\w=\z@\else\edef\opt{\opt,width=\w}\fi
    \ifdim\h=\z@\else\edef\opt{\opt,height=\h}\fi
    \iflh@verbose\typeout{Loading EPS file: \opt\space(\eps@fn)}\fi
    \expandafter\includegraphics\expandafter[\opt]{\eps@fn}%
  }{%
    \typeout{EPS file not found: <\eps@fn>}%
    \ifdim\w=\z@\def\w{.8\hsize}\fi\ifdim\h=\z@\def\h{40mm}\fi
    \xyfparbox[b]\w\h{\texttt{\@Sanitize{#1 }\\
      \@Sanitize{\eps@fn}}}%
  }%
  \endgroup
}
```

\@Sanitize Thanks to Peter Schmitt A8131DAL@AWIUNI11.EDVZ.UniVie.AC.AT for the \@Sanitize trick. Note: If arg is empty "csnameendcsname" is printed. Note: Arg can not be dimen, skip, ...! (syntax error results)

```
\newcommand\@Sanitize[1]{{\escapechar=-1
  \expandafter\string\csname#1\endcsname}}
```

\EPSopt Parameters which will always be inserted into the optional argument of \includegraphics.

```
\newcommand\EPSopt{}
```

\addEPSopt Specify options to \includegraphics which are used for every \includegraphics.

```
\newcommand\addEPSopt[1]{%
  \edef\EPSopt{\EPSopt,#1}}
```

5.6 List environments and aides

\listlabelleft
\listlabelleftindent Set horizontal list spacing. They are meant to go into the second argument of a list environment.

```
\newcommand\listlabelleft[4]{%
  \leftmargin #1\labelwidth #2\labelsep #3\rightmargin #4
  \advance\leftmargin\labelwidth\advance\leftmargin\labelsep
  \def\makelabel##1{##1\hfil}%
  \listparindent\z@}
\newcommand\listlabelleftindent[1]{\listlabelleft{1.5em}{#1}{1.5em}{4.5em}}
```

\listshort All vertical spacing is set to zero.

```
\newcommand\listshort{\topsep\z@\partopsep\z@\itemsep\z@\parsep\z@}
```

Eenumerate
Eitemize Add an additional, required argument and insert it into the second argument of the list environment.

Would a form of \begin[formatting for list]{itemize} have been better? It has a reasonably high risk of conflicting with other packages which also redefine the enumerate and itemize environments.

Copied as is from latex.tex (25 Mar 92), '#1' added. Jun 2000: Modified to be the same as LaTeX 2_ε.

```
\newcommand\Eenumerate[1]{\ifnum \@enumdepth >\thr@@ \@toodeep\else
  \advance\@enumdepth \@ne
  \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
  \expandafter\list\csname label\@enumctr\endcsname
    {\usecounter\@enumctr
    \def\makelabel##1{\hss\llap{##1}}#1}%
  \fi}
\let\endEenumerate=\endlist
\newcommand\Eitemize[1]{\ifnum \@itemdepth >\thr@@ \@toodeep\else
  \advance\@itemdepth \@ne
  \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
  \expandafter\list\csname \@itemitem\endcsname
    {\def\makelabel##1{\hss\llap{##1}}#1}%
  \fi}
\let\endEitemize=\endlist
```

enumerateshort
itemizeshort

As enumerate, itemize but with reduced vertical spacing.

```
\newenvironment{enumerateshort}{\Eenumerate{\listshort}}{\endEenumerate}
\newenvironment{itemizeshort}{\Eitemize{\listshort}}{\endEitemize}
```

5.7 Starting new pages

\newoddpage
\newevenpage
\clearoddpage
\clearevenpage

Always start a new odd/even page, even if the document is not twoside. Works with both single and double column. `\clear...` also write out all leftover floats (as `\clearpage`, `\cleardoublepage`).

\newoddpage*
\newevenpage*
\clearthispage

`\newoddpage*`, `\newevenpage*` and `\clearthispage` can be used multiple times, e.g. `\newoddpage*\newoddpage*` starts an odd page and leaves another 2 blank pages; `\clearthispage\clearthispage` finishes the current page and leaves one more blank page. Note: careful with `\new...` after float pages! Check `\pageref`'s!

```
\newcommand\clearoddpage{\clearpage\@newodd}
\newcommand\clearevenpage{\clearpage\@neweven}
\newcommand\clearthispage{\null\clearpage}
\newcommand\newoddpage{\@ifstar{\null}{}\newpage\@newodd}
\newcommand\newevenpage{\@ifstar{\null}{}\newpage\@neweven}
\newcommand\@newodd{\ifodd\c@page\else
  \null\newpage\if@twocolumn\null\newpage\fi\fi}
\newcommand\@neweven{\ifodd\c@page
  \null\newpage\if@twocolumn\null\newpage\fi\fi}
```

5.8 One and two columns

\ensureonecolumn
\ensuretwocolumn
\ensurecolumnend

Defining `\endensure...` might make it work as nestable environment.

```
\newcommand\ensureonecolumn{%
  \if@twocolumn
    \onecolumn
    \def\ensurecolumnend{\twocolumn\let\ensurecolumnend\relax}%
  \fi
}
\def\endensureonecolumn{\ensurecolumnend}
\newcommand\ensuretwocolumn{%
  \if@twocolumn\else
    \twocolumn
    \def\ensurecolumnend{\onecolumn\let\ensurecolumnend\relax}%
  \fi
}
\def\endensuretwocolumn{\ensurecolumnend}
\newcommand\ensurecolumnend{}
```

5.9 Hanging indentation

The code for \hangindent is copied from the newsgroup comp.text.tex. It might not have been a good idea to incorporate it into lhelp instead of leaving it as a separate package, but in here it also contains a bugfix.

As noted in section 3.12, the lhelp-copyright does not cover \absval and \hanghere.

First some initial setup:

```
\newdimen \minlinelen
\minlinelen=\ifx\marginparwidth\Und@phined .2\hsize \else \marginparwidth \fi
\ifx \@tempdimb\Und@fyned \csname newdimen\endcsname \@tempdimb\fi
```

\absval Useful little macro: gives absolute value of a number or a dimension (if in a dimension register). Note that this makes use of TeX's confusing habit of expanding \if's within a number.

```
\def\absval#1{\ifnum#1<\z@ -\fi#1}
```

\hanghere The actual code for \hanghere:

```
\newcommand\hanghere{\leavevmode
\ifinner\else \begingroup
 \displaywidowpenalty\widowpenalty
 \skip@\lastskip \unskip\unpenalty \penalty\@M \hskip\skip@ \null
 $$% Need a display to measure previous width
 \lineskiplimit-999\p@ % so we get a baselineskip that we can cancel with:
 \abovedisplayskip-\baselineskip \abovedisplayshortskip-\baselineskip
 \belowdisplayskip\z@skip \belowdisplayshortskip\z@skip
 \halign{##\cr\noalign{\global\dimen@i\prevdepth}% get depth of line above
   \hbox{\vrule width\z@ depth\dimen@i }\cr}% preserve its depth
 \dimen@\hsize \advance\dimen@-\minlinelen
 \ifdim\absval\predisplaysize>\dimen@
   \global\dimen@i\z@  \else
%  use the width of the line above (\predisplaysize-2em):
   \global\dimen@i\predisplaysize \global\advance\dimen@i-2em
 \fi
 $$\endgraf
 \ifdim\dimen@i>\z@ % then back up a line
  \@tempdimb\prevdepth
  \prevdepth-999\p@ % make sure I get an exact \baselineskip
  \parskip-999\p@   % but cancel the extra space
  \advance\parskip-\baselineskip % cancel the \baselineskip
  \advance\parskip-\@tempdimb\relax
 \else
  \parskip\z@skip
 \fi \parindent\z@ \leavevmode
%% \@tempdimb does not get past the \endgroup!,
%% \vrule has to be before \endgroup.
%% blame Volker if this is not true.
 \vrule depth\@tempdimb width\z@
 \endgroup
 \hangindent\dimen@i \hangafter\z@
%% \vrule depth\@tempdimb width\z@ % see 4 lines above.
\fi}
```

\labelhangindent Hanging indentation with the width of the printed label.

```
\newcommand\labelhangindent[1]{\setbox\@tempboxa\hbox{#1}%
  \expandafter\everypar\expandafter=\expandafter{\expandafter%
    \hangindent\the\wd\@tempboxa\hangafter 0\relax}%
  \leavevmode\box\@tempboxa\hangafter 1\ignorespaces}
```

5.10 Etc

\gobble \gobbletwo These macros simply discard their argument(s).

```
\long\def\gobble#1{}
\long\def\gobbletwo#1#2{}
```

narrowpars Narrower paragraphs than the rest of the text. Perhaps it would have been better to set `\leftskip` and `\rightskip`; but both methods fail in some cases.

```
\newenvironment{narrowpars}[1]{%
  \everypar={\hangindent #1\hangafter 0\relax}%
  \advance\columnwidth-#1\relax
  \advance\hsize-#1\relax
}{%
  \par
}
```

\thinthinspace Approximately half a thinspace. `\,` = `\thinspace` = `\kern .16667em`

```
\providecommand\thinthinspace{\kern .08em\relax}
```

\setTBstruts Fine-tune vertical spacing in tabular and array. (Taken from "TeX and TUG NEWS", \T Vol. 2, No. 3, 1993, p. 10.)
\B

```
\newcommand\setTBstruts{\def\T{\rule{\z@}{2.6ex}}%
  \def\B{\rule[-1.2ex]{\z@}{\z@}}}
```

\placepos Place text at a given arbitrary position. Equivalent to `\mbox`, but does not contain a `\leavevmode`.

```
\newcommand\placepos[3]{\hbox to\z@{\kern#1
  \raisebox{-#2}[\z@][\z@]{#3}\hss}\ignorespaces}
```

\PSadjust This once used to be necessary when using the PS fonts with psnfss. It changes the line breaking parameters such that some breaking is possible. Current versions of psnfss don't need this.

```
\newcommand\PSadjust{
  \tolerance 800
  \emergencystretch 2em
  \doublehyphendemerits 5000
  \hfuzz 0pt
  \leftskip 0pt \@minus1pt
  \rightskip 0pt \@minus1pt
}
```

⟨/package⟩

Change History

Index

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