Displaying page layout variables

Kent McPherson a.o.

2014/10/28

This file is maintained by the \LaTeX{} Project team. Bug reports can be opened (category tools) at http://latex-project.org/bugs.html.

1 Introduction

This \LaTeX{} 2ε package is a reimplementation of layout.sty by Kent McPherson. It defines the command \texttt{\layout} which produces an overview of the layout of the current document. The command \texttt{\layout*} recomputes the values it uses to produce the overview.

The figure on the next page shows the output of the \texttt{\layout} command for this document.

2 The implementation

This package prints a figure to illustrate the layout that is implemented by the document class. In the figure several words appear. They are stored in control sequences to be able to select a different language.

\begin{verbatim}
1 \comment{package}
2 \DeclareOption{dutch}{%
3 \def\Headertext{\textit{Kopregel}}
4 \def\Bodytext{\textit{Broodtekst}}
5 \def\Footertext{\textit{Voetregel}}
6 \def\MarginNotestext{\textit{Marge\Notities}}
7 \def\oneinchtext{een inch}
8 \def\notshown{niet getoond}
9 %}
10 \DeclareOption{german}{%
11 \def\Headertext{\textit{Kopfzeile}}
12 \def\Bodytext{\textit{Haupttext}}
13 \def\Footertext{\textit{Fu\sszeile}}
14 \def\MarginNotestext{\textit{Rand-\notizen}}
15 \def\oneinchtext{ein Zoll}
16 \def\notshown{ohne Abbildung}
17 %}
18 \DeclareOption{ngerman}{\ExecuteOptions{german}}
\end{verbatim}

*Converted for \LaTeX{} 2ε by Johannes Braams and modified by Hideo Umeki*
1. one inch + \hoffset
2. one inch + \voffset
3. \oddsidemargin = 73pt
4. \topmargin = 17pt
5. \headheight = 12pt
6. \headsep = 25pt
7. \textheight = 598pt
8. \textwidth = 355pt
9. \marginparsep = 11pt
10. \marginparwidth = 126pt
11. \footskip = 30pt
   \marginparpush = 0pt (not shown)
   \hoffset = 0pt
   \voffset = 0pt
   \paperwidth = 597pt
   \paperheight = 845pt
This package has an option \texttt{verbose}. Using it will make the command \texttt{\layout} type some of the parameters on the terminal.
The normal behaviour of this package when showing the values of the parameters is to truncate them. However, if you want to see the real parameter values you can use the option `reals` to get that effect.

\def\lay@value{}
\DeclareOption{integers}{\renewcommand*{\lay@value}[2]{\expandafter\number\csname #1@#2\endcsname pt}}
\DeclareOption{reals}{\renewcommand*{\lay@value}[2]{\the\csname #2\endcsname}}

The default language is English, the default mode is `silent` and the default way of showing parameter values is to use integers.
\ExecuteOptions{english,silent,integers}
\ProcessOptions

\LayOutbs Define \LayOutbs to produce a backslash. We use a definition which also works with OT1 fonts.
\newcommand\LayOutbs{}
\chardef\LayOutbs'\\n\ConvertToCount This macro stores the value of a `length` register in a `count` register.
\def\ConvertToCount#1#2{\#1=#2\relax\divide\#1 by 65536}
The result of this is that the `count` register holds the value of the `length` register in points.

\SetToHalf Small macros used in computing positions.
\def\SetToHalf#1#2{#1=#2\relax\divide#1 by \tw@}

\SetToQuart Small macros used in identifying dimensions.
\def\SetToQuart#1#2{#1=#2\relax\divide#1 by 4}

\Identify A small macro used in identifying dimensions.
\def\Identify#1{\put(\PositionX,\PositionY){\circle{20}}\put(\PositionX,\PositionY){\makebox(0,0){\tiny #1}}}

\InsideHArrow This macro is used to produce two horizontal arrows inside a box. The argument gives the width of the box.
\def\InsideHArrow#1{\ArrowLength = #1\divide\ArrowLength by \tw@\advance\ArrowLength by -10\advance\PositionX by -10\ifnum\ArrowLength<\z@\put(\PositionX,\PositionY){\vector(1,0){-\ArrowLength}}\advance\PositionX by 20\put(\PositionX,\PositionY){\vector(-1,0){-\ArrowLength}}\else

{\LayOutbs Define \LayOutbs to produce a backslash. We use a definition which also works with OT1 fonts.
\newcommand\LayOutbs{}
\chardef\LayOutbs'\\n\ConvertToCount This macro stores the value of a `length` register in a `count` register.
\def\ConvertToCount#1#2{\#1=#2\relax\divide\#1 by 65536}
The result of this is that the `count` register holds the value of the `length` register in points.

\SetToHalf Small macros used in computing positions.
\def\SetToHalf#1#2{#1=#2\relax\divide#1 by \tw@}

\SetToQuart Small macros used in identifying dimensions.
\def\SetToQuart#1#2{#1=#2\relax\divide#1 by 4}

\Identify A small macro used in identifying dimensions.
\def\Identify#1{\put(\PositionX,\PositionY){\circle{20}}\put(\PositionX,\PositionY){\makebox(0,0){\tiny #1}}}

\InsideHArrow This macro is used to produce two horizontal arrows inside a box. The argument gives the width of the box.
\def\InsideHArrow#1{\ArrowLength = #1\divide\ArrowLength by \tw@\advance\ArrowLength by -10\advance\PositionX by -10\ifnum\ArrowLength<\z@\put(\PositionX,\PositionY){\vector(1,0){-\ArrowLength}}\advance\PositionX by 20\put(\PositionX,\PositionY){\vector(-1,0){-\ArrowLength}}\else
\InsideVArrow This macro is used to produce two vertical arrows inside a box. The argument gives the height of the box.
\OutsideHArrow This macro is used to produce two horizontal arrows to delimit a length. The first argument is the position for the right arrow, the second argument gives the length and the third specifies the length of the arrows.
\OutsideVArrow This macro is used to produce two vertical arrows to delimit a length. The first argument is the position for the lower arrow, the second argument gives the length and the third and fourth specify the lengths of the lower and upper arrow.
\Show Macro used in the table that shows the setting of the parameters.
\Type Macro used to show a setting of a parameter on the terminal.
\oneinch A constant, giving the length of an inch in points (approximately)
Because the overview of the layout is produced in a figure environment we need to allocate a number of counters that are used to store the values of various dimensions.

\texttt{\textbackslash cnt@paperwidth} The dimensions of the paper
\texttt{\textbackslash cnt@paperheight}

\texttt{\textbackslash cnt@hoffset \textbackslash cnt@voffset} the offsets,
\texttt{\textbackslash cnt@textheight \textbackslash cnt@textwidth} dimensions of the text area,
\texttt{\textbackslash cnt@topmargin \textbackslash cnt@oddsidemargin \textbackslash cnt@evensidemargin} margins,
\texttt{\textbackslash cnt@headheight \textbackslash cnt@headsep} dimensions of the running heads,
\texttt{\textbackslash cnt@marginparsep \textbackslash cnt@marginparwidth \textbackslash cnt@marginparpush} marginal paragraphs,
\texttt{\textbackslash cnt@footskip} the distance between the running footers and the text,
\texttt{\textbackslash fheight}

\texttt{\textbackslash ref@top} The position of the top of the ‘printable area’ is one inch below the top of the paper by default. The value of \texttt{\textbackslash ref@top} is relative to the lower left corner of the picture environment that will be used.
For the offsets,
\ref@hoffset and \ref@voffset values are added to the default offset of one inch.
\ref@hoffset=\cnt@hoffset \advance\cnt@hoffset by \oneinch
\ref@voffset=\cnt@voffset
\cnt@voffset is converted to be relative to the origin of the picture.
\ref@offset=\ref@top
\advance\cnt@voffset by \-\ref@voffset
\ref@head and the text areas, running heads,
\ref@body body of the text
\ref@foot and running footers.
\ref@margin These are different for even and odd pages, so they are computed by \layout.
\ref@marginwidth
\ref@marginpar The following are a number of scratch registers, used in the positioning of the various pieces of the picture.
\Interval
\ExtraYPos
\PositionX
\PositionY
\ArrowLength

All values that might change during the document are computed by calling the macro \lay@getvalues. By default this macro is executed at \begin{document}.
\def\lay@getvalues{%
\ConvertToCount\cnt@textheight\textheight
\ConvertToCount\cnt@textwidth\textwidth
\ConvertToCount\cnt@topmargin\topmargin
\ConvertToCount\cnt@oddsidemargin\oddsidemargin
\ConvertToCount\cnt@evensidemargin\evensidemargin
\ConvertToCount\cnt@headheight\headheight
\ConvertToCount\cnt@headsep\headsep
\ConvertToCount\cnt@marginparsep\marginparsep
\ConvertToCount\cnt@marginparwidth\marginparwidth
\ConvertToCount\cnt@marginparpush\marginparpush
\ConvertToCount\cnt@footskip\footskip
\ref@head=\ref@top
\advance\ref@head by \-\ref@voffset
\advance\ref@head by \-\cnt@topmargin
\advance\ref@head by \-\cnt@headheight
\ref@body=\ref@head
\advance\ref@body by \-\cnt@headsep
The command \layout makes the picture and table that display the current settings of the layout parameters.

\newcommand\layout{%
  \@ifstar{\lay@getvalues\lay@xlayout}{\lay@xlayout}%
}\def\lay@xlayout{%
  \lay@layout%
  \if@twoside\lay@layout\fi}%

\lay@layout The internal macro \lay@layout does all the dirty work.

\newcommand\lay@layout{%
  \thispagestyle{empty}
  \if@twoside\ifodd\count\z@% Here we deal with an odd page in the twosided case.
    \typeout{Two-sided document style, odd page.}%
    \ref@marginwidth=\cnt@oddsidemargin
    \ref@marginpar=\oneinch
    \advance\ref@marginpar by \ref@hoffset
    \advance\ref@marginpar by \cnt@oddsidemargin
    \ref@margin\ref@marginpar
    \if@reversemargin% Here we deal with an even page in the twosided case.
    \advance\ref@marginpar by \cnt@textwidth
    \advance\ref@marginpar by \cnt@marginparsep
  \else% Here we deal with an even page in the twosided case.
    \typeout{Two-sided document style, even page.}%
    \ref@marginwidth=\cnt@evensidemargin
    \ref@marginpar=\oneinch
    \advance\ref@marginpar by \ref@hoffset
    \advance\ref@marginpar by \cnt@evensidemargin
  \fi%
% So we compute \ref@marginwidth, \ref@marginpar and \ref@margin.
% \typeout{Two-sided document style, odd page.}%
  \if@twoside% Here we deal with an odd page in the twosided case.
    \typeout{Two-sided document style, odd page.}%
    \ref@marginwidth=\cnt@oddsidemargin
    \ref@marginpar=\oneinch
    \advance\ref@marginpar by \ref@hoffset
    \advance\ref@marginpar by \cnt@oddsidemargin
    \if@reversemargin% Here we deal with an even page in the twosided case.
    \advance\ref@marginpar by \cnt@textwidth
    \advance\ref@marginpar by \cnt@marginparsep
  \else% Here we deal with an even page in the twosided case.
    \typeout{Two-sided document style, even page.}%
    \ref@marginwidth=\cnt@evensidemargin
    \ref@marginpar=\oneinch
    \advance\ref@marginpar by \ref@hoffset
    \advance\ref@marginpar by \cnt@evensidemargin
  \fi%
Finally we the case for single sided printing.

\typeout{One-sided document style.}
\def\refmarginwidth{\oddsidemargin}
\def\refmarginpar{\oneinch}
\advance\refmarginpar by \hoffset
\advance\refmarginpar by \oddsidemargin
\if@reversemargin
  \advance\refmarginpar by -\marginparsep
  \advance\refmarginpar by -\marginparwidth
\else
  \advance\refmarginpar by \textwidth
  \advance\refmarginpar by \marginparsep
\fi
\fi

Now we begin the picture environment; dividing all the lengths by two is done by setting \unitlength to 0.5pt
\setlength{\unitlength}{.5pt}
\begin{picture}(\paperwidth,\paperheight)
  \centering
  \thicklines
  First we have the pagebox and reference lines,
  \put(0,0){\framebox(\paperwidth,\paperheight){\mbox{}}}
  \put(0,\voffset){\dashbox{10}(\paperwidth,0){\mbox{}}}
  \put(\hoffset,0){\dashbox{10}(0,\paperheight){\mbox{}}}
  then the header,
  \put(\margin,\head){% 
    \framebox(\textwidth,\headheight)%
    %{\footnotesize\Headertext}}
  the body of the text area,
  \put(\margin,\body){%
    \framebox(\textwidth,\textheight){\Bodytext}}
  the footer
  \put(\margin,\foot){%
    \framebox(\textwidth,\fheight){%\footnotesize\Footertext}}
  and the space for marginal notes.
  \put(\marginpar,\body){%
    \framebox(\marginparwidth,\textheight)%
    %{\footnotesize\shortstack{\MarginNotestext}}}
Then we start putting in ‘arrows’ to mark the various parameters. From here we use \thinlines.

\thinlines

\PositionX and \PositionY will be the coordinates of the center of the arrow displaying \textwidth.

\SetToHalf\PositionX\cnt@textwidth
\advance\PositionX by \ref@margin

The arrow should be a bit above the bottom of the ‘body box’.

\PositionY = \ref@body
\advance\PositionY by 50

An identifying number is put here, in a circle.

\Identify{8}

Then the arrow is drawn.

\InsideHArrow\cnt@textwidth

Now the \textheight

\SetToHalf\PositionY\cnt@textheight
\advance\PositionY by \ref@body

The x-position of the arrow is at 4/5 of the width of the ‘body box’.

\PositionX = \cnt@textwidth
\divide\PositionX by 5
\multiply\PositionX by 4
\advance\PositionX by \ref@margin

An identifying number is put here, in a circle.

\Identify{7}

\InsideVArrow\cnt@textheight

The \hoffset,

\PositionY = 50
\SetToHalf\PositionX\cnt@hoffset
\Identify{1}
\InsideHArrow\cnt@hoffset

The width of the margin.

\SetToQuart\PositionY\cnt@textheight
\advance\PositionY by \ref@body
\ifnum\ref@marginwidth > 0
\OutsideHArrow\ref@margin\ref@marginwidth{20}
\PositionX = \cnt@hoffset
\else
\OutsideHArrow\cnt@hoffset{-\ref@marginwidth}{20}
\PositionX = \ref@margin
\fi
\advance\PositionX by -30
\Identify{3}

the \marginparwidth,

\SetToQuart\PositionY\cnt@textheight
\advance\PositionY by \ref@body
This arrow has to be bit below the one for the `\oddsidemargin` or `\evensidemargin`.

\advance\PositionY by 30
\SetToHalf\PositionX\cnt@marginparwidth
\advance\PositionX by \ref@marginpar
\Identify{10}
\InsideHArrow\cnt@marginparwidth

The `\marginparsep`, this depends on single or double sided printing.

\advance\PositionY by 30
\if@twoside
Twosided mode, reversemargin:
\ifodd\count\z@
\OutsideHArrow\ref@margin\cnt@marginparsep{20}
\PositionX = \ref@margin
\else
\OutsideHArrow\ref@marginpar\cnt@marginparsep{20}
\PositionX = \ref@marginpar
\fi
\else
Single sided mode.
\if@reversemargin
\OutsideHArrow\ref@margin\cnt@marginparsep{20}
\PositionX = \ref@margin
\else
\OutsideHArrow\ref@marginpar\cnt@marginparsep{20}
\PositionX = \ref@marginpar
\fi
\fi
\fi
\if@twoside
Not reversemargin:
\ifodd\count\z@
\OutsideHArrow\ref@margin\cnt@marginparsep{20}
\PositionX = \ref@margin
\else
\OutsideHArrow\ref@marginpar\cnt@marginparsep{20}
\PositionX = \ref@marginpar
\fi
\else
\Single sided mode.
\if@reversemargin
\OutsideHArrow\ref@margin\cnt@marginparsep{20}
\PositionX = \ref@margin
\else
\OutsideHArrow\ref@marginpar\cnt@marginparsep{20}
\PositionX = \ref@marginpar
\fi
\fi
\fi
\advance\PositionX by -\cnt@marginparsep
\advance\PositionX by -30
\Identify{9}

Identify the `\footskip`. The arrow will be located on 1/8th of the `\textwidth`.
\PositionX = \cnt@textwidth
\divide\PositionX by 8
\advance\PositionX by \ref@margin
\OutsideVArrow\ref@foot\cnt@footskip{20}{20}
\PositionY = \ref@foot
\advance\PositionY by \cnt@footskip
\advance\PositionY by 30
\Identify{11}
Identify the \voffset. The arrow will be located a bit to the left of the edge of the paper.

\PositionX = \cnt@paperwidth
\advance\PositionX by -50
\PositionY = \cnt@paperheight
\ExtraYPos = \PositionY
\advance\ExtraYPos by -\cnt@voffset
\advance\PositionY by \cnt@voffset
\divide\PositionY by \tw@
\Identify{2}
\InsideVArrow\ExtraYPos

Identify \topmargin, \headheight and \headsep.

The arrows will be located on 1/8th of the \textwidth, with intervals of the same size, stored in \Interval.
\Interval = \cnt@textwidth
\divide\Interval by 8
\PositionX = \ref@margin
\advance\PositionX by \Interval
First the \topmargin. If \topmargin has a positive value, the arrow is upward.
Otherwise, it is downward. The number label is always placed at the base of the arrow.
\ifnum\cnt@topmargin > \z@
  \ExtraYPos = \ref@head
  \advance\ExtraYPos by \cnt@headheight
  \OutsideVArrow\ExtraYPos\cnt@topmargin{20}{20}
  \PositionY = \ExtraYPos
  \advance\PositionY by \cnt@topmargin
\else
  \ExtraYPos = \cnt@voffset
  \OutsideVArrow\ExtraYPos{-\cnt@topmargin}{20}{20}
  \PositionY = \ExtraYPos
  \advance\PositionY by -\cnt@topmargin
\fi
\advance\PositionX by 30
\Identify{4}
\advance\PositionX by \Interval

Then the \headheight
\OutsideVArrow\ref@head\cnt@headheight{20}{20}
\PositionY = \ref@head
\advance\PositionY by \cnt@headheight
\advance\PositionY by 30
\Identify{5}
\advance\PositionX by \Interval

and finally the \headsep
\ExtraYPos=\ref@body
\advance\ExtraYPos by \cnt@textheight
\OutsideVArrow\ExtraYPos\cnt@headsep{20}{20}
\PositionY = \ref@body
\advance\PositionY by \cnt@textheight
\advance\PositionY by -30
\Identify{6}
Here we can end the picture environment and insert a little space.
\end{picture}
\medskip

Below the picture we put a table to show the actual values of the parameters. Note that fractional points are truncated, i.e., 72.27pt is displayed as 72pt.

The table is typeset inside a box with a depth of 0 to always keep it on the same page as the picture.

\vtop to 0pt{\@minipagerestore\footnotesize\ttfamily
\begin{tabular}{@{}rl@{\hspace{20pt}}rl}
1 & \oneinchtext + \LayOutbs\texttt{hoffset} \\
2 & \oneinchtext + \LayOutbs\texttt{voffset} \\
3 & \if@twoside \\
\ifodd\count\z@ \Show{cnt}{oddsidemargin} \\
\else \Show{cnt}{evensidemargin} \\
\fi \\
\else \\
\Show{cnt}{oddsidemargin} \\
\fi & 2 & \Show{cnt}{topmargin} \\
3 & \Show{cnt}{headheight} & 4 & \Show{cnt}{headsep} \\
4 & \Show{cnt}{textheight} & 5 & \Show{cnt}{textwidth} \\
5 & \Show{cnt}{marginparsep} & 6 & \Show{cnt}{marginparwidth} \\
6 & \Show{cnt}{marginparpush} & 7 & \Show{cnt}{footskip} \\
7 & \Show{cnt}{paperwidth} & 8 & \Show{cnt}{paperheight} \\
& \Show{cnt}{hoffset} & & \Show{cnt}{voffset} \\
& \Show{cnt}{voffset} & & \Show{cnt}{voffset} \\
& \Show{cnt}{paperheight} & & \Show{cnt}{paperheight} \\
\end{tabular}\vss}

When the option verbose was used the following lines will show dimensions on the terminal.
\Type{ref}{hoffset}
\Type{ref}{voffset}
\Type{cnt}{textheight}
\Type{cnt}{textwidth}

Finally we start a new page.
\newpage

}