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
https://latex-project.org/bugs.html.

1 Introduction

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

The figure on the next page shows the output of the \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 (*package)
2 \DeclareOption{dutch}{%
3 \def\Headertext{Kopregel}
4 \def\Bodytext{Broodtekst}
5 \def\Footertext{Voetregel}
6 \def\MarginNotestext{Marge\ Notities}
7 \def\oneinchtext{een inch}
8 \def\notshown{niet getoond}
9 %}
10 \DeclareOption{german}{%
11 \def\Headertext{Kopfzeile}
12 \def\Bodytext{Haupttext}
13 \def\Footertext{Fu\sszeile}
14 \def\MarginNotestext{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 \textwidth = 355pt \hspace{2pt} 2 \paperheight = 845pt
3 \headheight = 12pt \hspace{2pt} 4 \textwidth = 597pt
4 \topmargin = 17pt \hspace{2pt} 5 \headsep = 25pt
5 \oddsidemargin = 73pt \hspace{2pt} 6 \headheight = 12pt
6 \topmargin = 17pt \hspace{2pt} 7 \headsep = 25pt
7 \textheight = 598pt \hspace{2pt} 8 \textwidth = 355pt
8 \textwidth = 355pt \hspace{2pt} 9 \headheight = 12pt
9 \topmargin = 17pt \hspace{2pt} 10 \headsep = 25pt
10 \headheight = 12pt \hspace{2pt} 11 \textwidth = 355pt
11 \textwidth = 355pt \hspace{2pt} 1 \textwidth = 355pt
This package has an option `verbose`. Using it will make the command `\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 \texttt{reals} to get that effect.

\begin{verbatim}
\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}}
\end{verbatim}

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

\begin{verbatim}
\ExecuteOptions{english,silent,integers}
\ProcessOptions
\end{verbatim}

Define \texttt{\LayOutbs} to produce a backslash. We use a definition which also works with OT1 fonts.

\begin{verbatim}
\newcommand{\LayOutbs}{}
\chardef\LayOutbs'\\end{verbatim}

This macro stores the value of a length register in a count register.

\begin{verbatim}
\def\ConvertToCount#1#2{\let\lay@value\relax
  \divide #1 by 65536}
\end{verbatim}

The result of this is that the count register holds the value of the length register in points.

Small macros used in computing positions.

\begin{verbatim}
\def\SetToHalf#1#2{#1=#2\relax\divide#1by\tw@
  \divide\PositionX by \tw@}
\def\SetToQuart#1#2{#1=#2\relax\divide#1by4}
\end{verbatim}

A small macro used in identifying dimensions.

\begin{verbatim}
\def\Identify#1{\put(\PositionX,\PositionY){
\circle{20}}
\put(\PositionX,\PositionY){\makebox(0,0){\tiny #1}}}
\end{verbatim}

This macro is used to produce two horizontal arrows inside a box. The argument gives the width of the box.

\begin{verbatim}
\def\InsideHArrow#1{{\let\lay@value\relax
  \ArrowLength = #1
  \divide\ArrowLength by \tw@
  \divide\PositionX by \tw@
  \ifnum\ArrowLength<\z@
    \put(\PositionX,\PositionY){\vector(1,0){\ArrowLength}}
    \advance\PositionX by 20
    \put(\PositionX,\PositionY){\vector(-1,0){\ArrowLength}}
  \else
    \put(\PositionX,\PositionY){\vector(1,0){\ArrowLength}}
    \advance\PositionX by -10
    \put(\PositionX,\PositionY){\vector(-1,0){\ArrowLength}}
  \fi}
\end{verbatim}
\InsideVArrow This macro is used to produce two vertical arrows inside a box. The argument gives the height of the box.
\def\InsideVArrow#1{{% 
\ArrowLength = #1 
\divide\ArrowLength by \tw@ 
\advance\ArrowLength by -10 
\advance\PositionY by -10 
\put(\PositionX,\PositionY){\vector(0,-1){\ArrowLength}} 
\advance\PositionY by 20 
\put(\PositionX,\PositionY){\vector(0,+1){\ArrowLength}} 
}}

\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.
\def\OutsideHArrow#1#2#3{{% 
\PositionX = #1 
\advance\PositionX by #3 
\put(\PositionX,\PositionY){\vector(-1,0){#3}} 
\PositionX = #1 \advance\PositionX-#2 
\advance\PositionX by -#3 
\put(\PositionX,\PositionY){\vector(+1,0){#3}} 
}}

\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.
\def\OutsideVArrow#1#2#3#4{{% 
\PositionY = #1 
\advance\PositionY by -#3 
\put(\PositionX,\PositionY){\vector(0,+1){#3}} 
\PositionY = #1 
\advance\PositionY#2 
\advance\PositionY#4 
\put(\PositionX,\PositionY){\vector(0,-1){#4}} 
}}

\Show Macro used in the table that shows the setting of the parameters.
\def\Show#1#2{\LayOutbs #2 = \lay@value{#1}{#2}}

\Type Macro used to show a setting of a parameter on the terminal.
\def\Type#1#2{% 
\LayOuttype{#2 = \lay@value{#1}{#2}}}
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.

\cnt@paperwidth The dimensions of the paper
\cnt@paperheight
\cnt@hoffset the offsets,
\cnt@voffset
\cnt@textheight dimensions of the text area,
\cnt@textwidth
\cnt@topmargin margins,
\cnt@oddsidemargin \cnt@evensidemargin
\cnt@headheight dimensions of the running heads,
\cnt@headsep
\cnt@marginparsep marginal paragraphs,
\cnt@marginparwidth \cnt@marginparpush
\cnt@footskip the distance between the running footers and the text,
and the height of the footers, which is needed here to display a box, but which isn’t used by \TeX.
\fheight
\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 \ref@top is relative to the lower left corner of the picture environment that will be used.
\fheight=12
\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 \ref@top is relative to the lower left corner of the picture environment that will be used.
\ref@top=\cnt@paperheight \advance\ref@top by \oneinch
For the offsets,
\ref@hoffset \newcount\ref@hoffset
\ref@voffset \newcount\ref@voffset

The \hoffset and \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@top=\cnt@voffset
\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
\ref@marginwidth
\ref@marginpar

These are different for even and odd pages, so they are computed by \layout.
\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

\ref@head=\ref@top
\advance\ref@head by -\ref@voffset
\advance\ref@head by -\cnt@topmargin
\advance\ref@head by -\cnt@headheight

\advance\ref@head by -\cnt@headsep

\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

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

Finally we the case for single sided printing.
\typeout{One-sided document \spaced{style}.} 
\ref@marginwidth=\cnt@oddsidemargin
\ref@marginpar=oneinch
\advance\ref@marginpar by \ref@offset
\advance\ref@marginpar by \cnt@oddsidemargin
\ref@margin=\ref@marginpar
\if@reversemargin
  \advance\ref@marginpar by -\cnt@marginparsep
  \advance\ref@marginpar by -\cnt@marginparwidth
\else
  \advance\ref@marginpar by \cnt@textwidth
  \advance\ref@marginpar by \cnt@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}(\cnt@paperwidth,\cnt@paperheight)
  \centering
  \thicklines

First we have the pagebox and reference lines,
\put(0,0){\framebox(\cnt@paperwidth,\cnt@paperheight){\mbox{}}}
\put(0,\cnt@voffset){\dashbox{10}(\cnt@paperwidth,0){\mbox{}}}
\put(\cnt@hoffset,0){\dashbox{10}(0,\cnt@paperheight){\mbox{}}}

then the header,
\put(\ref@margin,\ref@head){%
  \framebox(\cnt@textwidth,\cnt@headheight)%
  {\footnotesize\Headertext}}

the body of the text area,
\put(\ref@margin,\ref@body){%
  \framebox(\cnt@textwidth,\cnt@textheight){\Bodytext}}

the footer
\put(\ref@margin,\ref@foot){%
  \framebox(\cnt@textwidth,\fheight){\footnotesize\Footertext}}

and the space for marginal notes.
\put(\ref@marginpar,\ref@body){%
  \framebox(\cnt@marginparwidth,\cnt@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:
\if@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
Not reversemargin:
\ifodd\count\z@
\OutsideHArrow\ref@marginpar\cnt@marginparsep{20}
\PositionX = \ref@marginpar
\else
\OutsideHArrow\ref@margin\cnt@marginparsep{20}
\PositionX = \ref@margin
\fi
\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
\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.

\begin{verbatim}
\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
\end{verbatim}

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`.

\begin{verbatim}
\Interval = \cnt@textwidth
\divide\Interval by 8
\PositionX = \ref@margin
\advance\PositionX by \Interval
\fi
\else
\ExtraYPos = \cnt@voffset
\OutsideVArrow\ExtraYPos{-\cnt@topmargin}{20}{20}
\PositionY = \ExtraYPos
\advance\PositionY by -\cnt@topmargin
\fi
\advance\PositionY by 30
\Identify{4}
\advance\PositionX by \Interval
\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
\end{verbatim}

Then the `\headheight`

\begin{verbatim}
\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
\end{verbatim}

and finally the `\headsep`

\begin{verbatim}
\ExtraYPos=\ref@body
\advance\ExtraYPos\cnt@textheight
\OutsideVArrow\ExtraYPos\cnt@headsep{20}{20}
\PositionY = \ref@body
\advance\PositionY by \cnt@textheight
\advance\PositionY by -30
\Identify{6}
\end{verbatim}
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\footnotessize\ttfamily
\begin{tabular}{@{}rl@{\hspace{20pt}}rl}
1 & \oneinchtext\ + \LayOutbs\texttt{hoffset} & 2 & \oneinchtext\ + \LayOutbs\texttt{voffset} \\
3 & \if@twoside  & 4 & \Show{cnt}{topmargin} \\
 & \ifodd\count\z@  & 5 & \Show{cnt}{headheight} & 6 & \Show{cnt}{headsep} \\
 & \else & 7 & \Show{cnt}{textheight} & 8 & \Show{cnt}{textwidth} \\
 & \Show{cnt}{marginparsep}\&10\& & 9 & \Show{cnt}{marginparwidth} & 11 & \Show{cnt}{footskip} & 12 & \Show{cnt}{marginparpush} \\
 & \rlap{\notshown}\\
\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