1 Introduction

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

The figure on the next page shows the output of the \texttt{\textbackslash{}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}
\texttt{\textbackslash{}DeclareOption\{dutch\}\{}\%
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}Headertext\}}\{Kopregel\}\}
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}Bodytext\}}\{Broodtekst\}\}
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}Footertext\}}\{Voetregel\}\}
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}MarginNotestext\}}\{Marge\textbackslash{}\textbackslash{}Notities\}\}
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}oneinchtext\}}\{een inch\}\}
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}notshown\}}\{niet getoond\}\}
\texttt{\textbackslash{}\}\}
\texttt{\textbackslash{}\}
\texttt{\textbackslash{}DeclareOption\{german\}\{}\%
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}Headertext\}}\{Kopfzeile\}\}
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}Bodytext\}}\{Haupttext\}\}
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}Footertext\}}\{Fu\texttt{\{}sszeile\}\}
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}MarginNotestext\}}\{Rand\texttt{\{}n\texttt{otizen\}\}
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}oneinchtext\}}\{ein Zoll\}\}
  \texttt{\textbackslash{}def\texttt{\{}\texttt{\textbackslash{}notshown\}}\{ohne Abbildung\}\}
\texttt{\textbackslash{}\} \}
\texttt{\textbackslash{}\}
\texttt{\textbackslash{}DeclareOption\{ngerman\}\{\texttt{\textbackslash{}ExecuteOptions\{german\}}\} \}
\end{verbatim}

*Converted for \texttt{\textbackslash{}iptex} by Johannes Braams and modified by Hideo Umeki
layout package version v1.2c as of 2014/10/28

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  
12 \marginparpush = 0pt (not shown)  
13 \hoffset = 0pt  
14 \voffset = 0pt  
15 \paperwidth = 597pt  
16 \paperheight = 845pt
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 `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}'\\}

\ConvertToCount This macro stores the value of a `length` register in a `count` register.
\def{\ConvertToCount#1#2}{#1=#2\relax\divide#1by\tw@
divide #1 by 65536.
\def{\SetToHalf#1#2}{#1=#2\relax\divide#1by4}
\def{\SetToQuart#1#2}{#1=#2\relax\divide#1by2}

\Identify A small macro used in identifying dimensions.
\def{\Identify#1}{%\put{(\PositionX,\PositionY){\circle{20}}\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}{%\count\arrowlength=#1\divide\arrowlength by \tw@\advance\arrowlength by -10\advance\positionx by -10\arrowlength<\z@\put{\positionx,\positiony}{\vector(1,0){-\arrowlength}}\advance\positionx by 20\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.

\begin{verbatim}
\cnt@paperwidth \cnt@paperheight \cnt@hoffset \cnt@voffset \cnt@textheight \cnt@textwidth \cnt@topmargin \cnt@oddsidemargin \cnt@evensidemargin \cnt@headheight \cnt@headsep \cnt@marginparsep \cnt@marginparwidth \cnt@marginparpush \cnt@footskip
\end{verbatim}

...
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.
\cnt@voffset=\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.
\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

\lay@getvalues 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 command \layout makes the picture and table that display the current settings of the layout parameters.

The actions of \layout depend on the pagestyle.

Here we deal with an odd page in the twosided case.

So we compute \ref@marginwidth, \ref@marginpar and \ref@margin.

Here we deal with an even page in the twosided case.

So we compute \ref@marginwidth, \ref@marginpar and \ref@margin.
Finally we the case for single sided printing.

\typeout{One-sided document style.}
\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
\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.

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

\PositionX = \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

\PositionX = \SetToHalf\PositionX\cnt@textheight
\advance\PositionX 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.

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

Twosided mode, reversemargin:

Single sided mode.

Identify the \footskip. The arrow will be located on 1/8th of the \textwidth.
Identify the \texttt{\voffset}. The arrow will be located a bit to the left of the edge of the paper.

```latex
\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 \texttt{\topmargin}, \texttt{\headheight} and \texttt{\headsep}.

The arrows will be located on 1/8th of the \texttt{\textwidth}, with intervals of the same size, stored in \texttt{\Interval}.

```latex
\Interval = \cnt@textwidth
\divide\Interval by 8
\PositionX = \ref@margin
\advance\PositionX by \Interval
```

First the \texttt{\topmargin}. If \texttt{\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.

```latex
\ifnum\cnt@topmargin > \z@
  \ExtraYPos = \ref@head
  \advance\ExtraYPos\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\PositionY by 30
\Identify{4}
```

Then the \texttt{\headheight}

```latex
\OutsideVArrow\ref@head\cnt@headheight{20}{20}
\PositionY = \ref@head
\advance\PositionY by \cnt@headheight
\advance\PositionY by 30
\Identify{5}
```

and finally the \texttt{\headsep}

```latex
\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}
```
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 & 4 & \Show{cnt}{topmargin} \\
5 & \Show{cnt}{headheight} & 6 & \Show{cnt}{headsep} \\
7 & \Show{cnt}{textheight} & 8 & \Show{cnt}{textwidth} \\
9 & \Show{cnt}{marginparsep}\&\texttt{10k} & \Show{cnt}{marginparwidth} \\
11 & \Show{cnt}{footskip} & & \Show{cnt}{marginparpush} \\
& \rlap{\footnotesize\notshown}) & & \Show{ref}{hoffset} & & \Show{ref}{voffset} \\
& & \Show{cnt}{paperwidth} & & \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.
\neupage
\}