Displaying page layout variables

Kent McPherson a.o.*

2014/10/28

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.

```
 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{\ss}zeile}
14   \def\MarginNotestext{Rand-\\ notizen}
15   \def\oneinchtext{ein Zoll}
16   \def\notshown{ohne Abbildung}
17   }
18 \DeclareOption{ngerman}{\ExecuteOptions{german}}
```

---

*Converted for LaTeX 2ε by Johannes Braams and modified by Hideo Umeki

Header

Margin
Notes

Body

Footer
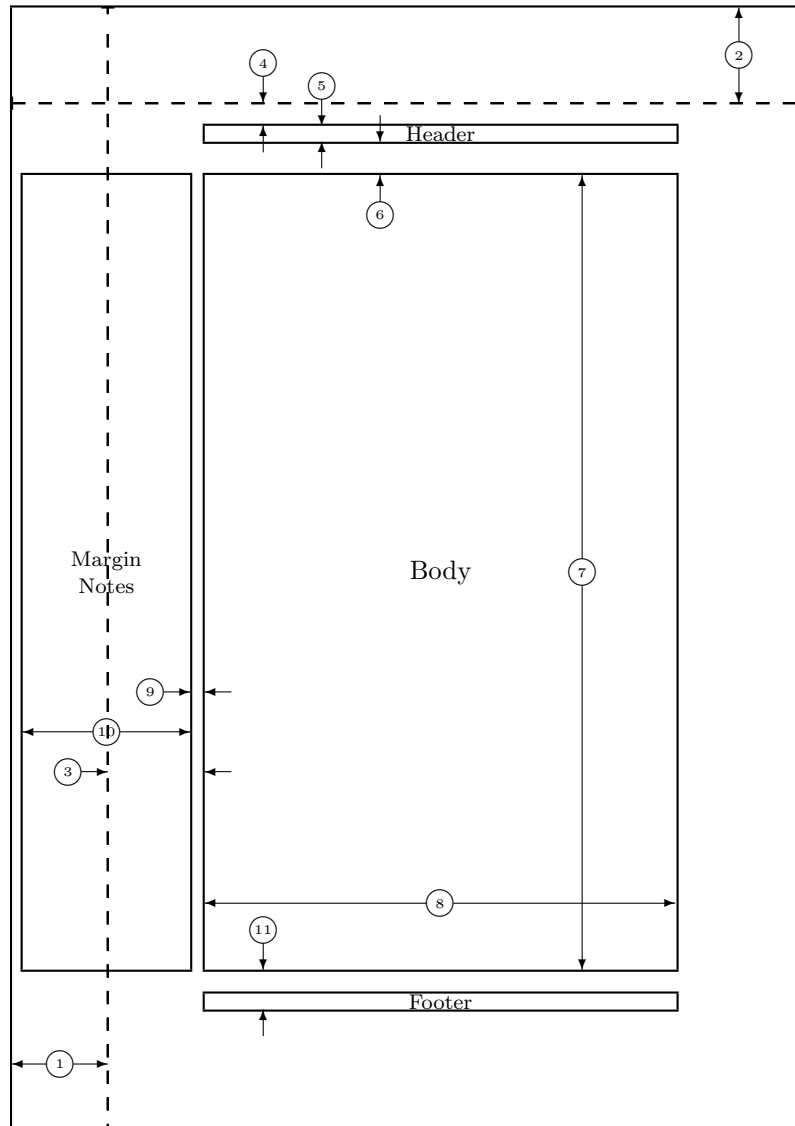
1	one inch + \hoffset	2	one inch + \voffset
3	\oddsidemargin = 73pt	4	\topmargin = 17pt
5	\headheight = 12pt	6	\headsep = 25pt
7	\textheight = 598pt	8	\textwidth = 355pt
9	\marginparsep = 11pt	10	\marginparwidth = 126pt
11	\footskip = 30pt		\marginparpush = 0pt (not shown)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 597pt		\paperheight = 845pt

```
19 \DeclareOption{english}{%
20   \def\Headertext{Header}
21   \def\Bodytext{Body}
22   \def\Footertext{Footer}
23   \def\MarginNotestext{Margin\\Notes}
24   \def\oneinchtext{one inch}
25   \def\notshown{not shown}
26   }
27 \DeclareOption{french}{%
28   \def\Headertext{Ent\^{e}te}
29   \def\Bodytext{Corps}
30   \def\Footertext{Pied de page}
31   \def\MarginNotestext{Marge\\Notes}
32   \def\oneinchtext{un pouce}
33   \def\notshown{non affich\'{e}}
34   }
35 \DeclareOption{francais}{\ExecuteOptions{french}}
36 \DeclareOption{spanish}{%
37   \def\Headertext{Encabezamiento}
38   \def\Bodytext{Cuerpo}
39   \def\Footertext{Pie de p\'agina}
40   \def\MarginNotestext{Notas\\ Marginales}
41   \def\oneinchtext{una pulgada}
42   \def\notshown{no mostradas}
43   }
44 \DeclareOption{portuguese}{%
45   \def\Headertext{Cabe\c{c}alho}
46   \def\Bodytext{Corpo}
47   \def\Footertext{Rodap\'e}
48   \def\MarginNotestext{Notas\\ Marginais}
49   \def\oneinchtext{uma polegada}
50   \def\notshown{n\~ao mostradas}
51   }
52 \DeclareOption{brazilian}{%
53   \def\Headertext{Cabe\c{c}alho}
54   \def\Bodytext{Corpo}
55   \def\Footertext{Rodap\'e}
56   \def\MarginNotestext{Notas\\ Marginais}
57   \def\oneinchtext{uma polegada}
58   \def\notshown{n\~ao mostradas}
59   }
60 \DeclareOption{italian}{%
61   \def\Headertext{Testatina}
62   \def\Bodytext{Corpo}
63   \def\Footertext{Piedino}
64   \def\MarginNotestext{Note\\ Marginali}
65   \def\oneinchtext{un pollice}
66   \def\notshown{non mostrato}
67   }
```

This package has an option verbose. Using it will make the command `\layout` type some of the parameters on the terminal.

```
68 \DeclareOption{verbose}{\let\LayOuttype\typeout}
69 \DeclareOption{silent}{\let\LayOuttype\@gobble}
```

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.

```
70 \def\lay@value{}
71 \DeclareOption{integers}{%
72   \renewcommand*{\lay@value}[2]{%
73     \expandafter\number\csname #1@#2\endcsname pt}}
74 \DeclareOption{reals}{%
75   \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.

```
76 \ExecuteOptions{english,silent,integers}
77 \ProcessOptions
```

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

```
78 \newcommand\LayOutbs{}
79 \chardef\LayOutbs`\\
```

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

```
80 \def\ConvertToCount#1#2{%
```

First copy the value

```
81   #1=#2
```

Then divide it by 65536.

```
82   \divide #1 by 65536}
```

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

\SetToHalf
\SetToQuart   Small macros used in computing positions.

```
83 \def\SetToHalf#1#2{#1=#2\relax\divide#1by\tw@}
84 \def\SetToQuart#1#2{#1=#2\relax\divide#1by4}
```

\Identify   A small macro used in identifying dimensions.

```
85 \def\Identify#1{%
86   \put(\PositionX,\PositionY){\circle{20}}
87   \put(\PositionX,\PositionY){\makebox(0,0){\tiny #1}}
88 }
```

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

```
89 \def\InsideHArrow#1{{%
90   \ArrowLength = #1
91   \divide\ArrowLength by \tw@
92   \advance\ArrowLength by -10
93   \advance\PositionX by -10
94   \ifnum\ArrowLength<\z@
95     \put(\PositionX,\PositionY){\vector(1,0){-\ArrowLength}}
96     \advance\PositionX by 20
97     \put(\PositionX,\PositionY){\vector(-1,0){-\ArrowLength}}
98   \else
```

```
99     \put(\PositionX,\PositionY){\vector(-1,0){\ArrowLength}}
100    \advance\PositionX by 20
101    \put(\PositionX,\PositionY){\vector(+1,0){\ArrowLength}}
102   \fi
103 }}
```

\InsideVArrow This macro is used to produce two vertical arrows inside a box. The argument gives the height of the box.

```
104 \def\InsideVArrow#1{{%
105   \ArrowLength = #1
106   \divide\ArrowLength by \tw@
107   \advance\ArrowLength by -10
108   \advance\PositionY by -10
109   \put(\PositionX,\PositionY){\vector(0,-1){\ArrowLength}}
110   \advance\PositionY by 20
111   \put(\PositionX,\PositionY){\vector(0,+1){\ArrowLength}}
112 }}
```

\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.

```
113 \def\OutsideHArrow#1#2#3{{%
114   \PositionX = #1
115   \advance\PositionX by #3
116   \put(\PositionX,\PositionY){\vector(-1,0){#3}}
117   \PositionX = #1 \advance\PositionX-#2
118   \advance\PositionX by -#3
119   \put(\PositionX,\PositionY){\vector(+1,0){#3}}
120 }}
```

\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 lenghts of the lower and upper arrow.

```
121 \def\OutsideVArrow#1#2#3#4{{%
122   \PositionY = #1
123   \advance\PositionY by -#3
124   \put(\PositionX,\PositionY){\vector(0,+1){#3}}
125   \PositionY = #1
126   \advance\PositionY#2
127   \advance\PositionY#4
128   \put(\PositionX,\PositionY){\vector(0,-1){#4}}
129 }}
```

\Show Macro used in the table that shows the setting of the parameters.

```
130 \def\Show#1#2{\LayOutbs #2 = \lay@value{#1}{#2}}
```

\Type Macro used to show a setting of a parameter on the terminal.

```
131 \def\Type#1#2{%
132   \LayOuttype{#2 = \lay@value{#1}{#2}}}
```

\oneinch A constant, giving the length of an inch in points (approximately)

```
133 \newcount\oneinch
134 \oneinch=72
```

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

```
135 \newcount\cnt@paperwidth
136 \newcount\cnt@paperheight
137 \ConvertToCount\cnt@paperwidth\paperwidth
138 \ConvertToCount\cnt@paperheight\paperheight
```

\cnt@hoffset the offsets,
\cnt@voffset

```
139 \newcount\cnt@hoffset
140 \newcount\cnt@voffset
141 \ConvertToCount\cnt@hoffset\hoffset
142 \ConvertToCount\cnt@voffset\voffset
```

\cnt@textheight dimensions of the text area,
\cnt@textwidth

```
143 \newcount\cnt@textheight
144 \newcount\cnt@textwidth
```

\cnt@topmargin margins,
\cnt@oddsidemargin
\cnt@evensidemargin

```
145 \newcount\cnt@topmargin
146 \newcount\cnt@oddsidemargin
147 \newcount\cnt@evensidemargin
```

\cnt@headheight dimensions of the running heads,
\cnt@headsep

```
148 \newcount\cnt@headheight
149 \newcount\cnt@headsep
```

\cnt@marginparsep marginal paragraphs,
\cnt@marginparwidth
\cnt@marginparpush

```
150 \newcount\cnt@marginparsep
151 \newcount\cnt@marginparwidth
152 \newcount\cnt@marginparpush
```

\cnt@footskip the distance between the running footers and the text,

```
153 \newcount\cnt@footskip
```

and the height of the footers, which is needed here to display a box, but which isn't used by LaTeX.

\fheight

```
154 \newcount\fheight
155 \fheight=12
```

Apart from integer representations of the page layout parameters we also need registers to store reference values in.

\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.

```
156 \newcount\ref@top
157 \ref@top=\cnt@paperheight \advance\ref@top by -\oneinch
```

\ref@hoffset   For the offsets,
\ref@voffset

```
158 \newcount\ref@hoffset
159 \newcount\ref@voffset
```

The \hoffset and \voffset values are added to the default offset of one inch.

```
160 \ref@hoffset=\cnt@hoffset  \advance\cnt@hoffset by \oneinch
161 \ref@voffset=\cnt@voffset
```

   \cnt@voffset is converted to be relative to the origin of the picture.

```
162 \cnt@voffset=\ref@top
163 \advance\cnt@voffset by -\ref@voffset
```

\ref@head   and the text areas, running heads,

```
164 \newcount\ref@head
```

\ref@body   body of the text

```
165 \newcount\ref@body
```

\ref@foot   and running footers.

```
166 \newcount\ref@foot
```

\ref@margin   These are different for even and odd pages, so they are computed by \layout.
\ref@marginwidth
\ref@marginpar

```
167 \newcount\ref@margin
168 \newcount\ref@marginwidth
169 \newcount\ref@marginpar
```

   The following are a number of scratch registers, used in the positioning of the various pices of the picture.

```
170 \newcount\Interval
171 \newcount\ExtraYPos
172 \newcount\PositionX
173 \newcount\PositionY
174 \newcount\ArrowLength
```

\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}.

```
175 \def\lay@getvalues{%
176   \ConvertToCount\cnt@textheight\textheight
177   \ConvertToCount\cnt@textwidth\textwidth
178   \ConvertToCount\cnt@topmargin\topmargin
179   \ConvertToCount\cnt@oddsidemargin\oddsidemargin
180   \ConvertToCount\cnt@evensidemargin\evensidemargin
181   \ConvertToCount\cnt@headheight\headheight
182   \ConvertToCount\cnt@headsep\headsep
183   \ConvertToCount\cnt@marginparsep\marginparsep
184   \ConvertToCount\cnt@marginparwidth\marginparwidth
185   \ConvertToCount\cnt@marginparpush\marginparpush
186   \ConvertToCount\cnt@footskip\footskip
187   \ref@head=\ref@top
188     \advance\ref@head by -\ref@voffset
189     \advance\ref@head by -\cnt@topmargin
190     \advance\ref@head by -\cnt@headheight
191   \ref@body=\ref@head
192     \advance\ref@body by -\cnt@headsep
```

```
193     \advance\ref@body by -\cnt@textheight
194   \ref@foot=\ref@body
195     \advance\ref@foot by -\cnt@footskip
196   }
197 \AtBeginDocument{\lay@getvalues}
```

\computevalues
\layout
\layout*

The command \layout makes the picture and table that display the current settings of the layout parameters.

```
198 \newcommand\layout{%
199   \@ifstar{\lay@getvalues\lay@xlayout}{\lay@xlayout}}
200 \def\lay@xlayout{%
201   \lay@layout
202   \if@twoside
203     \lay@layout
204   \fi}
```

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

```
205 \newcommand\lay@layout{%
206   \thispagestyle{empty}
```

The actions of \layout depend on the pagestyle.

```
207   \if@twoside
208     \ifodd\count\z@
```

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

```
209       \typeout{Two-sided document style, odd page.}
```

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

```
210       \ref@marginwidth=\cnt@oddsidemargin
211       \ref@marginpar=\oneinch
212       \advance\ref@marginpar by \ref@hoffset
213       \advance\ref@marginpar by \cnt@oddsidemargin
214       \ref@margin\ref@marginpar
215       \if@reversemargin
216         \advance\ref@marginpar by -\cnt@marginparsep
217         \advance\ref@marginpar by -\cnt@marginparwidth
218       \else
219         \advance\ref@marginpar by \cnt@textwidth
220         \advance\ref@marginpar by \cnt@marginparsep
221       \fi
222     \else
```

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

```
223   \typeout{Two-sided document style, even page.}
```

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

```
224       \ref@marginwidth=\cnt@evensidemargin
225       \ref@marginpar=\oneinch
226       \advance\ref@marginpar by \ref@hoffset
227       \advance\ref@marginpar by \cnt@evensidemargin
228       \ref@margin\ref@marginpar
229       \if@reversemargin
230         \advance\ref@marginpar by \cnt@textwidth
231         \advance\ref@marginpar by \cnt@marginparsep
```

```
232        \else
233          \advance\ref@marginpar by -\cnt@marginparsep
234          \advance\ref@marginpar by -\cnt@marginparwidth
235        \fi
236      \fi
237    \else
```

Finally we the case for single sided printing.

```
238      \typeout{One-sided document style.}
239      \ref@marginwidth=\cnt@oddsidemargin
240      \ref@marginpar=\oneinch
241      \advance\ref@marginpar by \ref@hoffset
242      \advance\ref@marginpar by \cnt@oddsidemargin
243      \ref@margin\ref@marginpar
244      \if@reversemargin
245        \advance\ref@marginpar by -\cnt@marginparsep
246        \advance\ref@marginpar by -\cnt@marginparwidth
247      \else
248        \advance\ref@marginpar by \cnt@textwidth
249        \advance\ref@marginpar by \cnt@marginparsep
250      \fi
251    \fi
```

Now we begin the picture environment; dividing all the lengths by two is done by setting `\unitlength` to `0.5pt`

```
252    \setlength{\unitlength}{.5pt}
253    \begin{picture}(\cnt@paperwidth,\cnt@paperheight)
254      \centering
255      \thicklines
```

First we have the pagebox and reference lines,

```
256      \put(0,0){\framebox(\cnt@paperwidth,\cnt@paperheight){\mbox{}}}
257      \put(0,\cnt@voffset){\dashbox{10}(\cnt@paperwidth,0){\mbox{}}}
258      \put(\cnt@hoffset,0){\dashbox{10}(0,\cnt@paperheight){\mbox{}}}
```

then the header,

```
259      \put(\ref@margin,\ref@head){%
260        \framebox(\cnt@textwidth,\cnt@headheight)%
261          {\footnotesize\Headertext}}
```

the body of the text area,

```
262      \put(\ref@margin,\ref@body){%
263        \framebox(\cnt@textwidth,\cnt@textheight){\Bodytext}}
```

the footer

```
264      \put(\ref@margin,\ref@foot){%
265        \framebox(\cnt@textwidth,\fheight){\footnotesize\Footertext}}
```

and the space for marginal notes.

```
266      \put(\ref@marginpar,\ref@body){%
267        \framebox(\cnt@marginparwidth,\cnt@textheight)%
268                {\footnotesize\shortstack{\MarginNotestext}}}
```

Then we start putting in 'arrows' to mark the various parameters. From here we use `\thinlines`.

269   `\thinlines`

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

270   `\SetToHalf\PositionX\cnt@textwidth`
271   `\advance\PositionX by \ref@margin`

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

272   `\PositionY = \ref@body`
273   `\advance\PositionY by 50`

An identifying number is put here, in a circle.

274   `\Identify{8}`

Then the arrow is drawn.

275   `\InsideHArrow\cnt@textwidth`

Now the `\textheight`

276   `\SetToHalf\PositionY\cnt@textheight`
277   `\advance\PositionY by \ref@body`

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

278   `\PositionX = \cnt@textwidth`
279   `\divide\PositionX by 5`
280   `\multiply \PositionX by 4`
281   `\advance\PositionX by \ref@margin`

An identifying number is put here, in a circle.

282   `\Identify{7}`
283   `\InsideVArrow\cnt@textheight`

The `\hoffset`,

284   `\PositionY = 50`
285   `\SetToHalf\PositionX\cnt@hoffset`
286   `\Identify{1}`
287   `\InsideHArrow\cnt@hoffset`

The width of the margin.

288   `\SetToQuart\PositionY\cnt@textheight`
289   `\advance\PositionY by \ref@body`
290   `\ifnum\ref@marginwidth > 0`
291   `\OutsideHArrow\ref@margin\ref@marginwidth{20}`
292   `\PositionX = \cnt@hoffset`
293   `\else`
294   `\OutsideHArrow\cnt@hoffset{-\ref@marginwidth}{20}`
295   `\PositionX = \ref@margin`
296   `\fi`
297   `\advance\PositionX by -30`
298   `\Identify{3}`

the `\marginparwidth`,

299   `\SetToQuart\PositionY\cnt@textheight`
300   `\advance\PositionY by \ref@body`

This arrow has to be bit below the one for the `\oddsidemargin` or `\evensidemargin`.

```
301     \advance\PositionY by 30
302     \SetToHalf\PositionX\cnt@marginparwidth
303     \advance\PositionX by \ref@marginpar
304     \Identify{10}
305     \InsideHArrow\cnt@marginparwidth
```

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

```
306     \advance\PositionY by 30
307     \if@twoside
```

Twosided mode, reversemargin;

```
308       \if@reversemargin
309         \ifodd\count\z@
310           \OutsideHArrow\ref@margin\cnt@marginparsep{20}
311           \PositionX = \ref@margin
312         \else
313           \OutsideHArrow\ref@marginpar\cnt@marginparsep{20}
314           \PositionX = \ref@marginpar
315         \fi
316       \else
```

Not reversemargin;

```
317       \ifodd\count\z@
318         \OutsideHArrow\ref@marginpar\cnt@marginparsep{20}
319         \PositionX = \ref@marginpar
320       \else
321         \OutsideHArrow\ref@margin\cnt@marginparsep{20}
322         \PositionX = \ref@margin
323       \fi
324     \fi
325   \else
```

Single sided mode.

```
326       \if@reversemargin
327         \OutsideHArrow\ref@margin\cnt@marginparsep{20}
328         \PositionX = \ref@margin
329       \else
330         \OutsideHArrow\ref@marginpar\cnt@marginparsep{20}
331         \PositionX = \ref@marginpar
332       \fi
333     \fi
334     \advance\PositionX by -\cnt@marginparsep
335     \advance\PositionX by -30
336     \Identify{9}
```

Identify the `\footskip`. The arrow will be located on 1/8th of the `\textwidth`.

```
337     \PositionX = \cnt@textwidth
338     \divide\PositionX by 8
339     \advance\PositionX by \ref@margin
340     \OutsideVArrow\ref@foot\cnt@footskip{20}{20}
341     \PositionY = \ref@foot
342     \advance\PositionY by \cnt@footskip
343     \advance\PositionY by 30
344     \Identify{11}
```

Identify the `\voffset`. The arrow will be located a bit to the left of the edge of the paper.

```
345    \PositionX = \cnt@paperwidth
346    \advance\PositionX by -50
347    \PositionY = \cnt@paperheight
348    \ExtraYPos = \PositionY
349    \advance\ExtraYPos by -\cnt@voffset
350    \advance\PositionY by \cnt@voffset
351    \divide\PositionY by \tw@
352    \Identify{2}
353    \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`.

```
354    \Interval = \cnt@textwidth
355    \divide\Interval by 8
356    \PositionX = \ref@margin
357    \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.

```
358    \ifnum\cnt@topmargin > \z@
359      \ExtraYPos = \ref@head
360      \advance\ExtraYPos\cnt@headheight
361      \OutsideVArrow\ExtraYPos\cnt@topmargin{20}{20}
362      \PositionY = \ExtraYPos
363      \advance\PositionY by \cnt@topmargin
364    \else
365      \ExtraYPos = \cnt@voffset
366      \OutsideVArrow\ExtraYPos{-\cnt@topmargin}{20}{20}
367      \PositionY = \ExtraYPos
368      \advance\PositionY by -\cnt@topmargin
369    \fi
370    \advance\PositionY by 30
371    \Identify{4}
372    \advance\PositionX by \Interval
```

Then the `\headheight`

```
373    \OutsideVArrow\ref@head\cnt@headheight{20}{20}
374    \PositionY = \ref@head
375    \advance\PositionY by \cnt@headheight
376    \advance\PositionY by 30
377    \Identify{5}
378    \advance\PositionX by \Interval
```

and finally the `\headsep`

```
379    \ExtraYPos=\ref@body
380    \advance\ExtraYPos\cnt@textheight
381    \OutsideVArrow\ExtraYPos\cnt@headsep{20}{20}
382    \PositionY = \ref@body
383    \advance\PositionY by \cnt@textheight
384    \advance\PositionY by -30
385    \Identify{6}
```

Here we can end the picture environment and insert a little space.

```
386    \end{picture}
387
388    \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.

```
389    \vtop to 0pt{%
390      \@minipagerestore\footnotesize\ttfamily
391      \begin{tabular}{@{}rl@{\hspace{20pt}}rl}
392        1 & \oneinchtext\ + \LayOutbs\texttt{hoffset}
393          & 2 & \oneinchtext\ + \LayOutbs\texttt{voffset} \\
394        3 & \if@twoside
395              \ifodd\count\z@ \Show{cnt}{oddsidemargin}
396              \else \Show{cnt}{evensidemargin}
397              \fi
398            \else
399              \Show{cnt}{oddsidemargin}
400            \fi                     & 4 & \Show{cnt}{topmargin} \\
401        5 & \Show{cnt}{headheight} & 6 & \Show{cnt}{headsep} \\
402        7 & \Show{cnt}{textheight} & 8 & \Show{cnt}{textwidth} \\
403        9 & \Show{cnt}{marginparsep}&10& \Show{cnt}{marginparwidth} \\
404        11& \Show{cnt}{footskip}   &   & \Show{cnt}{marginparpush}
405        \rlap{(\notshown)}\\
406          & \Show{ref}{hoffset}    &   & \Show{ref}{voffset} \\
407          & \Show{cnt}{paperwidth} &   & \Show{cnt}{paperheight} \\
408
409    \end{tabular}\vss}
```

When the option verbose was used the following lines will show dimensions on the terminal.

```
410    \Type{ref}{hoffset}
411    \Type{ref}{voffset}
412    \Type{cnt}{textheight}
413    \Type{cnt}{textwidth}
```

Finally we start a new page.

```
414    \newpage
415  }
416 ⟨/package⟩
```