

Displaying page layout variables

Kent McPherson a.o.*

2021-03-10

This file is maintained by the L^AT_EX Project team.
Bug reports can be opened (category `tools`) at
<https://latex-project.org/bugs.html>.

1 Introduction

This L^AT_EX 2_ε 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.

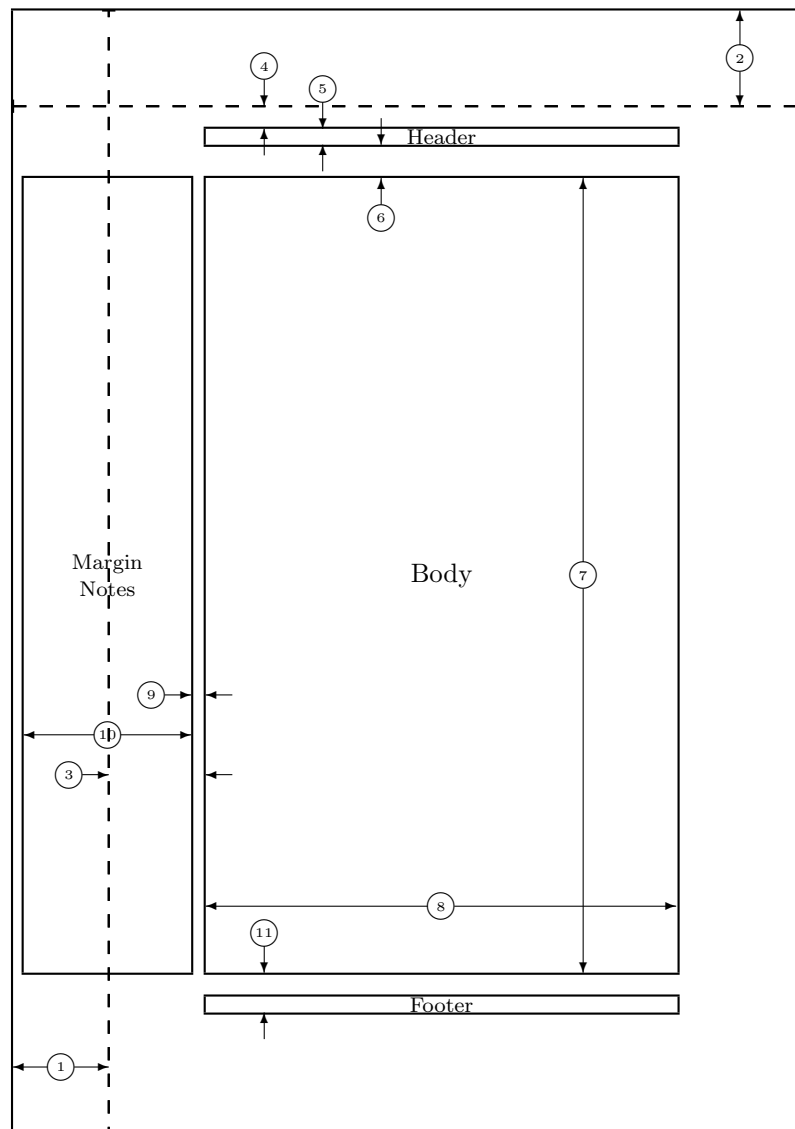
The figure on the next page shows the output of the `\layout` command for this document.

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 L^AT_EX 2_ε by Johannes Braams and modified by Hideo Umekei



- | | | | |
|----|-----------------------|----|----------------------------------|
| 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{français}{\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 }
68 \DeclareOption{romanian}{%
69   \def\Headertext{Antet}
70   \def\Bodytext{Corp}
71   \def\Footertext{Subsol}
72   \def\MarginNotestext{Note\\ Marginale}

```

```

73 \def\oneinchtext{un inch}
74 \def\notshown{neafi\textcommabelow sat}
75 }

76 \DeclareOption{japanese}{%
77 \def\Headertext{\square}
78 \def\Bodytext{\square\square\square}
79 \def\Footertext{\square}
80 \def\MarginNotestext{\square\backslash\square}
81 \def\oneinchtext{1\square\square}
82 \def\notshown{\square\square}
83 }

```

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

```

84 \DeclareOption{verbose}{\let\LayOuttype\typeout}
85 \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.

```

86 \def\lay@value{}
87 \DeclareOption{integers}{%
88 \renewcommand*\lay@value}[2]{%
89 \expandafter\number\csname #1@#2\endcsname pt}}
90 \DeclareOption{reals}{%
91 \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.

```

92 \ExecuteOptions{english,silent,integers}
93 \ProcessOptions

```

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

```

94 \newcommand\LayOutbs{}
95 \chardef\LayOutbs'\

```

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

```

96 \def\ConvertToCount#1#2{%

```

First copy the value

```

97 #1=#2

```

Then divide it by 65536.

```

98 \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.

```

\SetToQuart 99 \def\SetToHalf#1#2{#1=#2\relax\divide#1by\tw@}
100 \def\SetToQuart#1#2{#1=#2\relax\divide#1by4}

```

- \Identify** A small macro used in identifying dimensions.
- ```

101 \def\Identify#1{%
102 \put(\PositionX,\PositionY){\circle{20}}
103 \put(\PositionX,\PositionY){\makebox(0,0){\tiny #1}}
104 }

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

105 \def\InsideHArrow#1{%
106   \ArrowLength = #1
107   \divide\ArrowLength by \tw@
108   \advance\ArrowLength by -10
109   \advance\PositionX by -10
110   \ifnum\ArrowLength<\z@
111     \put(\PositionX,\PositionY){\vector(1,0){-\ArrowLength}}
112     \advance\PositionX by 20
113     \put(\PositionX,\PositionY){\vector(-1,0){-\ArrowLength}}
114   \else
115     \put(\PositionX,\PositionY){\vector(-1,0){\ArrowLength}}
116     \advance\PositionX by 20
117     \put(\PositionX,\PositionY){\vector(+1,0){\ArrowLength}}
118   \fi
119 }}

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

120 \def\InsideVArrow#1{%
121 \ArrowLength = #1
122 \divide\ArrowLength by \tw@
123 \advance\ArrowLength by -10
124 \advance\PositionY by -10
125 \put(\PositionX,\PositionY){\vector(0,-1){\ArrowLength}}
126 \advance\PositionY by 20
127 \put(\PositionX,\PositionY){\vector(0,+1){\ArrowLength}}
128 }}

```
- \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.
- ```

129 \def\OutsideHArrow#1#2#3{%
130   \PositionX = #1
131   \advance\PositionX by #3
132   \put(\PositionX,\PositionY){\vector(-1,0){#3}}
133   \PositionX = #1 \advance\PositionX-#2
134   \advance\PositionX by -#3
135   \put(\PositionX,\PositionY){\vector(+1,0){#3}}
136 }}

```
- \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.
- ```

137 \def\OutsideVArrow#1#2#3#4{%
138 \PositionY = #1

```

```

139 \advance\PositionY by -#3
140 \put(\PositionX,\PositionY){\vector(0,+1){#3}}
141 \PositionY = #1
142 \advance\PositionY#2
143 \advance\PositionY#4
144 \put(\PositionX,\PositionY){\vector(0,-1){#4}}
145 }}

```

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

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

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

```

147 \def\Type#1#2{%
148 \LayOuttype{#2 = \lay@value{#1}{#2}}

```

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

```

149 \newcount\oneinch
150 \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 151 \newcount\cnt@paperwidth
152 \newcount\cnt@paperheight
153 \ConvertToCount\cnt@paperwidth\paperwidth
154 \ConvertToCount\cnt@paperheight\paperheight

```

**\cnt@hoffset** the offsets,

```

\cnt@voffset 155 \newcount\cnt@hoffset
156 \newcount\cnt@voffset
157 \ConvertToCount\cnt@hoffset\hoffset
158 \ConvertToCount\cnt@voffset\voffset

```

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

```

\cnt@textwidth 159 \newcount\cnt@textheight
160 \newcount\cnt@textwidth

```

**\cnt@topmargin** margins,

```

\cnt@oddsidemargin 161 \newcount\cnt@topmargin
\cnt@evensidemargin 162 \newcount\cnt@oddsidemargin
163 \newcount\cnt@evensidemargin

```

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

```

\cnt@headsep 164 \newcount\cnt@headheight
165 \newcount\cnt@headsep

```

**\cnt@marginparsep** marginal paragraphs,

```

\cnt@marginparwidth 166 \newcount\cnt@marginparsep
\cnt@marginparpush 167 \newcount\cnt@marginparwidth
168 \newcount\cnt@marginparpush

```

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

```
169 \newcount\cnt@footskip
```

and the height of the footers, which is needed here to display a box, but which isn't used by L<sup>A</sup>T<sub>E</sub>X.

`\fheight`

```
170 \newcount\fheight
```

```
171 \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.

```
172 \newcount\ref@top
```

```
173 \ref@top=\cnt@paperheight \advance\ref@top by -\oneinch
```

`\ref@hoffset` For the offsets,

`\ref@voffset`

```
174 \newcount\ref@hoffset
```

```
175 \newcount\ref@voffset
```

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

```
176 \ref@hoffset=\cnt@hoffset \advance\cnt@hoffset by \oneinch
```

```
177 \ref@voffset=\cnt@voffset
```

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

```
178 \cnt@voffset=\ref@top
```

```
179 \advance\cnt@voffset by -\ref@voffset
```

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

```
180 \newcount\ref@head
```

`\ref@body` body of the text

```
181 \newcount\ref@body
```

`\ref@foot` and running footers.

```
182 \newcount\ref@foot
```

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

`\ref@marginwidth`

```
183 \newcount\ref@margin
```

`\ref@marginpar`

```
184 \newcount\ref@marginwidth
```

```
185 \newcount\ref@marginpar
```

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

```
186 \newcount\Interval
```

```
187 \newcount\ExtraYPos
```

```
188 \newcount\PositionX
```

```
189 \newcount\PositionY
```

```
190 \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}`.

```

191 \def\lay@getvalues{%
192 \ConvertToCount\cnt@textheight\textheight
193 \ConvertToCount\cnt@textwidth\textwidth
194 \ConvertToCount\cnt@topmargin\topmargin
195 \ConvertToCount\cnt@oddsidemargin\oddsidemargin
196 \ConvertToCount\cnt@evensidemargin\evensidemargin
197 \ConvertToCount\cnt@headheight\headheight
198 \ConvertToCount\cnt@headsep\headsep
199 \ConvertToCount\cnt@marginparsep\marginparsep
200 \ConvertToCount\cnt@marginparwidth\marginparwidth
201 \ConvertToCount\cnt@marginparpush\marginparpush
202 \ConvertToCount\cnt@footskip\footskip
203 \ref@head=\ref@top
204 \advance\ref@head by -\ref@voffset
205 \advance\ref@head by -\cnt@topmargin
206 \advance\ref@head by -\cnt@headheight
207 \ref@body=\ref@head
208 \advance\ref@body by -\cnt@headsep
209 \advance\ref@body by -\cnt@textheight
210 \ref@foot=\ref@body
211 \advance\ref@foot by -\cnt@footskip
212 }
213 \AtBeginDocument{\lay@getvalues}

```

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

```

\layout
\layout*
214 \newcommand\layout{%
215 \@ifstar{\lay@getvalues\lay@xlayout}{\lay@xlayout}}
216 \def\lay@xlayout{%
217 \lay@layout
218 \if@twoside
219 \lay@layout
220 \fi}

```

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

```

221 \newcommand\lay@layout{%
222 \thispagestyle{empty}

```

The actions of `\layout` depend on the pagestyle.

```

223 \if@twoside
224 \ifodd\count\z@

```

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

```

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

```

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

```

226 \ref@marginwidth=\cnt@oddsidemargin
227 \ref@marginpar=\oneinch
228 \advance\ref@marginpar by \ref@hoffset
229 \advance\ref@marginpar by \cnt@oddsidemargin
230 \ref@margin\ref@marginpar
231 \if@reversemargin

```



```

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

```

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

```

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

```

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

```

240 \ref@marginwidth=\cnt@evensidemargin
241 \ref@marginpar=\oneinch
242 \advance\ref@marginpar by \ref@hoffset
243 \advance\ref@marginpar by \cnt@evensidemargin
244 \ref@margin\ref@marginpar
245 \if@reversemargin
246 \advance\ref@marginpar by \cnt@textwidth
247 \advance\ref@marginpar by \cnt@marginparsep
248 \else
249 \advance\ref@marginpar by -\cnt@marginparsep
250 \advance\ref@marginpar by -\cnt@marginparwidth
251 \fi
252 \fi
253 \else

```

Finally we the case for single sided printing.

```

254 \typeout{One-sided document style.}
255 \ref@marginwidth=\cnt@oddsidemargin
256 \ref@marginpar=\oneinch
257 \advance\ref@marginpar by \ref@hoffset
258 \advance\ref@marginpar by \cnt@oddsidemargin
259 \ref@margin\ref@marginpar
260 \if@reversemargin
261 \advance\ref@marginpar by -\cnt@marginparsep
262 \advance\ref@marginpar by -\cnt@marginparwidth
263 \else
264 \advance\ref@marginpar by \cnt@textwidth
265 \advance\ref@marginpar by \cnt@marginparsep
266 \fi
267 \fi

```

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

```

268 \setlength{\unitlength}{.5pt}
269 \begin{picture}(\cnt@paperwidth,\cnt@paperheight)
270 \centering
271 \thicklines

```

First we have the pagebox and reference lines,

```

272 \put(0,0){\framebox(\cnt@paperwidth,\cnt@paperheight){\mbox{}}}
273 \put(0,\cnt@voffset){\dashbox{10}(\cnt@paperwidth,0){\mbox{}}}
274 \put(\cnt@hoffset,0){\dashbox{10}(0,\cnt@paperheight){\mbox{}}}

```

then the header,

```
275 \put(\ref@margin,\ref@head){%
276 \framebox(\cnt@textwidth,\cnt@headheight)%
277 {\footnotesize\Headertext}}
```

the body of the text area,

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

the footer

```
280 \put(\ref@margin,\ref@foot){%
281 \framebox(\cnt@textwidth,\fheight){\footnotesize\FooterText}}
```

and the space for marginal notes.

```
282 \put(\ref@marginpar,\ref@body){%
283 \framebox(\cnt@marginparwidth,\cnt@textheight)%
284 {\footnotesize\shortstack{\MarginNotestext}}}
```

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

```
285 \thinlines
```

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

```
286 \SetToHalf\PositionX\cnt@textwidth
287 \advance\PositionX by \ref@margin
```

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

```
288 \PositionY = \ref@body
289 \advance\PositionY by 50
```

An identifying number is put here, in a circle.

```
290 \Identify{8}
```

Then the arrow is drawn.

```
291 \InsideHArrow\cnt@textwidth
```

Now the `\textheight`

```
292 \SetToHalf\PositionY\cnt@textheight
293 \advance\PositionY by \ref@body
```

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

```
294 \PositionX = \cnt@textwidth
295 \divide\PositionX by 5
296 \multiply\PositionX by 4
297 \advance\PositionX by \ref@margin
```

An identifying number is put here, in a circle.

```
298 \Identify{7}
299 \InsideVArrow\cnt@textheight
```

The `\hoffset`,

```
300 \PositionY = 50
301 \SetToHalf\PositionX\cnt@hoffset
302 \Identify{1}
303 \InsideHArrow\cnt@hoffset
```

The width of the margin.

```

304 \SetToQuart\PositionY\cnt@textheight
305 \advance\PositionY by \ref@body
306 \ifnum\ref@marginwidth > 0
307 \OutsideHArrow\ref@margin\ref@marginwidth{20}
308 \PositionX = \cnt@hoffset
309 \else
310 \OutsideHArrow\cnt@hoffset{-\ref@marginwidth}{20}
311 \PositionX = \ref@margin
312 \fi
313 \advance\PositionX by -30
314 \Identify{3}

```

the \marginparwidth,

```

315 \SetToQuart\PositionY\cnt@textheight
316 \advance\PositionY by \ref@body

```

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

```

317 \advance\PositionY by 30
318 \SetToHalf\PositionX\cnt@marginparwidth
319 \advance\PositionX by \ref@marginpar
320 \Identify{10}
321 \InsideHArrow\cnt@marginparwidth

```

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

```

322 \advance\PositionY by 30
323 \if@twoside

```

Twosided mode, reversemargin;

```

324 \if@reversemargin
325 \ifodd\count\z@
326 \OutsideHArrow\ref@margin\cnt@marginparsep{20}
327 \PositionX = \ref@margin
328 \else
329 \OutsideHArrow\ref@marginpar\cnt@marginparsep{20}
330 \PositionX = \ref@marginpar
331 \fi
332 \else

```

Not reversemargin;

```

333 \ifodd\count\z@
334 \OutsideHArrow\ref@marginpar\cnt@marginparsep{20}
335 \PositionX = \ref@marginpar
336 \else
337 \OutsideHArrow\ref@margin\cnt@marginparsep{20}
338 \PositionX = \ref@margin
339 \fi
340 \fi
341 \else

```

Single sided mode.

```

342 \if@reversemargin
343 \OutsideHArrow\ref@margin\cnt@marginparsep{20}
344 \PositionX = \ref@margin
345 \else

```

```

346 \OutsideHArrow\ref@marginpar\cnt@marginparsep{20}
347 \PositionX = \ref@marginpar
348 \fi
349 \fi
350 \advance\PositionX by -\cnt@marginparsep
351 \advance\PositionX by -30
352 \Identify{9}

```

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

```

353 \PositionX = \cnt@textwidth
354 \divide\PositionX by 8
355 \advance\PositionX by \ref@margin
356 \OutsideVArrow\ref@foot\cnt@footskip{20}{20}
357 \PositionY = \ref@foot
358 \advance\PositionY by \cnt@footskip
359 \advance\PositionY by 30
360 \Identify{11}

```

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

```

361 \PositionX = \cnt@paperwidth
362 \advance\PositionX by -50
363 \PositionY = \cnt@paperheight
364 \ExtraYPos = \PositionY
365 \advance\ExtraYPos by -\cnt@voffset
366 \advance\PositionY by \cnt@voffset
367 \divide\PositionY by \tw@
368 \Identify{2}
369 \InsideVArrow\ExtraYPos

```

Identify `\topmargin`, `\headheight` and `\headsep`.

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

```

370 \Interval = \cnt@textwidth
371 \divide\Interval by 8
372 \PositionX = \ref@margin
373 \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.

```

374 \ifnum\cnt@topmargin > \z@
375 \ExtraYPos = \ref@head
376 \advance\ExtraYPos\cnt@headheight
377 \OutsideVArrow\ExtraYPos\cnt@topmargin{20}{20}
378 \PositionY = \ExtraYPos
379 \advance\PositionY by \cnt@topmargin
380 \else
381 \ExtraYPos = \cnt@voffset
382 \OutsideVArrow\ExtraYPos{-\cnt@topmargin}{20}{20}
383 \PositionY = \ExtraYPos
384 \advance\PositionY by -\cnt@topmargin
385 \fi
386 \advance\PositionY by 30
387 \Identify{4}

```

```

388 \advance\PositionX by \Interval
Then the \headheight
389 \OutsideVArrow\ref@head\cnt@headheight{20}{20}
390 \PositionY = \ref@head
391 \advance\PositionY by \cnt@headheight
392 \advance\PositionY by 30
393 \Identify{5}
394 \advance\PositionX by \Interval
and finally the \headsep
395 \ExtraYPos=\ref@body
396 \advance\ExtraYPos\cnt@textheight
397 \OutsideVArrow\ExtraYPos\cnt@headsep{20}{20}
398 \PositionY = \ref@body
399 \advance\PositionY by \cnt@textheight
400 \advance\PositionY by -30
401 \Identify{6}

```

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

```

402 \end{picture}
403
404 \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.

```

405 \vtop to 0pt{%
406 \@minipagerestore\footnotesize\ttfamily
407 \begin{tabular}{@{}rl@{\hspace{20pt}}rl}
408 1 & \oneinchtext\ + \LayOutbs\texttt{hoffset}
409 & 2 & \oneinchtext\ + \LayOutbs\texttt{voffset} \\
410 3 & \if@twoside
411 \ifodd\count\z@ \Show{cnt}{oddsidemargin}
412 \else \Show{cnt}{evensidemargin}
413 \fi
414 \else
415 \Show{cnt}{oddsidemargin}
416 \fi
417 & 4 & \Show{cnt}{topmargin} \\
418 5 & \Show{cnt}{headheight} & 6 & \Show{cnt}{headsep} \\
419 7 & \Show{cnt}{textheight} & 8 & \Show{cnt}{textwidth} \\
420 9 & \Show{cnt}{marginparsep}&10& \Show{cnt}{marginparwidth} \\
421 11& \Show{cnt}{footskip} & & \Show{cnt}{marginparpush}
422 \rlap{(\notshown)}\
423 & \Show{ref}{hoffset} & & \Show{ref}{voffset} \\
424 & \Show{cnt}{paperwidth} & & \Show{cnt}{paperheight} \\
425 \end{tabular}\vss}

```

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

```

426 \Type{ref}{hoffset}
427 \Type{ref}{voffset}
428 \Type{cnt}{textheight}
429 \Type{cnt}{textwidth}

```

Finally we start a new page.

```
430 \newpage
```

```
431 }
```

```
432 \end{package}
```