The **geometry** package

Hideo Umeki

latexgeometry@gmail.com

2010/03/13 v5.3

Abstract

This package provides a flexible and easy interface to page dimensions. You can change the page layout with intuitive parameters. For instance, if you want to set a margin to 2cm from each edge of the paper, you can go just `\usepackage[margin=2cm]{geometry}`. The page layout can be changed in the middle of the document with `\newgeometry` command.

1 Preface to version 5

- **Changing page layout mid-document.**

  The new commands `\newgeometry{···}` and `\restoregeometry` allow users to change page dimensions in the middle of the document. `\newgeometry` is almost similar to `\geometry` except that `\newgeometry` disables all the options specified in the preamble and skips the papersize-related options: `landscape,portrait` and paper size options (such as `papersize`, `paper=a4paper` and so forth).

- **A new set of options to specify the layout area.**

  The options specified for the area, in which the page dimensions are calculated, are added: `layout`, `layoutsize`, `layoutwidth`, `layoutheight` and so forth. These options would help to print the specified layout to a different sized paper. For example, with `a4paper` and `layout=a5paper`, the geometry package uses 'A5' layout to calculate margins with the paper size still 'A4'.

- **A new driver option** `xetex`**.**

  The new driver option `xetex` is added. The driver auto-detection routine has been revised so as to avoid an error with undefined control sequences. Note that 'geometry.cfg' in TeXLive, which disables the auto-detection routine and sets `pdftex`, is no longer necessary and has no problem even though it still exists. To set `xetex` is strongly recommended with XeLaTeX.

- **New paper size presets for JIS B-series.**

  The papersize presets `b0j` to `b6j` for JIS (Japanese Industrial Standards) B-series are added.

- **Changing default for underspecified margin.**

  In the previous version, if only one margin was specified, `bottom=1cm` for example, then geometry set the other margin with the margin ratio (1:1 by default for the vertical dimensions) and got `top=1cm` in this case. The version 5 sets the text-body size with the default `scale` (=0.7) and determine the unspecified margin. (See Section 6.5)

- **The option** `showframe` **works on every page.**

  With `showframe` option, the page frames are shown on every page. In addition, the new option `showcrop` would print cropmarks showing the corners of the layout (the same as paper by default) on every page.

- **Loading geometry.cfg precedes processing class options.**

  The previous version loaded geometry.cfg after processing the document class options. Now that the config file is loaded before processing the class options, you can change the behavior specified in geometry.cfg by adding options into `\documentclass` as well as `\usepackage` and `\geometry`.

- **Deleted options:** `compat2` **and** `twosideshift`**.** The version 5 has no longer compatibility with the previous ones. `compat2` and `twosideshift` are gone for simplicity.

2 Introduction

To set dimensions for page layout in LaTeX is not straightforward. You need to adjust several LaTeX native dimensions to place a text area where you want If you want to center the text area in the paper you use, for example, you have to specify native dimensions as follows:

```
\usepackage{calc}
\setlength\textwidth{7in}
\setlength\textheight{10in}
\setlength\oddsidemargin{(\paperwidth-\textwidth)/2 - 1in}
\setlength\topmargin{(\paperheight-\textheight
                     -\headheight-\headsep-\footskip)/2 - 1in}.
```

Without package *calc*, the above example would need more tedious settings. Package geometry provides an easy way to set page layout parameters. In this case, what you have to do is just

```
\usepackage[text={7in,10in},centering]{geometry}.
```

Besides centering problem, setting margins from each edge of the paper is also troublesome. But geometry also make it easy. If you want to set each margin 1.5in, you can go

```
\usepackage[margin=1.5in]{geometry}
```

Thus, the geometry package has an auto-completion mechanisim, in which unspecified dimensions are automatically determined. The geometry package will be also useful when you have to set page layout obeying the following strict instructions: for example,

> *The total allowable width of the text area is 6.5 inches wide by 8.75 inches high. The top margin on each page should be 1.2 inches from the top edge of the page. The left margin should be 0.9 inch from the left edge. The footer with page number should be at the bottom of the text area.*

In this case, using geometry you can go

```
\usepackage[total={6.5in,8.75in},
            top=1.2in, left=0.9in, includefoot]{geometry}.
```

Setting a text area on the paper in document preparation system has some analogy to placing a window on the background in the window system. The name 'geometry' comes from the -geometry option used for specifying a size and location of a window in X Window System.

3 Page geometry

Figure 1 shows the page layout dimensions defined in the geometry package. The page layout contains a *total body* (printable area) and *margins*. The *total body* consists of a *body* (text area) with an optional *header*, *footer* and marginal notes (marginpar). There are four margins: *left*, *right*, *top* and *bottom*. For twosided documents, horizontal margins should be called *inner* and *outer*.

$$
\begin{array}{rcl}
paper & : & total\ body\ \text{and}\ margins \\
total\ body & : & body\ \text{(text area)}\quad \text{(optional}\ head,\ foot\ \text{and}\ marginpar) \\
margins & : & left(inner),\ right(outer),\ top\ \text{and}\ bottom
\end{array}
$$

Each margin is measured from the corresponding edge of a paper. For example, left margin (inner margin) means a horizontal distance between the left (inner) edge of the paper and that of the total body. Therefore the left and top margins defined in geometry are different from the native dimensions \leftmargin and \topmargin. The size of a body (text area) can be modified by \textwidth and \textheight. The dimensions for paper, total body and margins have the following relations.

$$\text{paperwidth} \;=\; \text{left} + \text{width} + \text{right} \tag{1}$$

$$\text{paperheight} \;=\; \text{top} + \text{height} + \text{bottom} \tag{2}$$

The total body `width` and `height` would be defined:

$$\text{width} \;:=\; \text{textwidth} \;\;(+\text{marginparsep} + \text{marginparwidth}) \tag{3}$$

$$\text{height} \;:=\; \text{textheight} \;\;(+\text{headheight} + \text{headsep} + \text{footskip}) \tag{4}$$
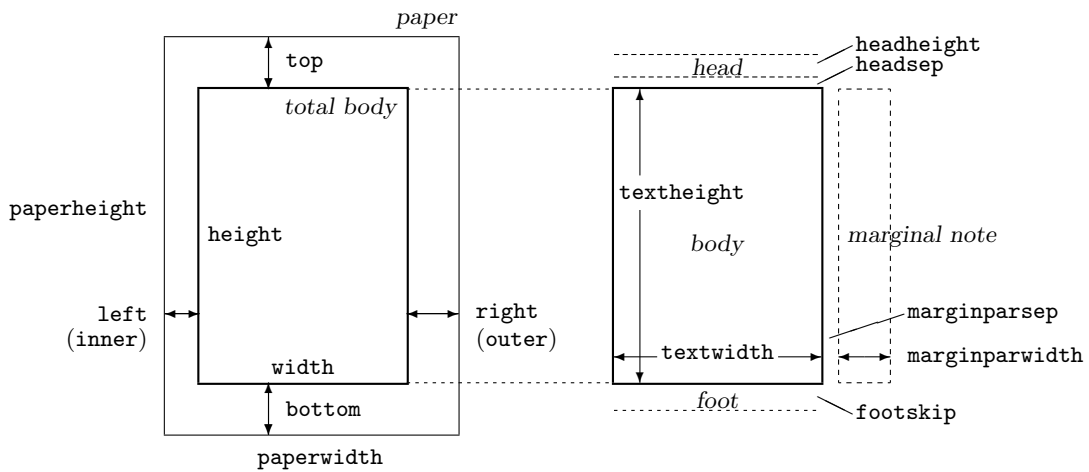
Figure 1: Dimension names used in the `geometry` package. `width=textwidth` and `height=textheight` by default. `left`, `right`, `top` and `bottom` are margins. If margins on verso pages are swapped by `twoside` option, margins specified by `left` and `right` options are used for the inside and outside margins respectively. `inner` and `outer` are aliases of `left` and `right` respectively.
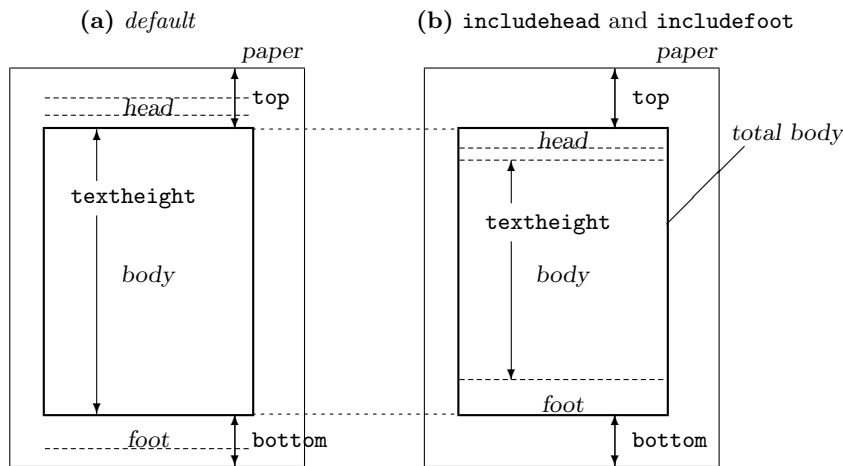

Figure 2: `includehead` and `includefoot` include the head and foot respectively into *total body*. **(a)** `height = textheight` (default). **(b)** `height = textheight + headheight + headsep + footskip` if `includehead` and `includefoot`. If the top and bottom margins are specified, `includehead` and `includefoot` result in shorter `textheight`.

In Equation (3) `width:=textwidth` by default, while `marginparsep` and `marginparwidth` are included in `width` if `includemp` option is set `true`. In Equation (4), `height:=textheight` by default. If `includehead` is set to `true`, `headheight` and `headsep` are considered as a part of `height`. In the same way, `includefoot` takes `footskip` into `height`. Figure 2 shows how these options work in the vertical direction.

Thus, the page layout consists of three parts (lengths) in each direction: one body and two margins. If the two of them are explicitly specified, the other length is obvious and no need to be specified. Figure 3 shows a simple model of page dimensions. When a length L is given and is partitioned into the body `b`, the margins `a` and `c`, it's obvious that

$$L = a + b + c \tag{5}$$

The specification with two of the three (`a`,`b` and `c`) fixed explicitly is solvable. If two or more are left unspecified or 'underspecified', Equation (5) cannot be solved without any other relation between them. If all of them are specified, then it needs to check whether or not they satisfy Equation (5), that is too much specification or 'overspecified'.

The `geometry` package has auto-completion mechanism that saves the trouble of specifying the page layout dimensions. For example, you can set

```
\usepackage[width=14cm, left=3cm]{geometry}
```
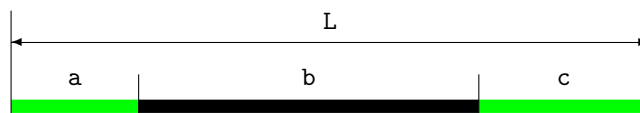
Figure 3: A simple model of page dimensions.

on A4 paper. In this case you don't have to set the right margin The details of auto-completion will be described in Section 6.5.

4 User interface

4.1 Commands

The geometry package provides the following commands:

- \geometry{⟨options⟩}
- \newgeometry{⟨options⟩} and \restoregeometry
- \savegeometry{⟨name⟩} and \loadgeometry{⟨name⟩}

\geometry{⟨options⟩} changes the page layout according to the options specified in the argument. This command, if any, should be placed only in the preamble (before \begin{document}).

The geometry package may be used as part of a class or another package you use in your document. The command \geometry can overwrite some of the settings in the preamble. Multiple use of \geometry is allowed and then processed with the options concatenated. If geometry is not yet loaded, you can use only \usepackage[⟨options⟩]{geometry} instead of \geometry.

\newgeometry{⟨options⟩} changes the page layout mid-document. \newgeometry is almost similar to \geometry except that \newgeometry disables all the options specified by \usepackage and \geometry in the preamble and skips papersize-related options. \restoregeometry restores the page layout specified in the preamble. This command has no arguments. See Section 7 for details.

\savegeometry{⟨name⟩} saves the page dimensions as ⟨name⟩ where you put this command. \loadgeometry{⟨name⟩} loads the page dimensions saved as ⟨name⟩. See Section 7 for details.

4.2 Optional argument

The geometry package adopts keyval interface '⟨key⟩=⟨value⟩' for the optional argument to \usepackage, \geometry and \newgeometry.

The argument includes a list of comma-separated keyval options and has basic rules as follows:

- Multiple lines are allowed, while blank lines are not.
- Any spaces between words are ignored.
- Options are basically order-independent. (There are some exceptions. See Section 6.2 for details.)

For example,

```
\usepackage[ a5paper ,  hmargin = { 3cm,
             .8in } , height
      =  10in ]{geometry}
```

is equivalent to

```
\usepackage[height=10in,a5paper,hmargin={3cm,0.8in}]{geometry}
```

Some options are allowed to have sub-list, e.g. {3cm,0.8in}. Note that the order of values in the sub-list is significant. The above setting is also equivalent to the followings:

```
\usepackage{geometry}
\geometry{height=10in,a5paper,hmargin={3cm,0.8in}}
```

or

```
\usepackage[a5paper]{geometry}
\geometry{hmargin={3cm,0.8in},height=8in}
\geometry{height=10in}.
```

Thus, multiple use of \geometry just appends options.

geometry supports package *calc*[1]. For example,

```
\usepackage{calc}
\usepackage[textheight=20\baselineskip+10pt]{geometry}
```

4.3 Option types

geometry options are categorized into four types:

1. **Boolean type**

   takes a boolean value (`true` or `false`). If no value, `true` is set by default.

   $\langle key \rangle$=`true` | `false`.
   $\langle key \rangle$ with no value is equivalent to $\langle key \rangle$=`true`.

   Examples: `verbose=true`, `includehead`, `twoside=false`.
   Paper name is the exception. The preferred paper name should be set with no values. Whatever value is given, it is ignored. For instance, `a4paper=XXX` is equivalent to `a4paper`.

2. **Single-valued type**

   takes a mandatory value.

   $\langle key \rangle$=$\langle value \rangle$.

   Examples: `width=7in`, `left=1.25in`, `footskip=1cm`, `height=.86\paperheight`.

3. **Double-valued type**

   takes a pair of comma-separated values in braces. The two values can be shortened to one value if they are identical.

   $\langle key \rangle$={$\langle value1 \rangle$,$\langle value2 \rangle$}.
   $\langle key \rangle$=$\langle value \rangle$ is equivalent to $\langle key \rangle$={$\langle value \rangle$,$\langle value \rangle$}.

   Examples: `hmargin={1.5in,1in}`, `scale=0.8`, `body={7in,10in}`.

4. **Triple-valued type**

   takes three mandatory, comma-separated values in braces.

   $\langle key \rangle$={$\langle value1 \rangle$,$\langle value2 \rangle$,$\langle value3 \rangle$}

   Each value must be a dimension or null. When you give an empty value or '`*`', it means null and leaves the appropriate value to the auto-completion mechanism. You need to specify at least one dimension, typically two dimensions. You can set nulls for all the values, but it makes no sense.
   Examples:
   `hdivide={2cm,*,1cm}`, `vdivide={3cm,19cm, }`, `divide={1in,*,1in}`.

5 Option details

This section describes all the options available in geometry. Options with a dagger † are not available as a argument of \newgeometry (See Section 7).

---

[1]CTAN: `macros/latex/required/tools`

5.1 Paper size

The options below set paper/media size and orientation.

† **paper | papername**

specifies the paper size by name. `paper=⟨paper-name⟩`. For convenience, you can specify the paper name without `paper=`. For example, `a4paper` is equivalent to `paper=a4paper`.

† **a0paper, a1paper, a2paper, a3paper, a4paper, a5paper, a6paper,**
b0paper, b1paper, b2paper, b3paper, b4paper, b5paper, b6paper,
b0j, b1j, b2j, b3j, b4j, b5j, b6j,
ansiapaper, ansibpaper, ansicpaper, ansidpaper, ansiepaper,
letterpaper, executivepaper, legalpaper

specifies paper name. The value part is ignored even if any. For example, the followings have the same effect: `a5paper`, `a5paper=true`, `a5paper=false` and so forth.
`a[0-6]paper` and `b[0-6]paper` are ISO A and B series of paper sizes. The JIS (Japanese Industrial Standards) A-series is identical to the ISO A-series, but the JIS B-series is different from the ISO B-series. `b[0-6]j` should be used for the JIS B-series.

† **screen**   a special paper size with (W,H) = (225mm,180mm). For presentation with PC and video projector, "`screen,centering`" with 'slide' documentclass would be useful.

† **paperwidth**   width of the paper. `paperwidth=⟨length⟩`.

† **paperheight**   height of the paper. `paperheight=⟨length⟩`.

† **papersize**   width and height of the paper. `papersize={⟨width⟩,⟨height⟩}` or `papersize=⟨length⟩`.

† **landscape**   switches the paper orientation to landscape mode.

† **portrait**   switches the paper orientation to portrait mode. This is equivalent to `landscape=false`.

The options for paper names (e.g., `a4paper`) and orientation (`portrait` and `landscape`) can be set as document class options. For example, you can set `\documentclass[a4paper,landscape]{article}`, then `a4paper` and `landscape` are processed in `geometry` as well. This is also the case for `twoside` and `twocolumn` (see also Section 5.5).

5.2 Layout size

You can specify the layout area with options described in this section regardless of the paper size. The options would help to print the specified layout to a different sized paper. For example, with `a4paper` and `layout=a5paper`, the package uses 'A5' layout to calculate margins on 'A4' paper. The layout size defaults to the same as the paper. The options for the layout size are available in `\newgeometry`, so that you can change the layout size in the middle of the document. The paper size itself can't be changed though. Figure 4 shows what the difference between `layout` and `paper` is.

layout   specifies the layout size by paper name. `layout=⟨paper-name⟩`. All th paper names defined in `geometry` are available. See Section 5.1 for details.

layoutwidth   width of the layout. `layoutwidth=⟨length⟩`.

layoutheight   height of the layout. `layoutheight=⟨length⟩`.

layoutsize   width and height of the layout. `layoutsize={⟨width⟩,⟨height⟩}` or `layoutsize=⟨length⟩`.

layouthoffset   specifies the horizontal offset from the left edge of the paper. `layouthoffset=⟨length⟩`.

layoutvoffset   specifies the vertical offset from the top edge of the paper. `layoutvoffset=⟨length⟩`.

layoutoffset   specifies both horizontal and vertical offsets. `layoutoffset={⟨hoffset⟩,⟨voffset⟩}` or `layoutsize=⟨length⟩`.

5.3 Body size

The options specifying the size of *total body* are described in this section.

hscale   ratio of width of *total body* to `\paperwidth`. `hscale=⟨h-scale⟩`, e.g., `hscale=0.8` is equivalent to `width=0.8\paperwidth`. (0.7 by default)

vscale   ratio of height of *total body* to `\paperheight`, e.g., `vscale=⟨v-scale⟩`. (0.7 by default) `vscale=0.9` is equivalent to `height=0.9\paperheight`.

layoutwidth layouthoffset layoutvoffset

head

body

foot

layout

paper

head
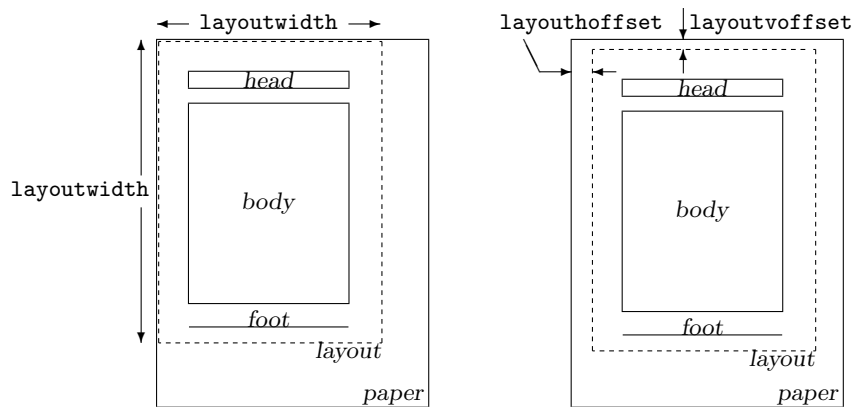
body

foot

layout

paper

layoutwidth

Figure 4: The dimensions related to the layout size. Note that the layout size defaults to the same size as the paper, so you don't have to specify layout-related options explicitly in most cases.

scale
: ratio of *total body* to the paper. `scale={⟨h-scale⟩,⟨v-scale⟩}` or `scale=⟨scale⟩`. (0.7 by default)

width │ totalwidth
: width of *total body*. `width=⟨length⟩` or `totalwidth=⟨length⟩`. This dimension defaults to textwidth, but if `includemp` is set to `true`, `width ≥ textwidth` because `width` includes the width of the marginal notes. If `textwidth` and `width` are specified at the same time, `textwidth` takes priority over `width`.

height │ totalheight
: height of *total body*, excluding header and footer by default. If `includehead` or `includefoot` is set, `height` includes the head or foot of the page as well as `textheight`. `height=⟨length⟩` or `totalheight=⟨length⟩`. If both `textheight` and `height` are specified, `height` will be ignored.

total
: width and height of *total body*.
`total={⟨width⟩,⟨height⟩}` or `total=⟨length⟩`.

textwidth
: specifies `\textwidth`, the width of *body* (the text area). `textwidth=⟨length⟩`.

textheight
: specifies `\textheight`, the height of *body* (the text area). `textheight=⟨length⟩`.

text │ body
: specifies both `\textwidth` and `\textheight` of the body of page.
`body={⟨width⟩,⟨height⟩}` or `text=⟨length⟩`.

lines
: enables users to specify `\textheight` by the number of lines. `lines=⟨integer⟩`.

includehead
: includes the head of the page, `\headheight` and `\headsep`, into *total body*. It is set to `false` by default. It is opposite to `ignorehead`. See Figure 2 and Figure 5.

includefoot
: includes the foot of the page, `\footskip`, into *total body*. It is opposite to `ignorefoot`. It is `false` by default. See Figure 2 and Figure 5.

includeheadfoot
: sets both `includehead` and `includefoot` to `true`, which is opposite to `ignoreheadfoot`. See Figure 2 and Figure 5.

includemp
: includes the margin notes, `\marginparwidth` and `\marginparsep`, into *body* when calculating horizontal calculation.

includeall
: sets both `includeheadfoot` and `includemp` to `true`. See Figure 5.

ignorehead
: disregards the head of the page, `headheight` and `headsep`, in determining vertical layout, but does not change those lengths. It is equivalent to `includehead=false`. It is set to `true` by default. See also `includehead`.

ignorefoot
: disregards the foot of page, `footskip`, in determining vertical layout, but does not change that length. This option defaults to `true`. See also `includefoot`.

ignoreheadfoot
: sets both `ignorehead` and `ignorefoot` to `true`. See also `includeheadfoot`.

ignoremp
: disregards the marginal notes in determining the horizontal margins (defaults to `true`). If marginal notes overrun the page, the warning message will be displayed when `verbose=true`. See also `includemp` and Figure 5.
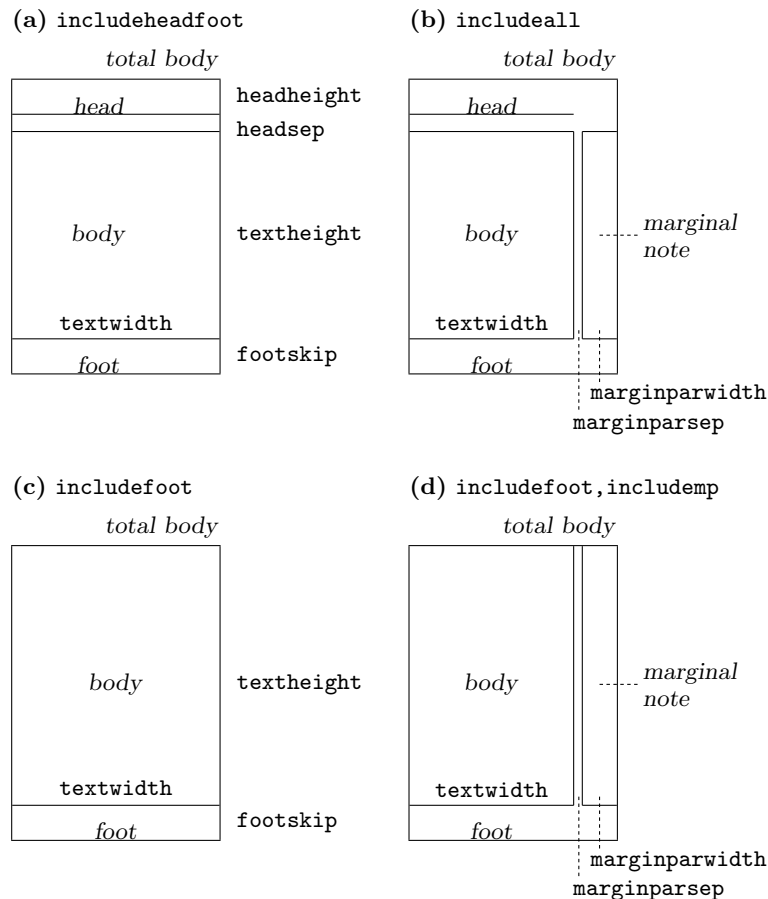
7

Figure 5: Sample layouts for *total body* with different switches. (a) `includeheadfoot`, (b) `includeall`, (c) `includefoot` and (d) `includefoot,includemp`. If `reversemp` is set to `true`, the location of the marginal notes are swapped on every page. Option `twoside` swaps both margins and marginal notes on verso pages. Note that the marginal note, if any, is printed despite `ignoremp` or `includemp=false` and overrun the page in some cases.

`ignoreall`	sets both `ignoreheadfoot` and `ignoremp` to `true`. See also `includeall`.
`heightrounded`	

This option rounds `\textheight` to n-times (n: an integer) of `\baselineskip` plus `\topskip` to avoid "underfull vbox" in some cases. For example, if `\textheight` is 486pt with `\baselineskip` 12pt and `\topskip` 10pt, then

$$(39 \times 12\text{pt} + 10\text{pt} =) 478\text{pt} < 486\text{pt} < 490\text{pt} (= 40 \times 12\text{pt} + 10\text{pt}),$$

as a result `\textheight` is rounded to 490pt. `heightrounded=false` by default.

Figure 5 illustrates various layouts with different layout modes. The dimensions for a header and a footer can be controlled by `nohead` or `nofoot` mode, which sets each length to 0pt directly. On the other hand, options with the prefix `ignore` do *not* change the corresponding native dimensions.

The following options can specify body and margins simultaneously with three comma-separated values in braces.

`hdivide`	horizontal partitions (left,width,right). `hdivide={⟨left margin⟩,⟨width⟩,⟨right margin⟩}`. Note that you should not specify all of the three parameters. The best way of using this option is to specify two of three and leave the rest with null(nothing) or '`*`'. For example, when you set `hdivide={2cm,15cm, }`, the margin from the right-side edge of page will be determined calculating `paperwidth-2cm-15cm`.
`vdivide`	vertical partitions (top,height,bottom). `vdivide={⟨top margin⟩,⟨height⟩,⟨bottom margin⟩}`.
`divide`	`divide={A,B,C}` is interpreted as `hdivide={A,B,C}` and `vdivide={A,B,C}`.

5.4 Margin size

The options specifying the size of the margins are listed below.

left | lmargin | inner
    left margin (for oneside) or inner margin (for twoside) of *total body*. In other words, the distance between the left (inner) edge of the paper and that of *total body*. `left=⟨length⟩`. `inner` has no special meaning, just an alias of `left` and `lmargin`.

right | rmargin | outer
    right or outer margin of *total body*. `right=⟨length⟩`.

top | tmargin top margin of the page. `top=⟨length⟩`. Note this option has nothing to do with the native dimension `\topmargin`.

bottom | bmargin
    bottom margin of the page. `bottom=⟨length⟩`.

hmargin   left and right margin. `hmargin={⟨left margin⟩,⟨right margin⟩}` or `hmargin=⟨length⟩`.

vmargin   top and bottom margin. `vmargin={⟨top margin⟩,⟨bottom margin⟩}` or `vmargin=⟨length⟩`.

margin   `margin={A,B}` is equivalent to `hmargin={A,B}` and `vmargin={A,B}`. `margin=A` is automatically expanded to `hmargin=A` and `vmargin=A`.

hmarginratio horizontal margin ratio of `left` (inner) to `right` (outer). The value of ⟨*ratio*⟩ should be specified with colon-separated two values. Each value should be a positive integer less than 100 to prevent arithmetic overflow, e.g., `2:3` instead of `1:1.5`. The default ratio is `1:1` for oneside, `2:3` for twoside.

vmarginratio vertical margin ratio of `top` to `bottom`. The default ratio is `2:3`.

marginratio | ratio
    horizontal and vertical margin ratios. `marginratio={⟨horizontal ratio⟩,⟨vertical ratio⟩}` or `marginratio=⟨ratio⟩`.

hcentering sets auto-centering horizontally and is equivalent to `hmarginratio=1:1`. It is set to `true` by default for oneside. See also `hmarginratio`.

vcentering sets auto-centering vertically and is equivalent to `vmarginratio=1:1`. The default is `false`. See also `vmarginratio`.

centering sets auto-centering and is equivalent to `marginratio=1:1`. See also `marginratio`. The default is `false`. See also `marginratio`.

twoside   switches on twoside mode with left and right margins swapped on verso pages. The option sets `\@twoside` and `\@mparswitch` switches. See also `asymmetric`.

asymmetric implements a twosided layout in which margins are not swapped on alternate pages (by setting `\oddsidemargin` to `\evensidemargin + bindingoffset`) and in which the marginal notes stay always on the same side. This option can be used as an alternative to the twoside option. See also `twoside`.

bindingoffset
    removes a specified space from the lefthand-side of the page for oneside or the inner-side for twoside. `bindingoffset=⟨length⟩`. This is useful if pages are bound by a press binding (glued, stitched, stapled . . .). See Figure 6.

hdivide   See description in Section 5.3.

vdivide   See description in Section 5.3.

divide   See description in Section 5.3.

5.5 Native dimensions

The options below overwrite LaTeX native dimensions and switches for page layout (See the right-hand side in Figure 1).

headheight | head
    modifies `\headheight`, height of header. `headheight=⟨length⟩` or `head=⟨length⟩`.

headsep   modifies `\headsep`, separation between header and text (body). `headsep=⟨length⟩`.

footskip | foot
    modifies `\footskip`, distance separation between baseline of last line of text and baseline of footer. `footskip=⟨length⟩` or `foot=⟨length⟩`.

a) every page for oneside or odd pages for twoside
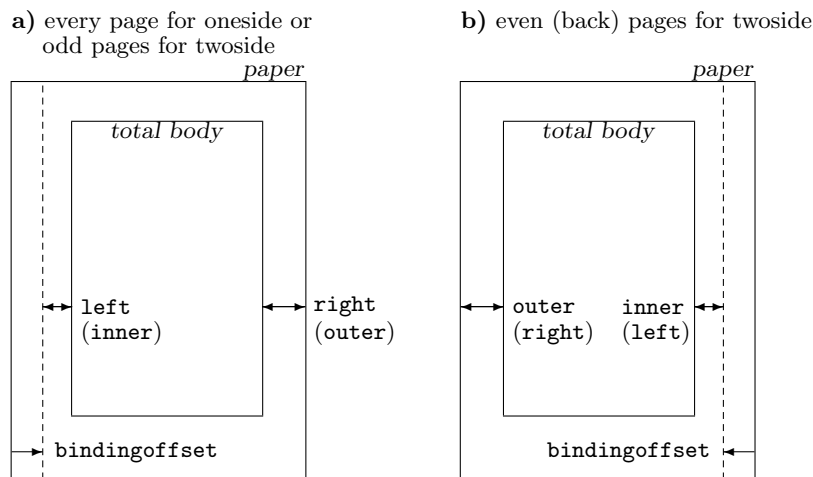
b) even (back) pages for twoside

Figure 6: The option `bindingoffset` adds the specified length to the inner margin. Note that `twoside` option swaps the horizontal margins and the marginal notes together with `bindingoffset` on even pages (see **b)**), but `asymmetric` option suppresses the swap of the margins and marginal notes (but `bindingoffset` is still swapped).

nohead	eliminates spaces for the head of the page, which is equivalent to both `\headheight=0pt` and `\headsep=0pt`.
nofoot	eliminates spaces for the foot of the page, which is equivalent to `\footskip=0pt`.
noheadfoot	equivalent to `nohead` and `nofoot`, which means that `\headheight`, `\headsep` and `\footskip` are all set to `0pt`.
footnotesep	changes the dimension `\skip\footins`, separation between the bottom of text body and the top of footnote text.
marginparwidth │ marginpar	modifies `\marginparwidth`, width of the marginal notes. `marginparwidth=⟨length⟩`.
marginparsep	modifies `\marginparsep`, separation between body and marginal notes. `marginparsep=⟨length⟩`.
nomarginpar	shrinks spaces for marginal notes to 0pt, which is equivalent to `\marginparwidth=0pt` and `\marginparsep=0pt`.
columnsep	modifies `\columnsep`, the separation between two columns in `twocolumn` mode.
hoffset	modifies `\hoffset`. `hoffset=⟨length⟩`.
voffset	modifies `\voffset`. `voffset=⟨length⟩`.
offset	horizontal and vertical offset. `offset={⟨hoffset⟩,⟨voffset⟩}` or `offset=⟨length⟩`.
twocolumn	sets `twocolumn` mode with `\@twocolumntrue`. `twocolumn=false` denotes onecolumn mode with `\@twocolumnfalse`. Instead of `twocolumn=false`, you can specify `onecolumn` (which defaults to `true`)
onecolumn	works as `twocolumn=false`. On the other hand, `onecolumn=false` is equivalent to `twocolumn`.
twoside	sets both `\@twosidetrue` and `\@mparswitchtrue`. See Section 5.4.
textwidth	sets `\textwidth` directly. See Section 5.3.
textheight	sets `\textheight` directly. See Section 5.3.
reversemp │ reversemarginpar	makes the marginal notes appear in the left (inner) margin with `\@reversemargintrue`. The option doesn't change `includemp` mode. It's set `false` by default.

5.6 Drivers

The package supports drivers `dvips`, `dvipdfm`, `pdftex`, `xetex` and `vtex`. You can also set `dvipdfm` for `dvipdfmx` and `xdvipdfmx`. `pdftex` for `pdflatex`, and `vtex` for VTEX environment. The driver options are exclusive. The driver can be set by either `driver=⟨driver name⟩` or any of the drivers directly like `pdftex`. By default, `geometry` guesses the driver appropriate to the system in use. Therefore, you don't have to set a driver in most cases. However, if you want to use `dvipdfm`, you should specify it explicitly.

† **driver** specifies the driver with `driver=⟨driver name⟩`. `dvips`, `dvipdfm`, `pdftex`, `vtex`, `xetex`, `auto` and `none` are available as a driver name. The names except for `auto` and `none` can be specified directly with the name without `driver=`. `driver=auto` makes the auto-detection work whatever the previous setting is. `driver=none` disables the auto-detection and sets no driver, which may be useful when you want to let other package work out the driver setting. For example, if you want to use `crop` package with `geometry`, you should call `\usepackage[driver=none]{geometry}` before the `crop` package.

† **dvips** writes the paper size in dvi output with the `\special` macro. If you use *dvips* as a DVI-to-PS driver, for example, to print a document with `\geometry{a3paper,landscape}` on A3 paper in landscape orientation, you don't need options "`-t a3 -t landscape`" to *dvips*.

† **dvipdfm** works like `dvips` except for landscape correction. You can set this option when using dvipdfmx and xdvipdfmx to process the dvi output.

† **pdftex** sets `\pdfpagewidth` and `\pdfpageheight` internally.

† **xetex** is the same as `pdftex` except for ignoring `\pdf{h,v}origin` undefined in XeLaTeX. This option is introduced in the version 5. Note that 'geometry.cfg' in TeXLive, which disables the auto-detection routine and sets `pdftex`, is no longer necessary, but has no problem even though it's left undeleted. Instead of `xetex`, you can specify `dvipdfm` with XeLaTeX if you want to use specials of dvipdfm XeTeX supports.

† **vtex** sets dimensions `\mediawidth` and `\mediaheight` for VTeX. When this driver is selected (explicitly or automatically), `geometry` will auto-detect which output mode (DVI, PDF or PS) is selected in VTeX, and do proper settings for it.

If explicit driver setting is mismatched with the typesetting program in use, the default driver `dvips` would be selected.

5.7 Other options

The other useful options are described here.

† **verbose** displays the parameter results on the terminal. `verbose=false` (default) still puts them into the log file.

† **reset** sets back the layout dimensions and switches to the settings before `geometry` is loaded. Options given in `geometry.cfg` are also cleared. Note that this cannot reset `pass` and `mag` with `truedimen`. `reset=false` has no effect and cannot cancel the previous `reset(=true)` if any. For example, when you go

```
\documentclass[landscape]{article}
\usepackage[twoside,reset,left=2cm]{geometry}
```

with `\ExecuteOptions{scale=0.9}` in `geometry.cfg`, then as a result, `landscape` and `left=2cm` remain effective, and `scale=0.9` and `twoside` are ineffective.

† **mag** sets magnification value (`\mag`) and automatically modifies `\hoffset` and `\voffset` according to the magnification. `mag=⟨value⟩`. Note that ⟨value⟩ should be an integer value with 1000 as a normal size. For example, `mag=1414` with `a4paper` provides an enlarged print fitting in `a3paper`, which is $1.414 (=\sqrt{2})$ times larger than `a4paper`. Font enlargement needs extra disk space. **Note that setting `mag` should precede any other settings with 'true' dimensions, such as** `1.5truein`, `2truecm` **and so on.** See also `truedimen` option.

† **truedimen** changes all internal explicit dimension values into *true* dimensions, e.g., `1in` is changed to `1truein`. Typically this option will be used together with `mag` option. Note that this is ineffective against externally specified dimensions. For example, when you set "`mag=1440, margin=10pt, truedimen`", margins are not 'true' but magnified. If you want to set exact margins, you should set like "`mag=1440, margin=10truept, truedimen`" instead.

† **pass** disables all of the geometry options and calculations except `verbose` and `showframe`. It can be used for checking out the page layout of the documentclass, other packages and manual settings without `geometry`.

† **showframe** shows visible frames for the text area and page, and the lines for the head and foot on the first page.

6 Processing options

6.1 Order of loading

If there's `geometry.cfg` somewhere TeX can find it, `geometry` loads it first. For example, in `geometry.cfg` you may write `\ExecuteOptions{a4paper}`, which specifies A4 paper as the default paper. Basically you can use all the options defined in `geometry` with `\ExecuteOptions{}`.

The order of loading in the preamble of your document is as follows:

1. `geometry.cfg` if it exists.

2. Options specified with `\documentclass[`⟨*options*⟩`]{...}`.

3. Options specified with `\usepackage[`⟨*options*⟩`]{geometry}`

4. Options specified with `\geometry{`⟨*options*⟩`}`, which can be called multiple times. (`reset` option will cancel the specified options ever given in `\usepackage{geometry}` or `\geometry`.)

6.2 Order of options

The specification of `geometry` options is order-independent, and overwrites the previous one for the same setting. For example,

$$[\texttt{left=2cm, right=3cm}] \text{ is equivalent to } [\texttt{right=3cm, left=2cm}].$$

The options called multiple times overwrite the previous settings. For example,

$$[\texttt{verbose=true, verbose=false}] \text{ results in } \texttt{verbose=false}.$$

`[hmargin={3cm,2cm}, left=1cm]` is the same as `hmargin={1cm,2cm}`, where the left (or inner) margin is overwritten by `left=1cm`.

`reset` and `mag` are exceptions. The `reset` option removes all the geometry options (except `pass`) before it. If you set

```
\documentclass[landscape]{article}
\usepackage[margin=1cm,twoside]{geometry}
\geometry{a5paper, reset, left=2cm}
```

then `margin=1cm`, `twoside` and `a5paper` are removed, and is eventually equivalent to

```
\documentclass[landscape]{article}
\usepackage[left=2cm]{geometry}
```

The `mag` option should be set in advance of any other settings with 'true' length, such as `left=1.5truecm`, `width=5truein` and so on. The `\mag` primitive can be set before this package is called.

6.3 Priority

There are several ways to set dimensions of the *body*: `scale`, `total`, `text` and `lines`. The `geometry` package gives higer priority to the more concrete specificaion. Here is the priority rule for *body*.

$$\text{priority:} \quad \text{low} \quad \longrightarrow \quad \text{high}$$

$$\left\{ \begin{array}{l} \texttt{hscale} \\ \texttt{vscale} \\ \texttt{scale} \end{array} \right\} < \left\{ \begin{array}{l} \texttt{width} \\ \texttt{height} \\ \texttt{total} \end{array} \right\} < \left\{ \begin{array}{l} \texttt{textwidth} \\ \texttt{textheight} \\ \texttt{text} \end{array} \right\} < \texttt{lines}.$$

For example,

```
\usepackage[hscale=0.8, textwidth=7in, width=18cm]{geometry}
```

is the same as `\usepackage[textwidth=7in]{geometry}`. Another example:

```
\usepackage[lines=30, scale=0.8, text=7in]{geometry}
```

results in `[lines=30, textwidth=7in]`.
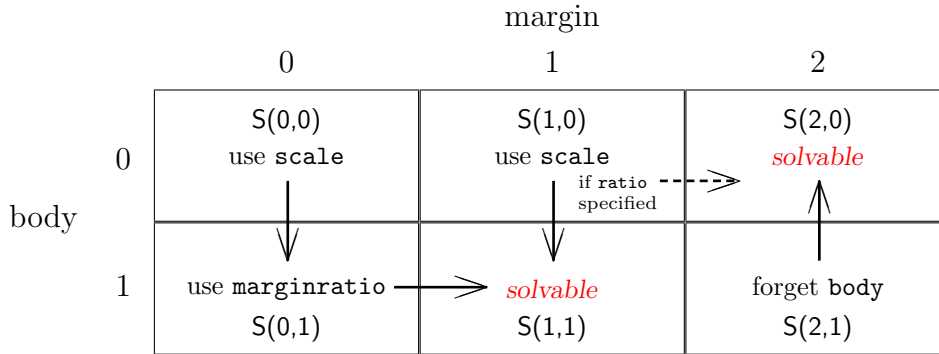
Figure 7: Specifications S(0,0) to S(2,1) and the completion rules (arrows). The horizonal and vertical denote the number of explicitly specified lengths for margin and body respectively. S(m,b) denote a specification with a set of the numbers (margin,body)=(m,b).

6.4 Defaults

This section sums up the default settings for the auto-completion described later.

The default vertical margin ratio is 2/3, namely,

$$\texttt{top} : \texttt{bottom} = 2 : 3 \qquad default. \tag{6}$$

As for the horizontal margin ratio, the default value depends on whether the document is onesided or twosided,

$$\texttt{left (inner)} : \texttt{right (outer)} = \left\{ \begin{array}{ll} 1 : 1 & \textit{default for oneside,} \\ 2 : 3 & \textit{default for twoside.} \end{array} \right. \tag{7}$$

Obviously the default horizontal margin ratio for oneside is 'centering'.

The geometry package has the following default setting for *onesided* documents:

- scale=0.7 (*body* is $0.7 \times paper$)
- marginratio={1:1, 2:3} (1:1 for horizontal and 2:3 for vertical margins)
- ignoreall (the header, footer, marginal notes are excluded when calculating the size of *body*.)

For *twosided* document with twoside option, the default setting is the same as *onesided* except that the horizontal margin ratio is set 2:3 as well.

Additional options overwrite the previous specified dimensions.

6.5 Auto-completion

Figure 7 shows schematically how many specification patterns exist and how to solve the ambiguity of the specifications. Each axis shows the numbers of lengths explicitly specified for body and margins. S(m,b) presents the specification with a set of numbers (margin,body)=(m,b).

For example, the specification width=14cm, left=3cm is categorized into S(1,1), which is an adequate specification. If you add right=4cm, it would be in S(2,1) and overspecified. if only width=14cm, it's in S(0,1), underspecified.

The geometry package has the auto-completion mechanism, in which if the layout parameters are underspecified or overspecified, geometry works out the ambiguity using the defaults and other relations. Here are the specifications and the completion rules.

S(0,0) Nothing is specified. The geometry package sets *body* with the default scale (=0.7). For example, width is set to be 0.7\paperwidth. Thus S(0,0) goes to S(0,1). See S(0,1).

S(0,1) Only *body* is specified, such as width=7in, lines=20, body={20cm,24cm}, scale=0.9 and so forth. Then geometry sets margins with the margin ratio. If the margin ratio is not specified, the default is used. The default vertical margin ratio is defined as

$$\texttt{top} : \texttt{bottom} = 2 : 3 \qquad default. \tag{8}$$

13

As for the horizontal margin ratio, the default value depends on whether the document is onesided or twosided,

$$\texttt{left (inner) : right (outer)} = \left\{ \begin{array}{ll} 1:1 & \textit{default for oneside,} \\ 2:3 & \textit{default for twoside.} \end{array} \right. \qquad (9)$$

For example, if `height=22cm` is specified on A4 paper, `geometry` calculates `top` margin as follows:

$$\begin{aligned} \texttt{top} \quad &= (\texttt{paperheight} - \texttt{height}) \times 2/5 \\ &= (29.7 - 22) \times 2/5 = 3.08 (\text{cm}) \end{aligned} \qquad (10)$$

Thus `top` margin and body `height` have been determined, the specification for the vertical goes to S(1,1) and all the parameters can be solved.

S(1,0)   Only one margin is specified, such as `bottom=2cm`, `left=1in`, `top=3cm`, and so forth.

- **If the margin ratio is _not_ specified**, geometry sets _body_ with the default `scale` ($= 0.7$). For example, if `top=2.4cm` is specified, geometry sets

$$\texttt{height} = 0.7\backslash\texttt{paperheight},$$

  then S(1,0) goes to S(1,1), in which `bottom` is calculated with $\texttt{paperheight} - (\texttt{height} + \texttt{top})$ and results in 6.51cm on A4 paper.

- **If the margin ratio is specified**, such as `hmarginratio={1:2}`, `vratio={3:4}` and so forth, geometry sets the other margin with the specified margin ratio. For example, if a set of options "`top=2.4cm, vratio={3:4}`" is specified, geometry sets `bottom` to be `3.2cm` calculating

$$\texttt{bottom} = \texttt{top}/3 \times 4 = 3.2\text{cm}$$

  Thus S(1,0) goes to S(2,0).

Notes that the version 4 or earlier used to set the other margin with the margin ratio. In the version 5, therefore, with the same specification, the result will be different from the one in the version 4. For example, if only `top=2.4cm` is specified, you got `bottom=2.4cm` in the version 4 or earlier, but you will get `bottom=6.51cm` in the version 5.

S(2,1)   The _body_ and two _margins_ are all specified, such as `vdivide={1in,8in,1.5in}`, "`left=3cm,width=13cm,right=4cm`" and so forth. Since `geometry` basically gives priority to _margins_ if dimensions are overspecified, `geometry` forgets and resets _body_. For example, if you specify

```
\usepackage[a4paper,left=3cm,width=13cm,right=4cm]{geometry},
```

`width` is reset to be 14cm because the width of a A4 paper is 21cm long.

7   Changing layout mid-document

The version 5 provides the new commands `\newgeometry{···}` and `\restoregeometry`, which allow you to change page dimensions in the middle of the document. Unlike `\geometry` in the preamble, `\newgeometry` is available only after `\begin{document}`, resets all the options ever specified except for the papersize-related options: `landscape`,`portrait`, and paper size options (such as `papersize`, `paper=a4paper` and so forth), which can't be changed with `\newgeometry`.

The command `\restoregeometry` restores the page layout specified in the preamble (before `\begin{document}`) with the options to `\usepackage{geometry}` and `\geometry`.

Note that both `\newgeometry` and `\restoregeometry` insert `\clearpage` where they are called.

Below is an example of changing layout mid-document. The layout L1 specified with `hmargin=3cm` (`left` and `right` margins are 3cm long) is changed to L2 with `left=3cm,right=1cm` and `bottom=0.1cm`. The layout L1 is restored with `\restoregeometry`.

```
\usepackage[hmargin=3cm]{geometry}
\begin{document}
```
Layout L1

```
\newgeometry{left=3cm,right=1cm,bottom=0.1cm}
```
Layout L2 (new)

```
\restoregeometry
```
Layout L1 (restored)

```
\newgeometry{margin=1cm,includefoot}
```
Layout L3 (new)

```
\end{document}
```
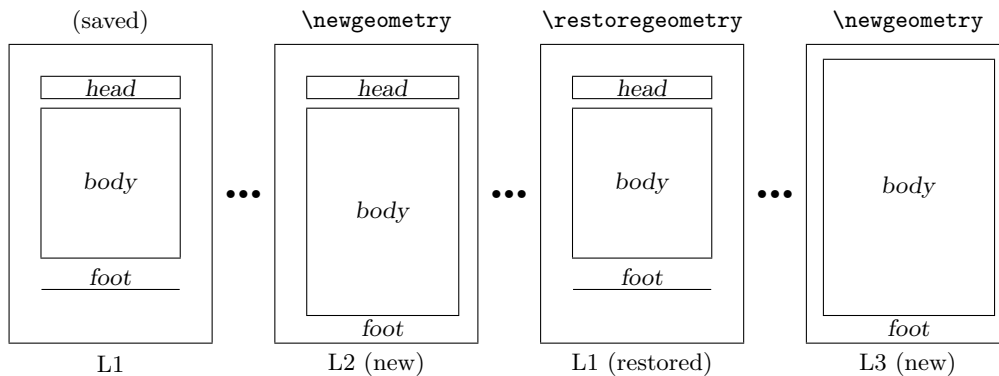

A set of commands \savegeometry{⟨*name*⟩} and \loadgeometry{⟨*name*⟩} is handy if you want to reuse more different layouts in your document. For example,

```
\usepackage[hmargin=3cm]{geometry}
\begin{document}
      L1
\newgeometry{left=3cm,right=1cm,bottom=0.1cm}
\savegeometry{L2}
      L2 (new, saved)
\restoregeometry
      L1 (restored)
\newgeometry{margin=1cm,includefoot}
      L3 (new)
\loadgeometry{L2}
      L2 (loaded)
\end{document}
```

8 Examples

- A onesided page layout with the text area centered in the paper. The examples below have the same result because the horizontal margin ratio is set `1:1` for oneside by default.

  - `centering`
  - `marginratio=1:1`
  - `vcentering`

- A twosided page layout with the inside offset for binding `1cm`.

  - `twoside, bindingoffset=1cm`

  In this case, `textwidth` is shorter than that of the default twosided document by $0.7 \times$ `1cm` ($=$`0.7cm`) because the default width of *body* is set with `scale=0.7` (which means width $= 0.7$\paperwidth).

- A layout with the left, right, and top margin `3cm`, `2cm` and `2.5in` respectively, with textheight of 40 lines, and with the head and foot of the page included in *total body*. The two examples below have the same result.

  - `left=3cm, right=2cm, lines=40, top=2.5in, includeheadfoot`
  - `hmargin={3cm,2cm}, tmargin=2.5in, lines=40, includeheadfoot`

- A layout with the height of *total body* `10in`, the bottom margin `2cm`, and the default width. The top margin will be calculated automatically. Each solution below results in the same page layout.

  - `vdivide={*, 10in, 2cm}`
  - `bmargin=2cm, height=10in`
  - `bottom=2cm, textheight=10in`

  Note that dimensions for *head* and *foot* are excluded from `height` of *total body*. An additional `includefoot` makes `\footskip` included in `totalheight`. Therefore, in the two cases below, `textheight` in the former layout is shorter than the latter (with 10in exactly) by `\footskip`. In other words, `height = textheight + footskip` when `includefoot=true` in this case.

  - `bmargin=2cm, height=10in, includefoot`
  - `bottom=2cm, textheight=10in, includefoot`

- A layout with `textwidth` and `textheight` 90% of the paper and with *body* centered. Each solution below results in the same page layout.

  - `scale=0.9, centering`
  - `text={.9\paperwidth,.9\paperheight}, ratio=1:1`
  - `width=.9\paperwidth, vmargin=.05\paperheight, marginratio=1:1`
  - `hdivide={*,0.9\paperwidth,*}, vdivide={*,0.9\paperheight,*}` (as for onesided documents)
  - `margin={0.05\paperwidth,0.05\paperheight}`

  You can add `heightrounded` to avoid an "underfull vbox warning" like

  ```
  Underfull \vbox (badness 10000) has occurred while \output is active.
  ```

  See Section 5.3 for the detail description about `heightrounded`.

- A layout with the width of marginal notes `3cm` and included in the width of *total body*. The following examples are the same.

  - `marginparwidth=3cm, includemp`
  - `marginpar=3cm, ignoremp=false`

- A layout the full scale *body* of the paper with A5 paper in landscape. The following examples are the same.

  - `a5paper, landscape, scale=1.0`
  - `landscape=TRUE, paper=a5paper, margin=0pt`

- A screen size layout appropriate to presentation with PC and video projector.

  ```
  \documentclass{slide}
  \usepackage[screen,margin=0.8in]{geometry}
   ...
  \begin{slide}
     ...
  \end{slide}
  ```

- A layout with fonts and spaces both enlarged from A4 to A3. In the case below, the resulted paper size is A3.

  - `a4paper, mag=1414.`

If you want to have a layout with two times bigger fonts, but without changing paper size, you can go

  – `letterpaper, mag=2000, truedimen`.

You can add `dvips` option, that is useful to preview it with proper paper size by `dviout` or `xdvi`.

- A complex page layout.

```
\usepackage[a5paper, landscape, twocolumn, twoside,
    left=2cm, hmarginratio=2:1, includemp, marginparwidth=43pt,
    bottom=1cm, foot=.7cm, includefoot, textheight=11cm, heightrounded,
    columnsep=1cm, dvips,  verbose]{geometry}
```

Try typesetting it and checking out the result yourself. `:-)`

9 Known problems

- With `pdftex=true`, mag $\neq$ 1000 and `truedimen`, `paperwidth` and `paperheight` shown in verbose mode are different from the real size of the resulted PDF. The PDF itself is correct anyway.

- With `pdftex=true`, mag $\neq$ 1000, *no* `truedimen`, and `hyperref`, `hyperref` should be loaded by `\usepackage` before `geometry`. Otherwise the resulted PDF size will become wrong.

- With `crop` package and mag $\neq$ 1000, `center` option of `crop` doesn't work well.

10 Acknowledgments

11 Implementation

1 ⟨∗package⟩

This package requires three other packages: keyval in LaTeX graphics bundle, ifpdf and ifvtex in 'oberdiek' bundle.

```
2 \RequirePackage{keyval}%
3 \RequirePackage{ifpdf}%
4 \RequirePackage{ifvtex}%
```

Internal switches are declared here.

```
5 \newif\ifGm@verbose
6 \newif\ifGm@landscape
7 \newif\ifGm@swap@papersize\Gm@swap@papersizefalse
8 \newif\ifGm@includehead
9 \newif\ifGm@includefoot
10 \newif\ifGm@includemp
11 \newif\ifGm@hbody
12 \newif\ifGm@vbody
13 \newif\ifGm@heightrounded
14 \newif\ifGm@showframe
15 \newif\ifGm@showcrop
16 \newif\ifGm@pass\Gm@passfalse
17 \newif\ifGm@resetpaper
18 \newif\ifGm@layout
19 \newif\ifGm@newgm
```

\Gm@cnth The counters for horizontal and vertical partitioning patterns.
\Gm@cntv
```
20 \newcount\Gm@cnth
21 \newcount\Gm@cntv
```

\c@Gm@tempcnt The counter is used to set number with calc.
```
22 \newcount\c@Gm@tempcnt
```

\Gm@bindingoffset The binding offset for the inner margin.
```
23 \newdimen\Gm@bindingoffset
```

\Gm@wd@mp Correction lengths for \textwidth, \oddsidemargin and \evensidemargin in includemp mode.
\Gm@odd@mp
\Gm@even@mp
```
24 \newdimen\Gm@wd@mp
25 \newdimen\Gm@odd@mp
26 \newdimen\Gm@even@mp
```

\Gm@layoutwidth The dimensions for the layout area.
\Gm@layoutheight
\Gm@layouthoffset
\Gm@layoutvoffset
```
27 \newdimen\Gm@layoutwidth
28 \newdimen\Gm@layoutheight
29 \newdimen\Gm@layouthoffset
30 \newdimen\Gm@layoutvoffset
```

\Gm@dimlist The token in which LaTeX native dimensions can be stored.
```
31 \newtoks\Gm@dimlist
```

\Gm@warning The macro to print warning messages.
```
32 \def\Gm@warning#1{\PackageWarningNoLine{geometry}{#1}}%
```

\ifGm@preamble The macro executes the option given as an argument only if it's specified in the preamble, as the options of \usepackage and/or the argument of \geometry. Otherwise, the macro would print the warning message and ignores the option setting.

```
33 \def\ifGm@preamble#1{%
34   \ifGm@newgm
35   \Gm@warning{'#1': not available in '\string\newgeometry'; skipped}%
36   \else
37     \expandafter\@firstofone
38   \fi}%
```

\Gm@Dhratio	The default values for the horizontal and vertical *marginalratio* are defined. \Gm@Dhratiotwo denotes
\Gm@Dhratiotwo	the default value of horizonal *marginratio* for twoside page layout with left and right margins swapped
\Gm@Dvratio	on verso pages, which is set by twoside.

```
39 \def\Gm@Dhratio{1:1}% = left:right default for oneside
40 \def\Gm@Dhratiotwo{2:3}% = inner:outer default for twoside.
41 \def\Gm@Dvratio{2:3}% = top:bottom default
```

\Gm@Dhscale	The default values for the horizontal and vertical *scale* are defined with 0.7.
\Gm@Dvscale	

```
42 \def\Gm@Dhscale{0.7}%
43 \def\Gm@Dvscale{0.7}%
```

\Gm@dvips	The driver names.
\Gm@dvipdfm	
\Gm@pdftex	
\Gm@xetex	
\Gm@vtex	

```
44 \def\Gm@dvips{dvips}%
45 \def\Gm@dvipdfm{dvipdfm}%
46 \def\Gm@pdftex{pdftex}%
47 \def\Gm@xetex{xetex}%
48 \def\Gm@vtex{vtex}%
```

\Gm@true	The macros for true and false.
\Gm@false	

```
49 \def\Gm@true{true}%
50 \def\Gm@false{false}%
```

\Gm@orgpw	These macros keep original paper (media) size intact.
\Gm@orgph	

```
51 \edef\Gm@orgpw{\the\paperwidth}%
52 \edef\Gm@orgph{\the\paperheight}%
```

\Gm@savelength	The macro saves the specified length to \Gm@restore.

```
53 \def\Gm@savelength#1{%
54   \g@addto@macro\Gm@restore{\expandafter\noexpand\expandafter\csname
55   #1\endcsname\expandafter=\expandafter\the\csname #1\endcsname\relax}}
```

\Gm@saveboolean	The macro saves the specified boolean to \Gm@restore.

```
56 \def\Gm@saveboolean#1{%
57   \csname if#1\endcsname
58     \g@addto@macro\Gm@restore{\expandafter\noexpand\csname #1true\endcsname}%
59   \else
60     \g@addto@macro\Gm@restore{\expandafter\noexpand\csname #1false\endcsname}%
61   \fi}%
```

\Gm@restore	The initialization for \Gm@restore.

```
62 \def\Gm@restore{}%
```

\Gm@save	The definition of the macro saving the real lengths LaTeX options.

```
63 \def\Gm@save{%
64   \Gm@savelength{paperwidth}%
65   \Gm@savelength{paperheight}%
66   \Gm@savelength{textwidth}%
67   \Gm@savelength{textheight}%
68   \Gm@savelength{evensidemargin}%
69   \Gm@savelength{oddsidemargin}%
70   \Gm@savelength{topmargin}%
71   \Gm@savelength{headheight}%
72   \Gm@savelength{headsep}%
73   \Gm@savelength{topskip}%
74   \Gm@savelength{footskip}%
75   \Gm@savelength{baselineskip}%
76   \Gm@savelength{marginparwidth}%
77   \Gm@savelength{marginparsep}%
78   \Gm@savelength{columnsep}%
79   \Gm@savelength{hoffset}%
80   \Gm@savelength{voffset}
81   \Gm@savelength{Gm@layouthoffset}%
82   \Gm@savelength{Gm@layoutvoffset}%
83   \Gm@saveboolean{@twocolumn}%
```

```
84    \Gm@saveboolean{@twoside}%
85    \Gm@saveboolean{@mparswitch}%
86    \Gm@saveboolean{@reversemargin}}%
```

\Gm@initnewgm   The macro initializes the parameters for layout in \newgeometry.

```
87 \def\Gm@initnewgm{%
88    \Gm@dimlist={}
89    \Gm@hbodyfalse
90    \Gm@vbodyfalse
91    \Gm@heightroundedfalse
92    \Gm@includeheadfalse
93    \Gm@includefootfalse
94    \Gm@includempfalse
95    \let\Gm@width\@undefined
96    \let\Gm@height\@undefined
97    \let\Gm@textwidth\@undefined
98    \let\Gm@textheight\@undefined
99    \let\Gm@lines\@undefined
100    \let\Gm@hscale\@undefined
101    \let\Gm@vscale\@undefined
102    \let\Gm@hmarginratio\@undefined
103    \let\Gm@vmarginratio\@undefined
104    \let\Gm@lmargin\@undefined
105    \let\Gm@rmargin\@undefined
106    \let\Gm@tmargin\@undefined
107    \let\Gm@bmargin\@undefined
108    \Gm@layoutfalse
109    \Gm@layouthoffset\z@
110    \Gm@layoutvoffset\z@
111    \Gm@bindingoffset\z@}%
```

\Gm@initall   This initialization is called as soon as the package is load It's also called as soon as reset option is specified.

```
112 \def\Gm@initall{%
113    \let\Gm@driver\@empty
114    \let\Gm@truedimen\@empty
115    \let\Gm@paper\@undefined
116    \Gm@resetpaperfalse
117    \Gm@landscapefalse
118    \Gm@verbosefalse
119    \Gm@showframefalse
120    \Gm@showcropfalse
121    \Gm@newgmfalse
122    \Gm@initnewgm}%
```

\Gm@setdriver   The macro sets the specified driver.

```
123 \def\Gm@setdriver#1{%
124    \expandafter\let\expandafter\Gm@driver\csname Gm@#1\endcsname}%
```

\Gm@unsetdriver   The macro unsets the specified driver if it has been set.

```
125 \def\Gm@unsetdriver#1{%
126    \expandafter\ifx\csname Gm@#1\endcsname\Gm@driver\let\Gm@driver\@empty\fi}%
```

\Gm@setbool   The macros for boolean option processing.
\Gm@setboolrev
```
127 \def\Gm@setbool{\@dblarg\Gm@@setbool}%
128 \def\Gm@setboolrev{\@dblarg\Gm@@setboolrev}%
129 \def\Gm@@setbool[#1]#2#3{\Gm@doif{#1}{#3}{\csname Gm@#2\Gm@bool\endcsname}}%
130 \def\Gm@@setboolrev[#1]#2#3{\Gm@doifelse{#1}{#3}%
131    {\csname Gm@#2\Gm@false\endcsname}{\csname Gm@#2\Gm@true\endcsname}}%
```

\Gm@doif   \Gm@doif excutes the third argument #3 using a boolean value #2 of a option #1. \Gm@doifelse
\Gm@doifelse   executes the third argument #3 if a boolean option #1 with its value #2 true, and executes the fourth argument #4 if false.

```
132 \def\Gm@doif#1#2#3{%
```

```
133    \lowercase{\def\Gm@bool{#2}}%
134    \ifx\Gm@bool\@empty
135      \let\Gm@bool\Gm@true
136    \fi
137    \ifx\Gm@bool\Gm@true
138    \else
139      \ifx\Gm@bool\Gm@false
140      \else
141        \let\Gm@bool\relax
142      \fi
143    \fi
144    \ifx\Gm@bool\relax
145      \Gm@warning{'#1' should be set to 'true' or 'false'}%
146    \else
147      #3
148    \fi}%
149 \def\Gm@doifelse#1#2#3#4{%
150    \Gm@doif{#1}{#2}{\ifx\Gm@bool\Gm@true #3\else #4\fi}}%
```

\Gm@reverse   The macro reverses a bool value.

```
151 \def\Gm@reverse#1{%
152    \csname ifGm@#1\endcsname
153    \csname Gm@#1false\endcsname\else\csname Gm@#1true\endcsname\fi}%
```

\Gm@defbylen   Macros \Gm@defbylen and \Gm@defbycnt can be used to define \Gm@xxxx variables by length and
\Gm@defbycnt   counter respectively with calc package.

```
154 \def\Gm@defbylen#1#2{%
155    \setlength\@tempdima{#2}%
156    \expandafter\edef\csname Gm@#1\endcsname{\the\@tempdima}}%
157 \def\Gm@defbycnt#1#2{%
158    \setcounter{Gm@tempcnt}{#2}%
159    \expandafter\edef\csname Gm@#1\endcsname{\the\value{Gm@tempcnt}}}%
```

\Gm@set@ratio   The macro parses the value of options specifying marginal ratios, which is used in \Gm@setbyratio
macro.

```
160 \def\Gm@sep@ratio#1:#2{\@tempcnta=#1\@tempcntb=#2}%
```

\Gm@setbyratio   The macro determines the dimension specified by #4 calculating #3$\times a/b$, where a and b are given
by \Gm@mratio with $a : b$ value. If #1 in brackets is b, a and b are swapped. The second argument
with h or v denoting horizontal or vertical is not used in this macro.

```
161 \def\Gm@setbyratio[#1]#2#3#4{% determine #4 by ratio
162    \expandafter\Gm@sep@ratio\Gm@mratio\relax
163    \if#1b
164      \edef\@@tempa{\the\@tempcnta}%
165      \@tempcnta=\@tempcntb
166      \@tempcntb=\@@tempa\relax
167    \fi
168    \expandafter\setlength\expandafter\@tempdimb\expandafter
169      {\csname Gm@#3\endcsname}%
170    \ifnum\@tempcntb>\z@
171      \multiply\@tempdimb\@tempcnta
172      \divide\@tempdimb\@tempcntb
173    \fi
174    \expandafter\edef\csname Gm@#4\endcsname{\the\@tempdimb}}%
```

\Gm@detiv   This macro determines the fourth length(#4) from #1(layoutwidth or layoutheight), #2 and #3. It is
used in \Gm@detall macro.

```
175 \def\Gm@detiv#1#2#3#4{% determine #4.
176    \expandafter\setlength\expandafter\@tempdima\expandafter
177      {\csname Gm@layout#1\endcsname}%
178    \expandafter\setlength\expandafter\@tempdimb\expandafter
179      {\csname Gm@#2\endcsname}%
180    \addtolength\@tempdima{-\@tempdimb}%
181    \expandafter\setlength\expandafter\@tempdimb\expandafter
```

21

```
182    {\csname Gm@#3\endcsname}%
183  \addtolength\@tempdima{-\@tempdimb}%
184  \ifdim\@tempdima<\z@
185    \Gm@warning{'#4' results in NEGATIVE (\the\@tempdima).%
186    ^^J\@spaces '#2' or '#3' should be shortened in length}%
187  \fi
188  \expandafter\edef\csname Gm@#4\endcsname{\the\@tempdima}}%
```

\Gm@detiiandiii This macro determines #2 and #3 from #1 with the first argument (#1) can be `width` or `height`, which is expanded into dimensions of paper and total body. It is used in \Gm@detall macro.

```
189  \def\Gm@detiiandiii#1#2#3{% determine #2 and #3.
190    \expandafter\setlength\expandafter\@tempdima\expandafter
191      {\csname Gm@layout#1\endcsname}%
192    \expandafter\setlength\expandafter\@tempdimb\expandafter
193      {\csname Gm@#1\endcsname}%
194    \addtolength\@tempdima{-\@tempdimb}%
195    \ifdim\@tempdima<\z@
196      \Gm@warning{'#2' and '#3' result in NEGATIVE (\the\@tempdima).%
197                  ^^J\@spaces '#1' should be shortened in length}%
198    \fi
199    \ifx\Gm@mratio\@undefined
200      \expandafter\Gm@sep@ratio\Gm@Dmratio\relax
201    \else
202      \expandafter\Gm@sep@ratio\Gm@mratio\relax
203      \ifnum\@tempcntb>\z@\else
204        \Gm@warning{margin ratio a:b should be non-zero; default used}%
205        \expandafter\Gm@sep@ratio\Gm@Dmratio\relax
206      \fi
207    \fi
208    \@tempdimb=\@tempdima
209    \advance\@tempcntb\@tempcnta
210    \divide\@tempdima\@tempcntb
211    \multiply\@tempdima\@tempcnta
212    \advance\@tempdimb-\@tempdima
213    \expandafter\edef\csname Gm@#2\endcsname{\the\@tempdima}%
214    \expandafter\edef\csname Gm@#3\endcsname{\the\@tempdimb}}%
```

\Gm@detall This macro determines partition of each direction. The first argument (#1) should be `h` or `v`, the second (#2) `width` or `height`, the third (#3) `lmargin` or `top`, and the last (#4) `rmargin` or `bottom`.

```
215  \def\Gm@detall#1#2#3#4{%
216    \@tempcnta\z@
217    \if#1h
218      \let\Gm@mratio\Gm@hmarginratio
219      \edef\Gm@Dmratio{\if@twoside\Gm@Dhratiotwo\else\Gm@Dhratio\fi}%
220    \else
221      \let\Gm@mratio\Gm@vmarginratio
222      \edef\Gm@Dmratio{\Gm@Dvratio}%
223    \fi
```

\@tempcnta is treated as a three-digit binary value with top, middle and bottom denoted `left(top)`, `width(height)` and `right(bottom)` margins user specified respectively.

```
224    \if#1h
225      \ifx\Gm@lmargin\@undefined\else\advance\@tempcnta4\relax\fi
226      \ifGm@hbody\advance\@tempcnta2\relax\fi
227      \ifx\Gm@rmargin\@undefined\else\advance\@tempcnta1\relax\fi
228      \Gm@cnth\@tempcnta
229    \else
230      \ifx\Gm@tmargin\@undefined\else\advance\@tempcnta4\relax\fi
231      \ifGm@vbody\advance\@tempcnta2\relax\fi
232      \ifx\Gm@bmargin\@undefined\else\advance\@tempcnta1\relax\fi
233      \Gm@cntv\@tempcnta
234    \fi
```

Case the value is `000` (=0) with nothing fixed (default):

```
235    \ifcase\@tempcnta
236      \if#1h
```

```
237       \edef\Gm@width{\Gm@Dhscale\Gm@layoutwidth}%
238     \else
239       \edef\Gm@height{\Gm@Dvscale\Gm@layoutheight}%
240     \fi
241     \Gm@detiiandiii{#2}{#3}{#4}%
```

Case 001 (=1) with right(bottom) fixed:

```
242     \or
243     \ifx\Gm@mratio\@undefined
244       \if#1h
245         \edef\Gm@width{\Gm@Dhscale\Gm@layoutwidth}%
246       \else
247         \edef\Gm@height{\Gm@Dvscale\Gm@layoutheight}%
248       \fi
249       \setlength\@tempdimc{\@nameuse{Gm@#4}}%
250       \Gm@detiiandiii{#2}{#3}{#4}%
251       \expandafter\let\csname Gm@#2\endcsname\@undefined
252       \expandafter\edef\csname Gm@#4\endcsname{\the\@tempdimc}%
253     \else
254       \Gm@setbyratio[f]{#1}{#4}{#3}%
255     \fi
256     \Gm@detiv{#2}{#3}{#4}{#2}%
```

Case 010 (=2) with width(height) fixed:

```
257     \or\Gm@detiiandiii{#2}{#3}{#4}%
```

Case 011 (=3) with both width(height) and right(bottom) fixed:

```
258     \or\Gm@detiv{#2}{#2}{#4}{#3}%
```

Case 100 (=4) with left(top) fixed:

```
259     \or
260     \ifx\Gm@mratio\@undefined
261       \if#1h
262         \edef\Gm@width{\Gm@Dhscale\Gm@layoutwidth}%
263       \else
264         \edef\Gm@height{\Gm@Dvscale\Gm@layoutheight}%
265       \fi
266       \setlength\@tempdimc{\@nameuse{Gm@#3}}%
267       \Gm@detiiandiii{#2}{#4}{#3}%
268       \expandafter\let\csname Gm@#2\endcsname\@undefined
269       \expandafter\edef\csname Gm@#3\endcsname{\the\@tempdimc}%
270     \else
271       \Gm@setbyratio[b]{#1}{#3}{#4}%
272     \fi
273     \Gm@detiv{#2}{#3}{#4}{#2}%
```

Case 101 (=5) with both left(top) and right(bottom) fixed:

```
274     \or\Gm@detiv{#2}{#3}{#4}{#2}%
```

Case 110 (=6) with both left(top) and width(height) fixed:

```
275     \or\Gm@detiv{#2}{#2}{#3}{#4}%
```

Case 111 (=7) with all fixed though it is over-specified:

```
276     \or\Gm@warning{Over-specification in `#1'-direction.%
277                     ^^J\@spaces `#2' (\@nameuse{Gm@#2}) is ignored}%
278     \Gm@detiv{#2}{#3}{#4}{#2}%
279     \else\fi}%
```

\Gm@clean   The macro for setting unspecified dimensions to be \@undefined. This is used by \geometry macro.

```
280 \def\Gm@clean{%
281     \ifnum\Gm@cnth<4\let\Gm@lmargin\@undefined\fi
282     \ifodd\Gm@cnth\else\let\Gm@rmargin\@undefined\fi
283     \ifnum\Gm@cntv<4\let\Gm@tmargin\@undefined\fi
284     \ifodd\Gm@cntv\else\let\Gm@bmargin\@undefined\fi
285     \ifGm@hbody\else
286       \let\Gm@hscale\@undefined
287       \let\Gm@width\@undefined
288       \let\Gm@textwidth\@undefined
```

```
289    \fi
290    \ifGm@vbody\else
291      \let\Gm@vscale\@undefined
292      \let\Gm@height\@undefined
293      \let\Gm@textheight\@undefined
294    \fi
295    }%
```

\Gm@parse@divide   The macro parses (h,v)divide options.

```
296  \def\Gm@parse@divide#1#2#3#4{%
297    \def\Gm@star{*}%
298    \@tempcnta\z@
299    \@for\Gm@tmp:=#1\do{%
300      \expandafter\KV@@sp@def\expandafter\Gm@frag\expandafter{\Gm@tmp}%
301      \edef\Gm@value{\Gm@frag}%
302      \ifcase\@tempcnta\relax\edef\Gm@key{#2}%
303        \or\edef\Gm@key{#3}%
304        \else\edef\Gm@key{#4}%
305      \fi
306      \@nameuse{Gm@set\Gm@key false}%
307      \ifx\empty\Gm@value\else
308      \ifx\Gm@star\Gm@value\else
309        \setkeys{Gm}{\Gm@key=\Gm@value}%
310      \fi\fi
311      \advance\@tempcnta\@ne}%
312    \let\Gm@star\relax}%
```

\Gm@branch   The macro splits a value into the same two values.

```
313  \def\Gm@branch#1#2#3{%
314    \@tempcnta\z@
315    \@for\Gm@tmp:=#1\do{%
316      \KV@@sp@def\Gm@frag{\Gm@tmp}%
317      \edef\Gm@value{\Gm@frag}%
318      \ifcase\@tempcnta\relax% cnta == 0
319        \setkeys{Gm}{#2=\Gm@value}%
320      \or% cnta == 1
321        \setkeys{Gm}{#3=\Gm@value}%
322      \else\fi
323      \advance\@tempcnta\@ne}%
324    \ifnum\@tempcnta=\@ne
325      \setkeys{Gm}{#3=\Gm@value}%
326    \fi}%
```

\Gm@magtooffset   This macro is used to adjust offsets by \mag.

```
327  \def\Gm@magtooffset{%
328    \@tempdima=\mag\Gm@truedimen sp%
329    \@tempdimb=1\Gm@truedimen in%
330    \divide\@tempdimb\@tempdima
331    \multiply\@tempdimb\@m
332    \addtolength{\hoffset}{1\Gm@truedimen in}%
333    \addtolength{\voffset}{1\Gm@truedimen in}%
334    \addtolength{\hoffset}{-\the\@tempdimb}%
335    \addtolength{\voffset}{-\the\@tempdimb}}%
```

\Gm@setlength   This macro stores LaTeX native dimensions, which are stored and set afterwards.

```
336  \def\Gm@setlength#1#2{%
337    \let\Gm@len=\relax\let\Gm@td=\relax
338    \edef\addtolist{\noexpand\Gm@dimlist=%
339    {\the\Gm@dimlist \Gm@len{#1}{#2}}}\addtolist}%
```

\Gm@expandlengths   This macro processes \Gm@dimlist.

```
340  \def\Gm@expandlengths{%
341    \def\Gm@td{\Gm@truedimen}%
342    \def\Gm@len##1##2{\setlength{##1}{##2}}%
343    \the\Gm@dimlist}%
```

\Gm@setsize The macro sets `paperwidth` and `paperheight` dimensions using `\Gm@setlength` macro.

```
344 \def\Gm@setsize#1(#2,#3)#4{%
345   \let\Gm@td\relax
346   \expandafter\Gm@setlength\csname #1width\endcsname{#2\Gm@td #4}%
347   \expandafter\Gm@setlength\csname #1height\endcsname{#3\Gm@td #4}%
348   \ifGm@landscape\Gm@swap@papersizetrue\else\Gm@swap@papersizefalse\fi}%
```

\Gm@setpaper@ifpre The macro changes the paper size.

```
349 \def\Gm@setpaper@ifpre#1{%
350   \ifGm@preamble{#1}{\def\Gm@paper{#1}\@nameuse{Gm@#1}{paper}}}%
```

Various paper size are defined here.

```
351 \@namedef{Gm@a0paper}#1{\Gm@setsize{#1}(841,1189){mm}}% ISO A0
352 \@namedef{Gm@a1paper}#1{\Gm@setsize{#1}(594,841){mm}}% ISO A1
353 \@namedef{Gm@a2paper}#1{\Gm@setsize{#1}(420,594){mm}}% ISO A2
354 \@namedef{Gm@a3paper}#1{\Gm@setsize{#1}(297,420){mm}}% ISO A3
355 \@namedef{Gm@a4paper}#1{\Gm@setsize{#1}(210,297){mm}}% ISO A4
356 \@namedef{Gm@a5paper}#1{\Gm@setsize{#1}(148,210){mm}}% ISO A5
357 \@namedef{Gm@a6paper}#1{\Gm@setsize{#1}(105,148){mm}}% ISO A6
358 \@namedef{Gm@b0paper}#1{\Gm@setsize{#1}(1000,1414){mm}}% ISO B0
359 \@namedef{Gm@b1paper}#1{\Gm@setsize{#1}(707,1000){mm}}% ISO B1
360 \@namedef{Gm@b2paper}#1{\Gm@setsize{#1}(500,707){mm}}% ISO B2
361 \@namedef{Gm@b3paper}#1{\Gm@setsize{#1}(353,500){mm}}% ISO B3
362 \@namedef{Gm@b4paper}#1{\Gm@setsize{#1}(250,353){mm}}% ISO B4
363 \@namedef{Gm@b5paper}#1{\Gm@setsize{#1}(176,250){mm}}% ISO B5
364 \@namedef{Gm@b6paper}#1{\Gm@setsize{#1}(125,176){mm}}% ISO B6
365 \@namedef{Gm@b0j}#1{\Gm@setsize{#1}(1030,1456){mm}}% JIS B0
366 \@namedef{Gm@b1j}#1{\Gm@setsize{#1}(728,1030){mm}}% JIS B1
367 \@namedef{Gm@b2j}#1{\Gm@setsize{#1}(515,728){mm}}% JIS B2
368 \@namedef{Gm@b3j}#1{\Gm@setsize{#1}(364,515){mm}}% JIS B3
369 \@namedef{Gm@b4j}#1{\Gm@setsize{#1}(257,364){mm}}% JIS B4
370 \@namedef{Gm@b5j}#1{\Gm@setsize{#1}(182,257){mm}}% JIS B5
371 \@namedef{Gm@b6j}#1{\Gm@setsize{#1}(128,182){mm}}% JIS B6
372 \@namedef{Gm@ansiapaper}#1{\Gm@setsize{#1}(8.5,11){in}}%
373 \@namedef{Gm@ansibpaper}#1{\Gm@setsize{#1}(11,17){in}}%
374 \@namedef{Gm@ansicpaper}#1{\Gm@setsize{#1}(17,22){in}}%
375 \@namedef{Gm@ansidpaper}#1{\Gm@setsize{#1}(22,34){in}}%
376 \@namedef{Gm@ansiepaper}#1{\Gm@setsize{#1}(34,44){in}}%
377 \@namedef{Gm@letterpaper}#1{\Gm@setsize{#1}(8.5,11){in}}%
378 \@namedef{Gm@legalpaper}#1{\Gm@setsize{#1}(8.5,14){in}}%
379 \@namedef{Gm@executivepaper}#1{\Gm@setsize{#1}(7.25,10.5){in}}%
380 \@namedef{Gm@screen}#1{\Gm@setsize{#1}(225,180){mm}}%
```

'paper' `paper` takes a paper name as its value.

```
381 \define@key{Gm}{paper}{\setkeys{Gm}{#1}}%
382 \let\KV@Gm@papername\KV@Gm@paper
```

'a[0-6]paper' The following paper names are available.
'b[0-6]paper'
'b[0-6]j'
'ansi[a-e]paper'
'letterpaper'
'legalpaper'
'executivepaper'
'screen'

```
383 \define@key{Gm}{a0paper}[true]{\Gm@setpaper@ifpre{a0paper}}%
384 \define@key{Gm}{a1paper}[true]{\Gm@setpaper@ifpre{a1paper}}%
385 \define@key{Gm}{a2paper}[true]{\Gm@setpaper@ifpre{a2paper}}%
386 \define@key{Gm}{a3paper}[true]{\Gm@setpaper@ifpre{a3paper}}%
387 \define@key{Gm}{a4paper}[true]{\Gm@setpaper@ifpre{a4paper}}%
388 \define@key{Gm}{a5paper}[true]{\Gm@setpaper@ifpre{a5paper}}%
389 \define@key{Gm}{a6paper}[true]{\Gm@setpaper@ifpre{a6paper}}%
390 \define@key{Gm}{b0paper}[true]{\Gm@setpaper@ifpre{b0paper}}%
391 \define@key{Gm}{b1paper}[true]{\Gm@setpaper@ifpre{b1paper}}%
392 \define@key{Gm}{b2paper}[true]{\Gm@setpaper@ifpre{b2paper}}%
393 \define@key{Gm}{b3paper}[true]{\Gm@setpaper@ifpre{b3paper}}%
394 \define@key{Gm}{b4paper}[true]{\Gm@setpaper@ifpre{b4paper}}%
395 \define@key{Gm}{b5paper}[true]{\Gm@setpaper@ifpre{b5paper}}%
396 \define@key{Gm}{b6paper}[true]{\Gm@setpaper@ifpre{b6paper}}%
397 \define@key{Gm}{b0j}[true]{\Gm@setpaper@ifpre{b0j}}%
398 \define@key{Gm}{b1j}[true]{\Gm@setpaper@ifpre{b1j}}%
399 \define@key{Gm}{b2j}[true]{\Gm@setpaper@ifpre{b2j}}%
```

```
400 \define@key{Gm}{b3j}[true]{\Gm@setpaper@ifpre{b3j}}%
401 \define@key{Gm}{b4j}[true]{\Gm@setpaper@ifpre{b4j}}%
402 \define@key{Gm}{b5j}[true]{\Gm@setpaper@ifpre{b5j}}%
403 \define@key{Gm}{b6j}[true]{\Gm@setpaper@ifpre{b6j}}%
404 \define@key{Gm}{ansiapaper}[true]{\Gm@setpaper@ifpre{ansiapaper}}%
405 \define@key{Gm}{ansibpaper}[true]{\Gm@setpaper@ifpre{ansibpaper}}%
406 \define@key{Gm}{ansicpaper}[true]{\Gm@setpaper@ifpre{ansicpaper}}%
407 \define@key{Gm}{ansidpaper}[true]{\Gm@setpaper@ifpre{ansidpaper}}%
408 \define@key{Gm}{ansiepaper}[true]{\Gm@setpaper@ifpre{ansiepaper}}%
409 \define@key{Gm}{letterpaper}[true]{\Gm@setpaper@ifpre{letterpaper}}%
410 \define@key{Gm}{legalpaper}[true]{\Gm@setpaper@ifpre{legalpaper}}%
411 \define@key{Gm}{executivepaper}[true]{\Gm@setpaper@ifpre{executivepaper}}%
412 \define@key{Gm}{screen}[true]{\Gm@setpaper@ifpre{screen}}%
```

`paperwidth` Direct specification for paper size is also possible.
`paperheight`
`papersize`
```
413 \define@key{Gm}{paperwidth}{\ifGm@preamble{paperwidth}{%
414   \Gm@setlength\paperwidth{#1}}}%
415 \define@key{Gm}{paperheight}{\ifGm@preamble{paperheight}{%
416   \Gm@setlength\paperheight{#1}}}%
417 \define@key{Gm}{papersize}{\ifGm@preamble{papersize}{%
418   \Gm@branch{#1}{paperwidth}{paperheight}}}%
```

`layout` Direct specification for layout size is also possible.
`layoutwidth`
`layoutheight`
`layoutsize`
```
419 \define@key{Gm}{layout}{\Gm@layouttrue\@nameuse{Gm@#1}{Gm@layout}}%
420 \let\KV@Gm@layoutname\KV@Gm@layout
421 \define@key{Gm}{layoutwidth}{\Gm@layouttrue\Gm@setlength\Gm@layoutwidth{#1}}%
422 \define@key{Gm}{layoutheight}{\Gm@layouttrue\Gm@setlength\Gm@layoutheight{#1}}%
423 \define@key{Gm}{layoutsize}{\Gm@branch{#1}{layoutwidth}{layoutheight}}%
```

`landscape` Paper orientation setting.
`portrait`
```
424 \define@key{Gm}{landscape}[true]{\ifGm@preamble{landscape}{%
425   \Gm@doifelse{landscape}{#1}%
426   {\ifGm@landscape\else\Gm@landscapetrue\Gm@reverse{swap@papersize}\fi}%
427   {\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}}}%
428 \define@key{Gm}{portrait}[true]{\ifGm@preamble{portrait}{%
429   \Gm@doifelse{portrait}{#1}%
430   {\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}%
431   {\ifGm@landscape\else\Gm@landscapetrue\Gm@reverse{swap@papersize}\fi}}}%
```

`hscale` These options can determine the length(s) of *total body* giving *scale(s)* against the paper size.
`vscale`
`scale`
```
432 \define@key{Gm}{hscale}{\Gm@hbodytrue\edef\Gm@hscale{#1}}%
433 \define@key{Gm}{vscale}{\Gm@vbodytrue\edef\Gm@vscale{#1}}%
434 \define@key{Gm}{scale}{\Gm@branch{#1}{hscale}{vscale}}%
```

`width` These options give concrete dimension(s) of *total body*. totalwidth and totalheight are aliases of
`height` width and height respectively.
`total`
`totalwidth`
`totalheight`
```
435 \define@key{Gm}{width}{\Gm@hbodytrue\Gm@defbylen{width}{#1}}%
436 \define@key{Gm}{height}{\Gm@vbodytrue\Gm@defbylen{height}{#1}}%
437 \define@key{Gm}{total}{\Gm@branch{#1}{width}{height}}%
438 \let\KV@Gm@totalwidth\KV@Gm@width
439 \let\KV@Gm@totalheight\KV@Gm@height
```

`textwidth` These options directly sets the dimensions \textwidth and \textheight. body is an alias of text.
`textheight`
`text`
`body`
```
440 \define@key{Gm}{textwidth}{\Gm@hbodytrue\Gm@defbylen{textwidth}{#1}}%
441 \define@key{Gm}{textheight}{\Gm@vbodytrue\Gm@defbylen{textheight}{#1}}%
442 \define@key{Gm}{text}{\Gm@branch{#1}{textwidth}{textheight}}%
443 \let\KV@Gm@body\KV@Gm@text
```

`lines` The option sets \textheight with the number of lines.
```
444 \define@key{Gm}{lines}{\Gm@vbodytrue\Gm@defbycnt{lines}{#1}}%
```

`includehead` The options take the corresponding dimensions as part of *body*.
`includefoot`
`includeheadfoot`
`includemp`
`includeall`
```
445 \define@key{Gm}{includehead}[true]{\Gm@setbool{includehead}{#1}}%
446 \define@key{Gm}{includefoot}[true]{\Gm@setbool{includefoot}{#1}}%
```

```
447 \define@key{Gm}{includeheadfoot}[true]{\Gm@doifelse{includeheadfoot}{#1}%
448   {\Gm@includeheadtrue\Gm@includefoottrue}%
449   {\Gm@includeheadfalse\Gm@includefootfalse}}%
450 \define@key{Gm}{includemp}[true]{\Gm@setbool{includemp}{#1}}%
451 \define@key{Gm}{includeall}[true]{\Gm@doifelse{includeall}{#1}%
452   {\Gm@includeheadtrue\Gm@includefoottrue\Gm@includemptrue}%
453   {\Gm@includeheadfalse\Gm@includefootfalse\Gm@includempfalse}}%
```

'ignorehead' These options exclude *head*, *foot* and *marginpars* when determining *body*.
'ignorefoot'
'ignoreheadfoot'
'ignoremp'
'ignoreall'

```
454 \define@key{Gm}{ignorehead}[true]{%
455   \Gm@setboolrev[ignorehead]{includehead}{#1}}%
456 \define@key{Gm}{ignorefoot}[true]{%
457   \Gm@setboolrev[ignorefoot]{includefoot}{#1}}%
458 \define@key{Gm}{ignoreheadfoot}[true]{\Gm@doifelse{ignoreheadfoot}{#1}%
459   {\Gm@includeheadfalse\Gm@includefootfalse}%
460   {\Gm@includeheadtrue\Gm@includefoottrue}}%
461 \define@key{Gm}{ignoremp}[true]{%
462   \Gm@setboolrev[ignoremp]{includemp}{#1}}%
463 \define@key{Gm}{ignoreall}[true]{\Gm@doifelse{ignoreall}{#1}%
464   {\Gm@includeheadfalse\Gm@includefootfalse\Gm@includempfalse}%
465   {\Gm@includeheadtrue\Gm@includefoottrue\Gm@includemptrue}}%
```

'heightrounded' The option rounds \textheight to n-times of \baselineskip plus \topskip.

```
466 \define@key{Gm}{heightrounded}[true]{\Gm@setbool{heightrounded}{#1}}%
```

'hdivide' The options are useful to specify partitioning in each direction of the paper.
'vdivide'
'divide'

```
467 \define@key{Gm}{hdivide}{\Gm@parse@divide{#1}{lmargin}{width}{rmargin}}%
468 \define@key{Gm}{vdivide}{\Gm@parse@divide{#1}{tmargin}{height}{bmargin}}%
469 \define@key{Gm}{divide}{\Gm@parse@divide{#1}{lmargin}{width}{rmargin}%
470   \Gm@parse@divide{#1}{tmargin}{height}{bmargin}}%
```

'lmargin' These options set *margins*. left, inner, innermargin are aliases of lmargin. right, outer,
'rmargin' outermargin are aliases of rmargin. top and bottom are aliases of tmargin and bmargin respec-
'tmargin' tively.
'bmargin'
'left'
'inner'
'innermargin'
'right'
'outer'
'outermargin'
'top'
'bottom'

```
471 \define@key{Gm}{lmargin}{\Gm@defbylen{lmargin}{#1}}%
472 \define@key{Gm}{rmargin}{\Gm@defbylen{rmargin}{#1}}%
473 \let\KV@Gm@left\KV@Gm@lmargin
474 \let\KV@Gm@inner\KV@Gm@lmargin
475 \let\KV@Gm@innermargin\KV@Gm@lmargin
476 \let\KV@Gm@right\KV@Gm@rmargin
477 \let\KV@Gm@outer\KV@Gm@rmargin
478 \let\KV@Gm@outermargin\KV@Gm@rmargin
479 \define@key{Gm}{tmargin}{\Gm@defbylen{tmargin}{#1}}%
480 \define@key{Gm}{bmargin}{\Gm@defbylen{bmargin}{#1}}%
481 \let\KV@Gm@top\KV@Gm@tmargin
482 \let\KV@Gm@bottom\KV@Gm@bmargin
```

'hmargin' These options are shorthands for setting *margins*.
'vmargin'
'margin'

```
483 \define@key{Gm}{hmargin}{\Gm@branch{#1}{lmargin}{rmargin}}%
484 \define@key{Gm}{vmargin}{\Gm@branch{#1}{tmargin}{bmargin}}%
485 \define@key{Gm}{margin}{\Gm@branch{#1}{lmargin}{tmargin}%
486   \Gm@branch{#1}{rmargin}{bmargin}}%
```

'hmarginratio' Options specifying the margin ratios.
'vmarginratio'
'marginratio'
'hratio'
'vratio'
'ratio'

```
487 \define@key{Gm}{hmarginratio}{\edef\Gm@hmarginratio{#1}}%
488 \define@key{Gm}{vmarginratio}{\edef\Gm@vmarginratio{#1}}%
489 \define@key{Gm}{marginratio}{\Gm@branch{#1}{hmarginratio}{vmarginratio}}%
490 \let\KV@Gm@hratio\KV@Gm@hmarginratio
491 \let\KV@Gm@vratio\KV@Gm@vmarginratio
492 \let\KV@Gm@ratio\KV@Gm@marginratio
```

'hcentering' Useful shorthands to place *body* centered.
'vcentering'
'centering'

```
493 \define@key{Gm}{hcentering}[true]{\Gm@doifelse{hcentering}{#1}%
494   {\def\Gm@hmarginratio{1:1}}{}}%
```

```
495 \define@key{Gm}{vcentering}[true]{\Gm@doifelse{vcentering}{#1}%
496   {\def\Gm@vmarginratio{1:1}}{}}%
497 \define@key{Gm}{centering}[true]{\Gm@doifelse{centering}{#1}%
498   {\def\Gm@hmarginratio{1:1}\def\Gm@vmarginratio{1:1}}{}}%
```

'twoside'   If twoside=true, \@twoside and \@mparswitch is set to true.

```
499 \define@key{Gm}{twoside}[true]{\Gm@doifelse{twoside}{#1}%
500   {\@twosidetrue\@mparswitchtrue}{\@twosidefalse\@mparswitchfalse}}%
```

'asymmetric'   asymmetric sets \@mparswitchfalse and \@twosidetrue A asymmetric=false has no effect.

```
501 \define@key{Gm}{asymmetric}[true]{\Gm@doifelse{asymmetric}{#1}%
502   {\@twosidetrue\@mparswitchfalse}{}}%
```

'bindingoffset'   The macro adds the specified space to the inner margin.

```
503 \define@key{Gm}{bindingoffset}{\Gm@setlength\Gm@bindingoffset{#1}}%
```

'headheight'   The direct settings of *head* and/or *foot* dimensions.
'headsep'
'footskip'
'head'
'foot'

```
504 \define@key{Gm}{headheight}{\Gm@setlength\headheight{#1}}%
505 \define@key{Gm}{headsep}{\Gm@setlength\headsep{#1}}%
506 \define@key{Gm}{footskip}{\Gm@setlength\footskip{#1}}%
507 \let\KV@Gm@head\KV@Gm@headheight
508 \let\KV@Gm@foot\KV@Gm@footskip
```

'nohead'   They are only shorthands to set *head* and/or *foot* to be 0pt.
'nofoot'
'noheadfoot'

```
509 \define@key{Gm}{nohead}[true]{\Gm@doifelse{nohead}{#1}%
510   {\Gm@setlength\headheight\z@\Gm@setlength\headsep\z@}{}}%
511 \define@key{Gm}{nofoot}[true]{\Gm@doifelse{nofoot}{#1}%
512   {\Gm@setlength\footskip\z@}{}}%
513 \define@key{Gm}{noheadfoot}[true]{\Gm@doifelse{noheadfoot}{#1}%
514   {\Gm@setlength\headheight\z@\Gm@setlength\headsep
515   \z@\Gm@setlength\footskip\z@}{}}%
```

'footnotesep'   The option directly sets a native dimension \footnotesep.

```
516 \define@key{Gm}{footnotesep}{\Gm@setlength{\skip\footins}{#1}}%
```

'marginparwidth'   They directly set native dimensions \marginparwidth and \marginparsep.
'marginpar'
'marginparsep'

```
517 \define@key{Gm}{marginparwidth}{\Gm@setlength\marginparwidth{#1}}%
518 \let\KV@Gm@marginpar\KV@Gm@marginparwidth
519 \define@key{Gm}{marginparsep}{\Gm@setlength\marginparsep{#1}}%
```

'nomarginpar'   The macro is a shorthand for \marginparwidth=0pt and \marginparsep=0pt.

```
520 \define@key{Gm}{nomarginpar}[true]{\Gm@doifelse{nomarginpar}{#1}%
521   {\Gm@setlength\marginparwidth\z@\Gm@setlength\marginparsep\z@}{}}%
```

'columnsep'   The option sets a native dimension \columnsep.

```
522 \define@key{Gm}{columnsep}{\Gm@setlength\columnsep{#1}}%
```

'hoffset'   The former two options set native dimensions \hoffset and \voffset. offset can set both of them
'voffset'   with the same value.
'offset'

```
523 \define@key{Gm}{hoffset}{\Gm@setlength\hoffset{#1}}%
524 \define@key{Gm}{voffset}{\Gm@setlength\voffset{#1}}%
525 \define@key{Gm}{offset}{\Gm@branch{#1}{hoffset}{voffset}}%
```

'layouthoffset'
'layoutvoffset'
'layoutoffset'

```
526 \define@key{Gm}{layouthoffset}{\Gm@setlength\Gm@layouthoffset{#1}}%
527 \define@key{Gm}{layoutvoffset}{\Gm@setlength\Gm@layoutvoffset{#1}}%
528 \define@key{Gm}{layoutoffset}{\Gm@branch{#1}{layouthoffset}{layoutvoffset}}%
```

'twocolumn'   The option sets \twocolumn switch.

```
529 \define@key{Gm}{twocolumn}[true]{%
530   \Gm@doif{twocolumn}{#1}{\csname @twocolumn\Gm@bool\endcsname}}%
```

'onecolumn' This option has the reverse effect of twocolumn option.

```
531 \define@key{Gm}{onecolumn}[true]{%
532   \Gm@doifelse{onecolumn}{#1}{\@twocolumnfalse}{\@twocolumntrue}}%
```

'reversemp' The both options set \reversemargin.
'reversemarginpar'

```
533 \define@key{Gm}{reversemp}[true]{%
534   \Gm@doif{reversemp}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%
535 \define@key{Gm}{reversemarginpar}[true]{%
536   \Gm@doif{reversemarginpar}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%
```

'dviver'

```
537 \define@key{Gm}{driver}{\ifGm@preamble{driver}{%
538   \edef\@@tempa{#1}\edef\@@auto{auto}\edef\@@none{none}%
539   \ifx\@@tempa\@empty\let\Gm@driver\relax\else
540   \ifx\@@tempa\@@none\let\Gm@driver\relax\else
541   \ifx\@@tempa\@@auto\let\Gm@driver\@empty\else
542   \setkeys{Gm}{#1}\fi\fi\fi\let\@@auto\relax\let\@@none\relax}}%
```

'dvips' The geometry package supports dvips, dvipdfm, pdflatex and vtex. dvipdfm works like dvips.
'dvipdfm'
'pdftex'
'xetex'
'vtex'

```
543 \define@key{Gm}{dvips}[true]{\ifGm@preamble{dvips}{%
544   \Gm@doifelse{dvips}{#1}{\Gm@setdriver{dvips}}{\Gm@unsetdriver{dvips}}}}%
545 \define@key{Gm}{dvipdfm}[true]{\ifGm@preamble{dvipdfm}{%
546   \Gm@doifelse{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{\Gm@unsetdriver{dvipdfm}}}}%
547 \define@key{Gm}{pdftex}[true]{\ifGm@preamble{pdftex}{%
548   \Gm@doifelse{pdftex}{#1}{\Gm@setdriver{pdftex}}{\Gm@unsetdriver{pdftex}}}}%
549 \define@key{Gm}{xetex}[true]{\ifGm@preamble{xetex}{%
550   \Gm@doifelse{xetex}{#1}{\Gm@setdriver{xetex}}{\Gm@unsetdriver{xetex}}}}%
551 \define@key{Gm}{vtex}[true]{\ifGm@preamble{vtex}{%
552   \Gm@doifelse{vtex}{#1}{\Gm@setdriver{vtex}}{\Gm@unsetdriver{vtex}}}}%
```

'verbose' The verbose mode.

```
553 \define@key{Gm}{verbose}[true]{\ifGm@preamble{verbose}{\Gm@setbool{verbose}{#1}}}%
```

'reset' The option cancels all the options specified before reset, except pass. mag ($\neq 1000$) with truedimen cannot be also reset.

```
554 \define@key{Gm}{reset}[true]{\ifGm@preamble{reset}{%
555   \Gm@doifelse{reset}{#1}{\Gm@restore@org\Gm@initall
556   \ProcessOptionsKV[c]{Gm}\Gm@setdefaultpaper}{}}}%
```

'resetpaper' If resetpaper is set to true, the paper size redefined in the package is discarded and the original one is restored. This option may be useful to print nonstandard sized documents with normal printers and papers.

```
557 \define@key{Gm}{resetpaper}[true]{\ifGm@preamble{resetpaper}{%
558   \Gm@setbool{resetpaper}{#1}}}%
```

'mag' mag is expanded immediately when it is specified. So reset can't reset mag when it is set with truedimen.

```
559 \define@key{Gm}{mag}{\ifGm@preamble{mag}{\mag=#1}}%
```

'truedimen' If truedimen is set to true, all of the internal explicit dimensions is changed to *true* dimensions, e.g., 1in is changed to 1truein.

```
560 \define@key{Gm}{truedimen}[true]{\ifGm@preamble{truedimen}{%
561   \Gm@doifelse{truedimen}{#1}{\let\Gm@truedimen\Gm@true}%
562   {\let\Gm@truedimen\@empty}}}%
```

'pass' The option makes all the options specified ineffective except verbose switch.

```
563 \define@key{Gm}{pass}[true]{\ifGm@preamble{pass}{\Gm@setbool{pass}{#1}}}%
```

'showframe' The showframe option prints page frames to help you understand what the resulting layout is like.

```
564 \define@key{Gm}{showframe}[true]{\Gm@setbool{showframe}{#1}}%
```

'showcrop' The showcrop option prints cropmarks at the corners.

```
565 \define@key{Gm}{showcrop}[true]{\Gm@setbool{showcrop}{#1}}%
```

\Gm@setdefaultpaper The macro stores paper dimensions. This macro should be called after `\ProcessOptionsKV[c]{Gm}`. If the landscape option in `\documentclass` is specified, the class immediately swaps the paper dimensions.

```
566 \def\Gm@setdefaultpaper{%
567   \ifx\Gm@paper\@undefined
568     \Gm@setsize{paper}(\strip@pt\paperwidth,\strip@pt\paperheight){pt}%
569     \Gm@setsize{Gm@layout}(\strip@pt\paperwidth,\strip@pt\paperheight){pt}%
570     \Gm@swap@papersizefalse
571   \fi}%
```

\Gm@adjustpaper The macro checks if paperwidth/height is larger than 0pt, which is used in `\Gm@process`. The paper dimensions can be swapped when paper orientation is changed over by landscape and portrait options.

```
572 \def\Gm@adjustpaper{%
573   \ifdim\paperwidth>\p@\else
574     \PackageError{geometry}{%
575     \string\paperwidth\space(\the\paperwidth) too short}{%
576     Set a paper type (e.g., 'a4paper').}%
577   \fi
578   \ifdim\paperheight>\p@\else
579     \PackageError{geometry}{%
580     \string\paperheight\space(\the\paperheight) too short}{%
581     Set a paper type (e.g., 'a4paper').}%
582   \fi
583   \ifGm@swap@papersize
584     \setlength\@tempdima{\paperwidth}%
585     \setlength\paperwidth{\paperheight}%
586     \setlength\paperheight{\@tempdima}%
587   \fi
588   \ifGm@layout\else
589     \let\Gm@layoutwidth\paperwidth
590     \let\Gm@layoutheight\paperheight
591   \fi}%
```

\Gm@checkmp The macro checks whether or not the marginpars overrun the page.

```
592 \def\Gm@checkmp{%
593   \ifGm@includemp\else
594     \@tempcnta\z@\@tempcntb\@ne
595     \if@twocolumn
596       \@tempcnta\@ne
597     \else
598       \if@reversemargin
599         \@tempcnta\@ne\@tempcntb\z@
600       \fi
601     \fi
602     \@tempdima\marginparwidth
603     \advance\@tempdima\marginparsep
604     \ifnum\@tempcnta=\@ne
605       \@tempdimc\@tempdima
606       \setlength\@tempdimb{\Gm@lmargin}%
607       \advance\@tempdimc-\@tempdimb
608       \ifdim\@tempdimc>\z@
609         \Gm@warning{The marginal notes overrun the paper edge.^^J
610         \@spaces Add \the\@tempdimc\space and more to the left margin}%
611       \fi
612     \fi
613     \ifnum\@tempcntb=\@ne
614       \@tempdimc\@tempdima
615       \setlength\@tempdimb{\Gm@rmargin}%
616       \advance\@tempdimc-\@tempdimb
617       \ifdim\@tempdimc>\z@
618         \Gm@warning{The marginal notes overrun the paper.^^J
619         \@spaces Add \the\@tempdimc\space and more to the right margin}%
620       \fi
621     \fi
```

```
622    \fi}%
```

\Gm@adjustmp The macro sets marginpar correction when `includemp` is set, which is used in `\Gm@process`. The variables `\Gm@wd@mp`, `\Gm@odd@mp` and `\Gm@even@mp` are set here. Note that `\Gm@even@mp` should be used only for twoside layout.

```
623 \def\Gm@adjustmp{%
624   \ifGm@includemp
625     \@tempdimb\marginparwidth
626     \advance\@tempdimb\marginparsep
627     \Gm@wd@mp\@tempdimb
628     \Gm@odd@mp\z@
629     \Gm@even@mp\z@
630     \if@twocolumn
631       \Gm@wd@mp2\@tempdimb
632       \Gm@odd@mp\@tempdimb
633       \Gm@even@mp\@tempdimb
634     \else
635       \if@reversemargin
636         \Gm@odd@mp\@tempdimb
637         \if@mparswitch\else
638           \Gm@even@mp\@tempdimb
639         \fi
640       \else
641         \if@mparswitch
642           \Gm@even@mp\@tempdimb
643         \fi
644       \fi
645     \fi
646   \fi}%
```

\Gm@adjustbody If the horizontal dimension of *body* is specified by user, `\Gm@width` is set properly here.

```
647 \def\Gm@adjustbody{
648   \ifGm@hbody
649     \ifx\Gm@width\@undefined
650       \ifx\Gm@hscale\@undefined
651         \edef\Gm@width{\Gm@Dhscale\Gm@layoutwidth}%
652       \else
653         \edef\Gm@width{\Gm@hscale\Gm@layoutwidth}%
654       \fi
655     \fi
656     \ifx\Gm@textwidth\@undefined\else
657       \setlength\@tempdima{\Gm@textwidth}%
658       \ifGm@includemp
659         \advance\@tempdima\Gm@wd@mp
660       \fi
661       \edef\Gm@width{\the\@tempdima}%
662     \fi
663   \fi
```

If the vertical dimension of *body* is specified by user, `\Gm@height` is set properly here.

```
664   \ifGm@vbody
665     \ifx\Gm@height\@undefined
666       \ifx\Gm@vscale\@undefined
667         \edef\Gm@height{\Gm@Dvscale\Gm@layoutheight}%
668       \else
669         \edef\Gm@height{\Gm@vscale\Gm@layoutheight}%
670       \fi
671     \fi
672     \ifx\Gm@lines\@undefined\else
```

`\topskip` has to be adjusted so that the formula "`\textheight` $= (lines - 1) \times$ `\baselineskip` $+$ `\topskip`" to be correct even if large font sizes are specified by users. If `\topskip` is smaller than `\ht\strutbox`, then `\topskip` is set to `\ht\strutbox`.

```
673       \ifdim\topskip<\ht\strutbox
674         \setlength\@tempdima{\topskip}%
675         \setlength\topskip{\ht\strutbox}%
```

```
676        \Gm@warning{\noexpand\topskip was changed from \the\@tempdima\space
677          to \the\topskip}%
678        \fi
679        \setlength\@tempdima{\baselineskip}%
680        \multiply\@tempdima\Gm@lines
681        \addtolength\@tempdima{\topskip}%
682        \addtolength\@tempdima{-\baselineskip}%
683        \edef\Gm@textheight{\the\@tempdima}%
684      \fi
685      \ifx\Gm@textheight\@undefined\else
686        \setlength\@tempdima{\Gm@textheight}%
687        \ifGm@includehead
688          \addtolength\@tempdima{\headheight}%
689          \addtolength\@tempdima{\headsep}%
690        \fi
691        \ifGm@includefoot
692          \addtolength\@tempdima{\footskip}%
693        \fi
694        \edef\Gm@height{\the\@tempdima}%
695      \fi
696    \fi}%
```

\Gm@process   The main macro processing the specified dimensions is defined.

```
697 \def\Gm@process{%
```

If `pass` is set, the original dimensions and switches are restored and process is ended here.

```
698    \ifGm@pass
699      \Gm@restore@org
700    \else
701      \Gm@@process
702    \fi}%
```

The main processing macro.

```
703 \def\Gm@@process{%
704    \Gm@expandlengths
705    \Gm@adjustpaper
706    \addtolength\Gm@layoutwidth{-\Gm@bindingoffset}%
707    \Gm@adjustmp
708    \Gm@adjustbody
709    \Gm@detall{h}{width}{lmargin}{rmargin}%
710    \Gm@detall{v}{height}{tmargin}{bmargin}%
```

The real dimensions are set properly according to the result of the auto-completion calculation.

```
711    \setlength\textwidth{\Gm@width}%
712    \setlength\textheight{\Gm@height}%
713    \setlength\topmargin{\Gm@tmargin}%
714    \setlength\oddsidemargin{\Gm@lmargin}%
715    \addtolength\oddsidemargin{-1\Gm@truedimen in}%
```

If `includemp` is set to `true`, \textwidth and \oddsidemargin are adjusted.

```
716    \ifGm@includemp
717      \advance\textwidth-\Gm@wd@mp
718      \advance\oddsidemargin\Gm@odd@mp
719    \fi
```

Determining \evensidemargin. In the twoside page layout, the right margin value \Gm@rmargin is used. If the marginal note width is included, \evensidemargin should be corrected by \Gm@even@mp.

```
720    \if@mparswitch
721      \setlength\evensidemargin{\Gm@rmargin}%
722      \addtolength\evensidemargin{-1\Gm@truedimen in}%
723      \ifGm@includemp
724        \advance\evensidemargin\Gm@even@mp
725      \fi
726    \else
727      \evensidemargin\oddsidemargin
728    \fi
```

The bindingoffset correction for \oddsidemargin.

```
729    \advance\oddsidemargin\Gm@bindingoffset
730    \addtolength\topmargin{-1\Gm@truedimen in}%
```

If the head of the page is included in *total body*, \headheight and \headsep are removed from \textheight, otherwise from \topmargin.

```
731    \ifGm@includehead
732      \addtolength\textheight{-\headheight}%
733      \addtolength\textheight{-\headsep}%
734    \else
735      \addtolength\topmargin{-\headheight}%
736      \addtolength\topmargin{-\headsep}%
737    \fi
```

If the foot of the page is included in *total body*, \footskip is removed from \textheight.

```
738    \ifGm@includefoot
739      \addtolength\textheight{-\footskip}%
740    \fi
```

If heightrounded is set, \textheight is rounded.

```
741    \ifGm@heightrounded
742      \setlength\@tempdima{\textheight}%
743      \addtolength\@tempdima{-\topskip}%
744      \@tempcnta\@tempdima
745      \@tempcntb\baselineskip
746      \divide\@tempcnta\@tempcntb
747      \setlength\@tempdimb{\baselineskip}%
748      \multiply\@tempdimb\@tempcnta
749      \advance\@tempdima-\@tempdimb
750      \multiply\@tempdima\tw@
751      \ifdim\@tempdima>\baselineskip
752        \addtolength\@tempdimb{\baselineskip}%
753      \fi
754      \addtolength\@tempdimb{\topskip}%
755      \textheight\@tempdimb
756    \fi
```

The paper width is set back by adding \Gm@bindingoffset.

```
757    \advance\oddsidemargin\Gm@layouthoffset%
758    \advance\evensidemargin\Gm@layouthoffset%
759    \advance\topmargin\Gm@layoutvoffset%
760    \addtolength\Gm@layoutwidth{\Gm@bindingoffset}%
761    }% end of \Gm@@process
```

\Gm@detectdriver   The macro checks the typeset environment and changes the driver option if necessary. To make the engine detection more robust, the macro is rewritten in the version 4 with packages ifpdf and ifvtex.

```
762 \def\Gm@detectdriver{%
```

If the driver option is not specified explicitly, then driver auto-detection works.

```
763    \ifx\Gm@driver\@empty
764      \typeout{*geometry* driver: auto-detecting}%
```

\ifpdf is defined in ifpdf package in 'oberdiek' bundle.

```
765      \ifpdf
766        \Gm@setdriver{pdftex}%
767      \else
768        \Gm@setdriver{dvips}%
769      \fi
```

\ifvtex is defined in ifvtex package in 'oberdiek' bundle.

```
770      \ifvtex
771        \Gm@setdriver{vtex}%
772      \fi
```

xetex

```
773      \@ifundefined{XeTeXversion}{}{\Gm@setdriver{xetex}}%
```

When the driver option is set by the user, check if it is valid or not.

```
774 \else
775   \ifx\Gm@driver\Gm@xetex %%
776     \@ifundefined{XeTeXversion}{\Gm@warning{%
777         Wrong driver setting: 'xetex'; trying 'pdftex' driver}%
778         \Gm@setdriver{pdftex}}{}%
779   \fi
780   \ifx\Gm@driver\Gm@vtex
781     \ifvtex\else
782       \Gm@warning{Wrong driver setting: 'vtex'; trying 'dvips' driver}%
783       \Gm@setdriver{dvips}%
784     \fi
785   \fi
786 \fi
787 \ifx\Gm@driver\relax
788   \typeout{*geometry* detected driver: <none>}%
789 \else
790   \typeout{*geometry* detected driver: \Gm@driver}%
791 \fi}%
```

\Gm@showparams Prints the resulted parammeters and dimensions to STDOUT if verbose is true. \Gm@width and
\Gm@height are expanded to get the real size.

```
792 \def\Gm@showparams#1{%
793   \ifGm@pass\else
794     \setlength\@tempdima{\Gm@width}\edef\Gm@width{\the\@tempdima}%
795     \setlength\@tempdima{\Gm@height}\edef\Gm@height{\the\@tempdima}%
796   \fi
797   \ifGm@verbose\expandafter\typeout\else\expandafter\wlog\fi
798   {\Gm@logcontent{#1}}}%
799 \def\Gm@showdim#1{* \string#1=\the#1^^J}%
800 \def\Gm@showbool#1{\@nameuse{ifGm@#1}#1\space\fi}%
```

\Gm@logcontent The content of geometry paramenters and native dimensions for the page layout.

```
801 \def\Gm@logcontent#1{%
802   *geometry* verbose mode - [ #1 ] result:^^J%
803   \ifGm@pass * pass (ignores the geometry layout)^^J%
804   \else
805   * driver: \if\Gm@driver\relax <none>\else\Gm@driver\fi^^J%
806   * paper: \ifx\Gm@paper\@undefined\the\paperwidth,
807     \the\paperheight\space(class default)\else\Gm@paper\fi^^J%
808   * layout: \ifGm@layout \the\Gm@layoutwidth, \the\Gm@layoutheight
809   \else\ifx\Gm@paper\@undefined\the\paperwidth,
810     \the\paperheight\else\Gm@paper\fi\fi^^J%
811   \@ifundefined{Gm@lines}{}{* lines: \Gm@lines^^J}%
812   \@ifundefined{Gm@hmarginratio}{}{* hratio: \Gm@hmarginratio^^J}%
813   \@ifundefined{Gm@vmarginratio}{}{* vratio: \Gm@vmarginratio^^J}%
814   \ifdim\Gm@bindingoffset=\z@\else
815   * bindingoffset: \the\Gm@bindingoffset^^J\fi
816   * switch(es): %
817   \Gm@showbool{landscape}%
818   \Gm@showbool{includehead}%
819   \Gm@showbool{includefoot}%
820   \Gm@showbool{includemp}%
821   \if@twoside twoside\space\fi%
822   \if@mparswitch\else\if@twoside asymmetric\space\fi\fi%
823   \Gm@showbool{heightrounded}%
824   \ifx\Gm@truedimen\@empty\else truedimen\space\fi%
825   ^^J%
826   * h-part:(L,W,R)=(\Gm@lmargin, \Gm@width, \Gm@rmargin)^^J%
827   * v-part:(T,H,B)=(\Gm@tmargin, \Gm@height, \Gm@bmargin)^^J%
828   \fi
829   \Gm@showdim{\paperwidth}%
830   \Gm@showdim{\paperheight}%
831   \Gm@showdim{\textwidth}%
832   \Gm@showdim{\textheight}%
```

```
833    \Gm@showdim{\oddsidemargin}%
834    \Gm@showdim{\evensidemargin}%
835    \Gm@showdim{\topmargin}%
836    \Gm@showdim{\headheight}%
837    \Gm@showdim{\headsep}%
838    \Gm@showdim{\footskip}%
839    \Gm@showdim{\marginparwidth}%
840    \Gm@showdim{\marginparsep}%
841    \Gm@showdim{\columnsep}%
842    * \string\skip\string\footins=\the\skip\footins^^J%
843    \Gm@showdim{\hoffset}%
844    \Gm@showdim{\voffset}%
845    \Gm@showdim{\mag}%
846    * \string\@twocolumn\if@twocolumn true\else false\fi^^J%
847    * \string\@twoside\if@twoside true\else false\fi^^J%
848    * \string\@mparswitch\if@mparswitch true\else false\fi^^J%
849    * \string\@reversemargin\if@reversemargin true\else false\fi^^J%
850    * (1in=72.27pt, 1cm=28.453pt)^^J}%
```

Macros for the page frames and cropmarks.

```
851 \def\Gm@cropmark(#1,#2,#3,#4){%
852   \begin{picture}(0,0)
853     \setlength\unitlength{1truemm}\thinlines
854     \put(0,0){\line(#1,#2){5}}
855     \put(0,0){\line(#3,#4){5}}
856   \end{picture}}%
857 \newcommand*{\Gm@vrules@mpi}{%
858   \hb@xt@\@tempdima{\llap{\vrule height\textheight}\ignorespaces
859   \hskip \textwidth\vrule height\textheight\hskip \marginparsep
860   \llap{\vrule height\textheight}\hfil\vrule height\textheight}}%
861 \newcommand*{\Gm@vrules@mpii}{%
862   \hb@xt@\@tempdima{\hskip-\marginparwidth\hskip-\marginparsep
863   \llap{\vrule height\textheight}\ignorespaces
864   \hskip \marginparwidth\rlap{\vrule height\textheight}\hskip \marginparsep
865   \llap{\vrule height\textheight}\hskip\textwidth\rlap{\vrule height\textheight}\hss}}%
866 \providecommand*\vb@xt@{\vbox to}%
867 \newcommand*{\Gm@pageframes}{%
868   \vb@xt@\z@{\baselineskip\z@skip\lineskip\z@skip\lineskiplimit\z@%
869     \ifGm@showcrop
870     \vb@xt@\z@{\vskip-1\Gm@truedimen in\vskip\Gm@layoutvoffset%
871       \hb@xt@\z@{\hskip-1\Gm@truedimen in\hskip\Gm@layouthoffset%
872       \vb@xt@\Gm@layoutheight{%
873         \let\protect\relax
874         \hb@xt@\Gm@layoutwidth{\Gm@cropmark(1,0,0,-1)\hfil\Gm@cropmark(-1,0,0,-1)}%
875         \vfil
876         \hb@xt@\Gm@layoutwidth{\Gm@cropmark(1,0,0,1)\hfil\Gm@cropmark(-1,0,0,1)}}%
877       \hss}%
878     \vss}%
879     \fi%
880     \ifGm@showframe
881     \if@twoside
882     \ifodd\count\z@
883       \let\@themargin\oddsidemargin
884     \else
885       \let\@themargin\evensidemargin
886     \fi
887     \fi
888     \moveright\@themargin%
889     \vb@xt@\z@{%
890     \vskip\topmargin\vb@xt@\z@{\vss\hrule width\textwidth}%
891     \vskip\headheight\vb@xt@\z@{\vss\hrule width\textwidth}%
892     \vskip\headsep\vb@xt@\z@{\vss\hrule width\textwidth}
893     \@tempdima\textwidth
894     \advance\@tempdima by \marginparsep
895     \advance\@tempdima by \marginparwidth
```

```
896      \if@mparswitch
897       \ifodd\count\z@
898        \Gm@vrules@mpi
899       \else
900        \Gm@vrules@mpii
901       \fi
902      \else
903       \Gm@vrules@mpi
904      \fi
905      \vb@xt@\z@{\vss\hrule width\textwidth}%
906      \vskip\footskip\vb@xt@\z@{\vss\hrule width\textwidth}%
907      \vss}%
908     \fi%
909    }}%
910 \let\Gm@shipout\shipout
911 \newcommand*{\gm@shipi}{%
912    \ifvoid\@cclv\expandafter\aftergroup\fi\gm@shipii}%
913 \newcommand*\gm@shipii{%
914    \ifvoid\@cclv
915      \Gm@shipout\box\@cclv
916    \else
917      \ifGm@showframe
918        \Gm@shipout\vbox{\Gm@pageframes\ifvbox\@cclv\unvbox\else\box\fi\@cclv}%
919      \else\ifGm@showcrop
920        \Gm@shipout\vbox{\Gm@pageframes\ifvbox\@cclv\unvbox\else\box\fi\@cclv}%
921      \else
922        \Gm@shipout\box\@cclv
923      \fi\fi
924    \fi}
```

\ProcessOptionsKV This macro can process class and package options using 'key=value' scheme. Only class options are processed with an optional argument 'c', package options with 'p', and both of them by default.

```
925 \def\ProcessOptionsKV{\@ifnextchar[%
926    {\@ProcessOptionsKV}{\@ProcessOptionsKV[]}}%
927 \def\@ProcessOptionsKV[#1]#2{%
928    \let\@tempa\@empty
929    \@tempcnta\z@
930    \if#1p\@tempcnta\@ne\else\if#1c\@tempcnta\tw@\fi\fi
931    \ifodd\@tempcnta
932     \edef\@tempa{\@ptionlist{\@currname.\@currext}}%
933    \else
934     \@for\CurrentOption:=\@classoptionslist\do{%
935        \@ifundefined{KV@#2@\CurrentOption}%
936        {}{\edef\@tempa{\@tempa,\CurrentOption,}}}%
937     \ifnum\@tempcnta=\z@
938        \edef\@tempa{\@tempa,\@ptionlist{\@currname.\@currext}}%
939     \fi
940    \fi
941    \edef\@tempa{\noexpand\setkeys{#2}{\@tempa}}%
942    \@tempa
943    \AtEndOfPackage{\let\@unprocessedoptions\relax}}%

944 \def\Gm@setkeys{\setkeys{Gm}}%
```

\Gm@processconf **\ExecuteOptions** is replaced with **\Gm@setkey** to make it possible to deal with '⟨key⟩=⟨value⟩' as its argument.

```
945 \def\Gm@processconfig{%
946    \let\Gm@origExecuteOptions\ExecuteOptions
947    \let\ExecuteOptions\Gm@setkeys
948    \InputIfFileExists{geometry.cfg}{}{}
949    \let\ExecuteOptions\Gm@origExecuteOptions}%
```

The original page layout before loading geometry is saved here. **\Gm@restore@org** is defined here for reset option.

```
950 \Gm@save
```

951 `\edef\Gm@restore@org{\Gm@restore}%`
952 `\Gm@initall`

Processing config file.

953 `\Gm@processconfig`

The optional arguments to `\documentclass` are processed here.

954 `\ProcessOptionsKV[c]{Gm}%`

Paper dimensions given by class default are stored.

955 `\Gm@setdefaultpaper`

The optional arguments to `\usepackage` are processed here.

956 `\ProcessOptionsKV[p]{Gm}%`

Actual settings and calculation for layout dimensions are processed.

957 `\Gm@process`

`\AtBeginDocument` The processes for `verbose`, `showframe` and drivers are added to `\AtBeginDocument`. `\Gm@restore@org` is redefined here with the paper size specified in the preamble for `\newgeometry` to use it. This should be done before magnifying the paper size with `\mag` because the layout calculation would be affected by changing the paper size.

958 `\AtBeginDocument{%`
959 `  \Gm@savelength{paperwidth}%`
960 `  \Gm@savelength{paperheight}%`
961 `  \edef\Gm@restore@org{\Gm@restore}%`

The original paper size is used if `resetpaper`.

962 `  \ifGm@resetpaper`
963 `    \edef\Gm@pw{\Gm@orgpw}%`
964 `    \edef\Gm@ph{\Gm@orgph}%`
965 `  \else`
966 `    \edef\Gm@pw{\the\paperwidth}%`
967 `    \edef\Gm@ph{\the\paperheight}%`
968 `  \fi`

If `pass` is not set, the paper size is multiplied according to the specified `mag`.

969 `  \ifGm@pass\else`
970 `    \ifnum\mag=\@m\else`
971 `      \Gm@magtooffset`
972 `      \divide\paperwidth\@m`
973 `      \multiply\paperwidth\the\mag`
974 `      \divide\paperheight\@m`
975 `      \multiply\paperheight\the\mag`
976 `    \fi`
977 `  \fi`

Checking the driver options.

978 `  \Gm@detectdriver`

If `xetex` and `\pdfpagewidth` is defined, `\pdfpagewidth` and `\pdfpageheight` would be set.

979 `  \ifx\Gm@driver\Gm@xetex`
980 `    \@ifundefined{pdfpagewidth}{}{%`
981 `      \setlength\pdfpagewidth{\Gm@pw}%`
982 `      \setlength\pdfpageheight{\Gm@ph}}%`
983 `    \ifnum\mag=\@m\else`
984 `      \ifx\Gm@truedimen\Gm@true`
985 `        \setlength\paperwidth{\Gm@pw}%`
986 `        \setlength\paperheight{\Gm@ph}%`
987 `      \fi`
988 `    \fi`
989 `  \fi`

If `pdftex` is set to `true`, pdf-commands are set properly. To avoid `pdftex` magnification problem, `\pdfhorigin` and `\pdfvorigin` are adjusted for `\mag`.

990 `  \ifx\Gm@driver\Gm@pdftex`
991 `    \@ifundefined{pdfpagewidth}{}{%`
992 `      \setlength\pdfpagewidth{\Gm@pw}%`
993 `      \setlength\pdfpageheight{\Gm@ph}}%`

```
994      \ifnum\mag=\@m\else
995        \@tempdima=\mag sp%
996        \@ifundefined{pdfhorigin}{}{%
997          \divide\pdfhorigin\@tempdima
998          \multiply\pdfhorigin\@m
999          \divide\pdfvorigin\@tempdima
1000         \multiply\pdfvorigin\@m}%
1001       \ifx\Gm@truedimen\Gm@true
1002         \setlength\paperwidth{\Gm@pw}%
1003         \setlength\paperheight{\Gm@ph}%
1004       \fi
1005     \fi
1006   \fi
```

With VTEX environment, VTEX variables are set here.

```
1007   \ifx\Gm@driver\Gm@vtex
1008     \@ifundefined{mediawidth}{}{%
1009       \mediawidth=\paperwidth
1010       \mediaheight=\paperheight}%
1011     \ifvtexdvi
1012       \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1013     \fi
1014   \fi
```

If dvips or dvipdfm is specified, paper size is embedded in dvi file with \special. For dvips, a landscape correction is added because a landscape document converted by dvips is upside-down in PostScript viewers.

```
1015   \ifx\Gm@driver\Gm@dvips
1016     \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1017     \ifx\Gm@driver\Gm@dvips\ifGm@landscape
1018       \AtBeginDvi{\special{! /landplus90 true store}}%
1019     \fi\fi
```

If dvipdfm is specified and atbegshi package in 'oberdiek' bundle is loaded, \AtBeginShipoutFirst is used instead of \AtBeginDvi for compatibility with hyperref and dvipdfm program.

```
1020   \else\ifx\Gm@driver\Gm@dvipdfm
1021     \ifcase\ifx\AtBeginShipoutFirst\relax\@ne\else
1022         \ifx\AtBeginShipoutFirst\@undefined\@ne\else\z@\fi\fi
1023       \AtBeginShipoutFirst{\special{papersize=\the\paperwidth,\the\paperheight}}%
1024     \or
1025       \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1026     \fi
1027   \fi\fi
```

If showframe=true, page frames and lines are showed on the first page.

```
1028   \ifGm@showframe
1029     \renewcommand*{\shipout}{\afterassignment\gm@shipi\setbox\@cclv=}
1030   \else\ifGm@showcrop
1031     \renewcommand*{\shipout}{\afterassignment\gm@shipi\setbox\@cclv=}
1032   \fi\fi
```

The layout dimensions for \restoregeometry are saved at the end of the \AtBeginDocument.

```
1033   \Gm@save
1034   \edef\Gm@restore@pkg{\Gm@restore}%
```

The package checks whether or not the marginpars overrun the page, if verbose and unless pass.

```
1035   \ifGm@verbose\ifGm@pass\else\Gm@checkmp\fi\fi
```

\Gm@showparams puts the resulting parameters and dimensions into the log file. With verbose, they are shown on the terminal as well.

```
1036   \Gm@showparams{preamble}%
```

The following lines free the memories no longer needed.

```
1037   \let\Gm@pw\relax
1038   \let\Gm@ph\relax
1039   }% end of \AtBeginDocument
```

\geometry   The macro `\geometry` can be called multiple times in the preamble (before `\begin{document}`).

```
1040 \newcommand{\geometry}[1]{%
1041   \Gm@clean
1042   \setkeys{Gm}{#1}%
1043   \Gm@process}%
1044 \@onlypreamble\geometry
```

\Gm@changelayout   The macro, which can be called from `\newgeometry`, `\restoregeometry` and `\loadgeometry`, changes the layout in the middle of the document.

```
1045 \DeclareRobustCommand\Gm@changelayout{%
1046   \setlength{\@colht}{\textheight}
1047   \setlength{\@colroom}{\textheight}%
1048   \setlength{\vsize}{\textheight}
1049   \setlength{\columnwidth}{\textwidth}%
1050   \if@twocolumn%
1051     \advance\columnwidth-\columnsep
1052     \divide\columnwidth\tw@%
1053     \@firstcolumntrue%
1054   \fi%
1055   \setlength{\hsize}{\columnwidth}%
1056   \setlength{\linewidth}{\hsize}}%
```

\newgeometry   The macro `\newgeometry`, which changes the layout, can be used only in the document. It would reset the options specified in the preamble except for paper size options and `\mag`.

```
1057 \newcommand\newgeometry[1]{%
1058   \clearpage
1059   \Gm@restore@org
1060   \Gm@initnewgm
1061   \Gm@newgmtrue
1062   \setkeys{Gm}{#1}%
1063   \Gm@newgmfalse
1064   \Gm@process
1065   \ifnum\mag=\@m\else\Gm@magtooffset\fi
1066   \Gm@changelayout
1067   \Gm@showparams{newgeometry}}%
```

\restoregeometry   The macro restores the resulting layout specified in the preamble, namely the first-page layout right after `\begin{document}`.

```
1068 \newcommand\restoregeometry{%
1069   \clearpage
1070   \Gm@restore@pkg
1071   \Gm@changelayout}%
```

\savegeometry   The macro saves the layout with the name specified with the argument. The saved layout can be loaded with `\loadgeometry{⟨name⟩}`.

```
1072 \newcommand*\savegeometry[1]{%
1073   \Gm@save
1074   \expandafter\edef\csname Gm@restore@@#1\endcsname{\Gm@restore}}%
```

\loadgeometry   The macro loads the layout saved with `\savegeometry{⟨name⟩}`. If the name is not found, the macro would warn it and do nothing for the layout.

```
1075 \newcommand*\loadgeometry[1]{%
1076   \clearpage
1077   \@ifundefined{Gm@restore@@#1}{%
1078     \PackageError{geometry}{%
1079     \string\loadgeometry : name '#1' undefined}{%
1080     The name '#1' should be predefined with \string\savegeometry}%
1081   }{\@nameuse{Gm@restore@@#1}%
1082   \Gm@changelayout}}%
1083 ⟨/package⟩
```

12 Config file

In the configuration file `geometry.cfg`, one can use `\ExecuteOptions` to set the site or user default settings.

```
1084 ⟨∗config⟩
1085 %<<SAVE_INTACT
1086
1087 %  Uncomment and edit the line below to set default options.
1088 %\ExecuteOptions{a4paper}
1089
1090 %SAVE_INTACT
1091 ⟨/config⟩
```

13 Sample file

Here is a sample document for the `geometry` package.

```
1092 ⟨∗samples⟩
1093 %<<SAVE_INTACT
1094 \documentclass[12pt]{article}% uses letterpaper by default
1095 % \documentclass[12pt,a4paper]{article}% for A4 paper
1096 %----------------------------------------------------------
1097 % Edit and uncomment one of the settings below
1098 %----------------------------------------------------------
1099 % \usepackage{geometry}
1100 % \usepackage[centering]{geometry}
1101 % \usepackage[width=10cm,vscale=.7]{geometry}
1102 % \usepackage[margin=1cm, papersize={12cm,19cm}, resetpaper]{geometry}
1103 % \usepackage[margin=1cm,includeheadfoot]{geometry}
1104 \usepackage[margin=1cm,includeheadfoot,includemp]{geometry}
1105 % \usepackage[margin=1cm,bindingoffset=1cm,twoside]{geometry}
1106 % \usepackage[hmarginratio=2:1, vmargin=2cm]{geometry}
1107 % \usepackage[hscale=0.5,twoside]{geometry}
1108 % \usepackage[hscale=0.5,asymmetric]{geometry}
1109 % \usepackage[hscale=0.5,heightrounded]{geometry}
1110 % \usepackage[left=1cm,right=4cm,top=2cm,includefoot]{geometry}
1111 % \usepackage[lines=20,left=2cm,right=6cm,top=2cm,twoside]{geometry}
1112 % \usepackage[width=15cm, marginparwidth=3cm, includemp]{geometry}
1113 % \usepackage[hdivide={1cm,,2cm}, vdivide={3cm,8in,}, nohead]{geometry}
1114 % \usepackage[headsep=20pt, head=40pt,foot=20pt,includeheadfoot]{geometry}
1115 % \usepackage[text={6in,8in}, top=2cm, left=2cm]{geometry}
1116 % \usepackage[centering,includemp,twoside,landscape]{geometry}
1117 % \usepackage[mag=1414,margin=2cm]{geometry}
1118 % \usepackage[mag=1414,margin=2truecm,truedimen]{geometry}
1119 % \usepackage[a5paper, landscape, twocolumn, twoside,
1120 %    left=2cm, hmarginratio=2:1, includemp, marginparwidth=43pt,
1121 %    bottom=1cm, foot=.7cm, includefoot, textheight=11cm, heightrounded,
1122 %    columnsep=1cm,verbose]{geometry}
1123 %----------------------------------------------------------
1124 % No need to change below
1125 %----------------------------------------------------------
1126 \geometry{verbose,showframe}% the options appended.
1127 \usepackage{lipsum}% for dummy text of 150 paragraphs
1128 \newcommand\mynote{\marginpar[\raggedright
1129 A sample margin note in the left side.]%
1130 {\raggedright A sample margin note.}}%
1131 \newcommand\myfootnote{\footnote{This is a sample footnote text.}}
1132 \begin{document}
1133 \lipsum[1-2]\mynote\lipsum[3-4]\mynote
1134 \lipsum[5-11]\mynote\lipsum[12]\myfootnote
1135 \lipsum[13-22]\mynote\lipsum[23-32]
1136 \end{document}
1137 %SAVE_INTACT
1138 ⟨/samples⟩
```