

# The geometry package

Hideo Umeki  
latexgeometry@gmail.com

2010/02/12 v5.0

## 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}`. With `\newgeometry` command you can change the layout anywhere in the document.

## 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 L<sup>A</sup>T<sub>E</sub>X is not straightforward. You need to adjust several L<sup>A</sup>T<sub>E</sub>X 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 mechanism, 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*.

<i>paper</i>	:	<i>total body</i> and <i>margins</i>
<i>total body</i>	:	<i>body</i> (text area) (optional <i>head</i> , <i>foot</i> and <i>marginpar</i> )
<i>margins</i>	:	<i>left(inner)</i> , <i>right(outer)</i> , <i>top</i> and <i>bottom</i>

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.

$$\text{paperwidth} = \text{left} + \text{width} + \text{right} \quad (1)$$

$$\text{paperheight} = \text{top} + \text{height} + \text{bottom} \quad (2)$$

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

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

$$\text{height} := \text{textheight} \quad (+\text{headheight} + \text{headsep} + \text{footskip}) \quad (4)$$

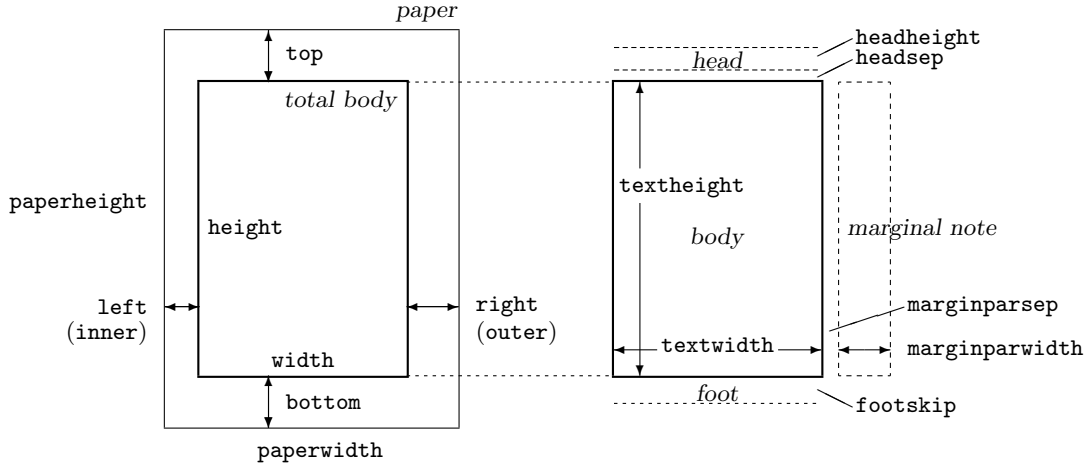


Figure 1: Dimension names used in the `geometry` package. `width=textwidth` and `height=textheight` by default. `left`, `right`, `top` and `bottom` are margins. If margins on verso pages are swapped by `twoside` option, margins specified by `left` and `right` options are used for the inside and outside margins respectively. `inner` and `outer` are aliases of `left` and `right` respectively.

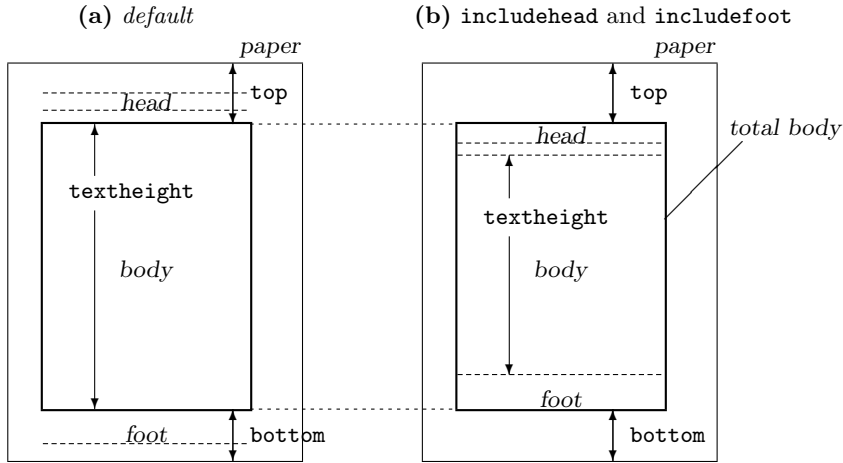


Figure 2: `includehead` and `includefoot` include the head and foot respectively into `total body`. (a) `height = textheight` (default). (b) `height = textheight + headheight + headsep + footskip` if `includehead` and `includefoot`. If the top and bottom margins are specified, `includehead` and `includefoot`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

$$L = a + b + c \quad (5)$$

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}
```

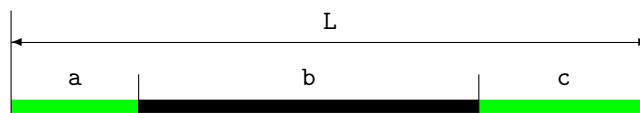


Figure 3: A simple model of page dimensions.

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*<sup>1</sup>. For example,

```
\usepackage{calc}
\usepackage[tehtheight=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.

$\langle key \rangle = \text{true} \mid \text{false}.$   
 $\langle key \rangle$  with no value is equivalent to  $\langle key \rangle = \text{true}.$

*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 <sup>†</sup> are not available as a argument of `\newgeometry` (See Section 7).

---

<sup>1</sup>CTAN: macros/latex/required/tools

## 5.1 Paper size

The options below set paper/media size and orientation.

<sup>†</sup> <code>paper</code>   <code>papername</code>	specifies the paper size by name. <code>paper=&lt;paper-name&gt;</code> . For convenience, you can specify the paper name without <code>paper=</code> . For example, <code>a4paper</code> is equivalent to <code>paper=a4paper</code> .
<sup>†</sup> <code>a0paper</code> , <code>a1paper</code> , <code>a2paper</code> , <code>a3paper</code> , <code>a4paper</code> , <code>a5paper</code> , <code>a6paper</code> , <code>b0paper</code> , <code>b1paper</code> , <code>b2paper</code> , <code>b3paper</code> , <code>b4paper</code> , <code>b5paper</code> , <code>b6paper</code> , <code>b0j</code> , <code>b1j</code> , <code>b2j</code> , <code>b3j</code> , <code>b4j</code> , <code>b5j</code> , <code>b6j</code> , <code>ansipaper</code> , <code>ansipaper</code> , <code>ansicpaper</code> , <code>ansidpaper</code> , <code>ansiepaper</code> , <code>letterpaper</code> , <code>executivepaper</code> , <code>legalpaper</code>	specifies paper name. The value part is ignored even if any. For example, the followings have the same effect: <code>a5paper</code> , <code>a5paper=true</code> , <code>a5paper=false</code> and so forth. <code>a[0-6]paper</code> and <code>b[0-6]paper</code> 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. <code>b[0-6]j</code> should be used for the JIS B-series.
<sup>†</sup> <code>screen</code>	a special paper size with (W,H) = (225mm,180mm). For presentation with PC and video projector, “ <code>screen,centering</code> ” with ‘slide’ documentclass would be useful.
<sup>†</sup> <code>paperwidth</code>	width of the paper. <code>paperwidth=&lt;length&gt;</code> .
<sup>†</sup> <code>paperheight</code>	height of the paper. <code>paperheight=&lt;length&gt;</code> .
<sup>†</sup> <code>papersize</code>	width and height of the paper. <code>papersize={&lt;width&gt;,&lt;height&gt;}</code> or <code>papersize=&lt;length&gt;</code> .
<sup>†</sup> <code>landscape</code>	switches the paper orientation to landscape mode.
<sup>†</sup> <code>portrait</code>	switches the paper orientation to portrait mode. This is equivalent to <code>landscape=false</code> .

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.

<code>layout</code>	specifies the layout size by paper name. <code>layout=&lt;paper-name&gt;</code> . All the paper names defined in <code>geometry</code> are available. See Section 5.1 for details.
<code>layoutwidth</code>	width of the layout. <code>layoutwidth=&lt;length&gt;</code> .
<code>layoutheight</code>	height of the layout. <code>layoutheight=&lt;length&gt;</code> .
<code>layoutsize</code>	width and height of the layout. <code>layoutsize={&lt;width&gt;,&lt;height&gt;}</code> or <code>layoutsize=&lt;length&gt;</code> .
<code>layoutoffset</code>	specifies the horizontal offset from the left edge of the paper. <code>layoutoffset=&lt;length&gt;</code> .
<code>layoutvoffset</code>	specifies the vertical offset from the top edge of the paper. <code>layoutvoffset=&lt;length&gt;</code> .
<code>layoutoffset</code>	specifies both horizontal and vertical offsets. <code>layoutoffset={&lt;hoffset&gt;,&lt;voffset&gt;}</code> or <code>layoutsize=&lt;length&gt;</code> .

## 5.3 Body size

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

<code>hscale</code>	ratio of width of <i>total body</i> to <code>\paperwidth</code> . <code>hscale=&lt;h-scale&gt;</code> , e.g., <code>hscale=0.8</code> is equivalent to <code>width=0.8\paperwidth</code> . (0.7 by default)
<code>vscale</code>	ratio of height of <i>total body</i> to <code>\paperheight</code> , e.g., <code>vscale=&lt;v-scale&gt;</code> . (0.7 by default) <code>vscale=0.9</code> is equivalent to <code>height=0.9\paperheight</code> .

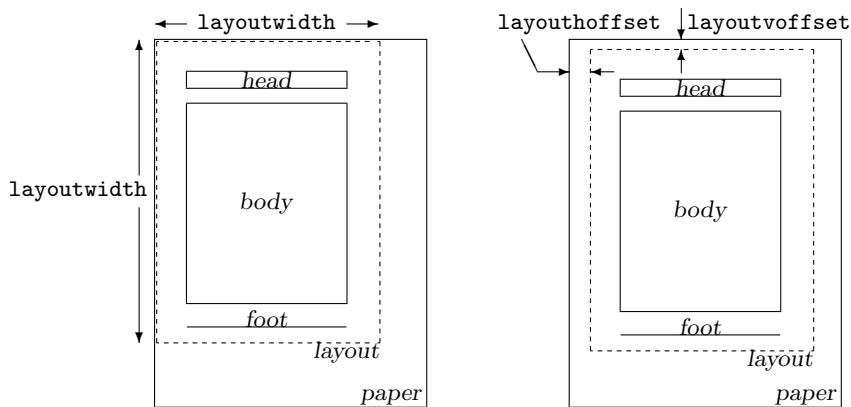


Figure 4: The dimensions related to the layout size. Note that the layout size defaults to the same size as the paper, so you don't have to specify layout-related options explicitly in most cases.

<code>scale</code>	ratio of <i>total body</i> to the paper. <code>scale={⟨<i>h-scale</i>⟩,⟨<i>v-scale</i>⟩}</code> or <code>scale=⟨<i>scale</i>⟩</code> . (0.7 by default)
<code>width</code>   <code>totalwidth</code>	width of <i>total body</i> . <code>width=⟨<i>length</i>⟩</code> or <code>totalwidth=⟨<i>length</i>⟩</code> . This dimension defaults to <code>textwidth</code> , but if <code>includemp</code> is set to <code>true</code> , <code>width</code> $\geq$ <code>textwidth</code> because <code>width</code> includes the width of the marginal notes. If <code>textwidth</code> and <code>width</code> are specified at the same time, <code>textwidth</code> takes priority over <code>width</code> .
<code>height</code>   <code>totalheight</code>	height of <i>total body</i> , excluding header and footer by default. If <code>includehead</code> or <code>includefoot</code> is set, <code>height</code> includes the head or foot of the page as well as <code>textheight</code> . <code>height=⟨<i>length</i>⟩</code> or <code>totalheight=⟨<i>length</i>⟩</code> . If both <code>textheight</code> and <code>height</code> are specified, <code>height</code> will be ignored.
<code>total</code>	width and height of <i>total body</i> . <code>total={⟨<i>width</i>⟩,⟨<i>height</i>⟩}</code> or <code>total=⟨<i>length</i>⟩</code> .
<code>textwidth</code>	specifies <code>\textwidth</code> , the width of <i>body</i> (the text area). <code>textwidth=⟨<i>length</i>⟩</code> .
<code>textheight</code>	specifies <code>\textheight</code> , the height of <i>body</i> (the text area). <code>textheight=⟨<i>length</i>⟩</code> .
<code>text</code>   <code>body</code>	specifies both <code>\textwidth</code> and <code>\textheight</code> of the body of page. <code>body={⟨<i>width</i>⟩,⟨<i>height</i>⟩}</code> or <code>text=⟨<i>length</i>⟩</code> .
<code>lines</code>	enables users to specify <code>\textheight</code> by the number of lines. <code>lines=⟨<i>integer</i>⟩</code> .
<code>includehead</code>	includes the head of the page, <code>\headheight</code> and <code>\headsep</code> , into <i>total body</i> . It is set to <code>false</code> by default. It is opposite to <code>ignorehead</code> . See Figure 2 and Figure 5.
<code>includefoot</code>	includes the foot of the page, <code>\footskip</code> , into <i>total body</i> . It is opposite to <code>ignorefoot</code> . It is <code>false</code> by default. See Figure 2 and Figure 5.
<code>includeheadfoot</code>	sets both <code>includehead</code> and <code>includefoot</code> to <code>true</code> , which is opposite to <code>ignoreheadfoot</code> . See Figure 2 and Figure 5.
<code>includemp</code>	includes the margin notes, <code>\marginparwidth</code> and <code>\marginparsep</code> , into <i>body</i> when calculating horizontal calculation.
<code>includeall</code>	sets both <code>includeheadfoot</code> and <code>includemp</code> to <code>true</code> . See Figure 5.
<code>ignorehead</code>	disregards the head of the page, <code>headheight</code> and <code>headsep</code> , in determining vertical layout, but does not change those lengths. It is equivalent to <code>includehead=false</code> . It is set to <code>true</code> by default. See also <code>includehead</code> .
<code>ignorefoot</code>	disregards the foot of page, <code>footskip</code> , in determining vertical layout, but does not change that length. This option defaults to <code>true</code> . See also <code>includefoot</code> .
<code>ignoreheadfoot</code>	sets both <code>ignorehead</code> and <code>ignorefoot</code> to <code>true</code> . See also <code>includeheadfoot</code> .
<code>ignoremp</code>	disregards the marginal notes in determining the horizontal margins (defaults to <code>true</code> ). If marginal notes overrun the page, the warning message will be displayed when <code>verbose=true</code> . See also <code>includemp</code> and Figure 5.



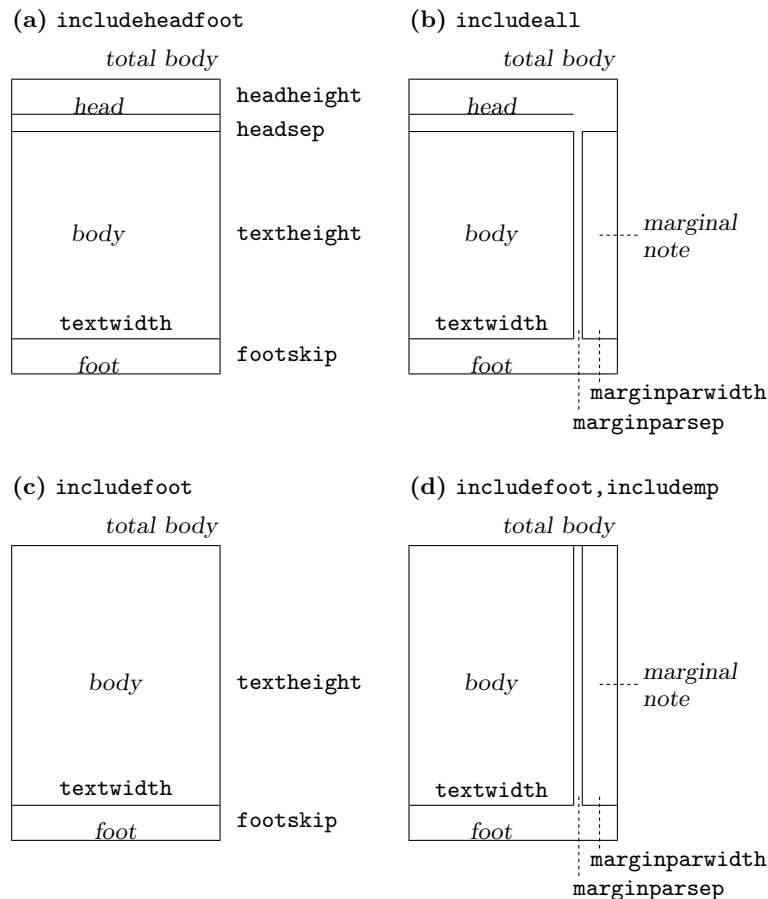


Figure 5: Sample layouts for *total body* with different switches. (a) `includeheadfoot`, (b) `includeall`, (c) `includefoot` and (d) `includefoot,includemp`. If `reversemp` is set to `true`, the location of the marginal notes are swapped on every page. Option `twoside` swaps both margins and marginal 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.

<code>left</code>   <code>lmargin</code>   <code>inner</code>	left margin (for <i>oneside</i> ) or inner margin (for <i>twoside</i> ) of <i>total body</i> . In other words, the distance between the left (inner) edge of the paper and that of <i>total body</i> . <code>left=⟨length⟩</code> . <code>inner</code> has no special meaning, just an alias of <code>left</code> and <code>lmargin</code> .
<code>right</code>   <code>rmargin</code>   <code>outer</code>	right or outer margin of <i>total body</i> . <code>right=⟨length⟩</code> .
<code>top</code>   <code>tmargin</code>	top margin of the page. <code>top=⟨length⟩</code> . Note this option has nothing to do with the native dimension <code>\topmargin</code> .
<code>bottom</code>   <code>bmargin</code>	bottom margin of the page. <code>bottom=⟨length⟩</code> .
<code>hmargin</code>	left and right margin. <code>hmargin={⟨left margin⟩,⟨right margin⟩}</code> or <code>hmargin=⟨length⟩</code> .
<code>vmargin</code>	top and bottom margin. <code>vmargin={⟨top margin⟩,⟨bottom margin⟩}</code> or <code>vmargin=⟨length⟩</code> .
<code>margin</code>	<code>margin={A,B}</code> is equivalent to <code>hmargin={A,B}</code> and <code>vmargin={A,B}</code> . <code>margin=A</code> is automatically expanded to <code>hmargin=A</code> and <code>vmargin=A</code> .
<code>hmarginratio</code>	horizontal margin ratio of <code>left</code> (inner) to <code>right</code> (outer). The value of <code>⟨ratio⟩</code> 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 <i>oneside</i> , 2:3 for <i>twoside</i> .
<code>vmarginratio</code>	vertical margin ratio of <code>top</code> to <code>bottom</code> . The default ratio is 2:3.
<code>marginratio</code>   <code>ratio</code>	horizontal and vertical margin ratios. <code>marginratio={⟨horizontal ratio⟩,⟨vertical ratio⟩}</code> or <code>marginratio=⟨ratio⟩</code> .
<code>hcentering</code>	sets auto-centering horizontally and is equivalent to <code>hmarginratio=1:1</code> . It is set to <code>true</code> by default for <i>oneside</i> . See also <code>hmarginratio</code> .
<code>vcentering</code>	sets auto-centering vertically and is equivalent to <code>vmarginratio=1:1</code> . The default is <code>false</code> . See also <code>vmarginratio</code> .
<code>centering</code>	sets auto-centering and is equivalent to <code>marginratio=1:1</code> . See also <code>marginratio</code> . The default is <code>false</code> . See also <code>marginratio</code> .
<code>twoside</code>	switches on <i>twoside</i> mode with left and right margins swapped on verso pages. The option sets <code>\@twoside</code> and <code>\@mparswitch</code> switches. See also <code>asymmetric</code> .
<code>asymmetric</code>	implements a <i>twosided</i> layout in which margins are not swapped on alternate pages (by setting <code>\oddsidemargin</code> to <code>\evensidemargin + bindingoffset</code> ) and in which the marginal notes stay always on the same side. This option can be used as an alternative to the <i>twoside</i> option. See also <code>twoside</code> .
<code>bindingoffset</code>	removes a specified space from the lefthand-side of the page for <i>oneside</i> or the inner-side for <i>twoside</i> . <code>bindingoffset=⟨length⟩</code> . This is useful if pages are bound by a press binding (glued, stitched, stapled ...). See Figure 6.
<code>hdivide</code>	See description in Section 5.3.
<code>vdivide</code>	See description in Section 5.3.
<code>divide</code>	See description in Section 5.3.

## 5.5 Native dimensions

The options below overwrite L<sup>A</sup>T<sub>E</sub>X native dimensions and switches for page layout (See the right-hand side in Figure 1).

<code>headheight</code>   <code>head</code>	modifies <code>\headheight</code> , height of header. <code>headheight=⟨length⟩</code> or <code>head=⟨length⟩</code> .
<code>headsep</code>	modifies <code>\headsep</code> , separation between header and text (body). <code>headsep=⟨length⟩</code> .
<code>footskip</code>   <code>foot</code>	modifies <code>\footskip</code> , distance separation between baseline of last line of text and baseline of footer. <code>footskip=⟨length⟩</code> or <code>foot=⟨length⟩</code> .

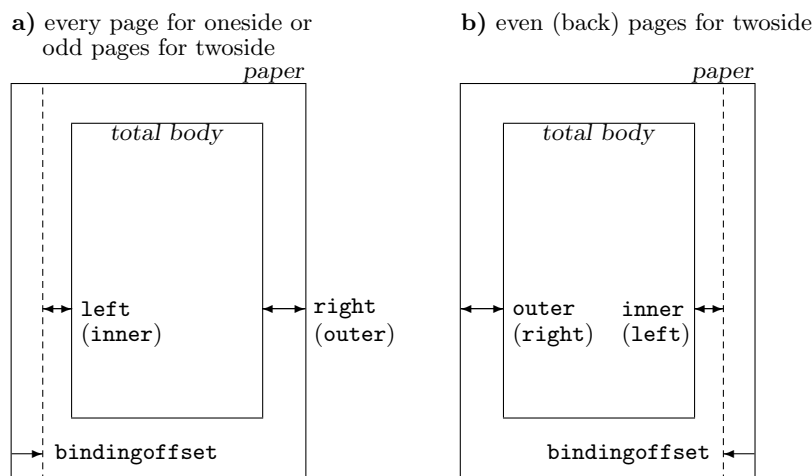


Figure 6: The option `bindingoffset` adds the specified length to the inner margin. Note that `twoside` option swaps the horizontal margins and the marginal notes together with `bindingoffset` on even pages (see **b**)), but `asymmetric` option suppresses the swap of the margins and marginal notes (but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

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

<sup>†</sup> `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.

<sup>†</sup> <code>dvips</code>	writes the paper size in dvi output with the <code>\special</code> macro. If you use <code>dvips</code> as a DVI-to-PS driver, for example, to print a document with <code>\geometry{a3paper,landscape}</code> on A3 paper in landscape orientation, you don't need options <code>"-t a3 -t landscape"</code> to <code>dvips</code> .
<sup>†</sup> <code>dvipdfm</code>	works like <code>dvips</code> except for landscape correction. You can set this option when using <code>dvipdfmx</code> and <code>xdvipdfmx</code> to process the dvi output.
<sup>†</sup> <code>pdftex</code>	sets <code>\pdfpagewidth</code> and <code>\pdfpageheight</code> internally.
<sup>†</sup> <code>xetex</code>	is the same as <code>pdftex</code> except for ignoring <code>\pdf{h,v}origin</code> 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 <code>pdftex</code> , is no longer necessary, but has no problem even though it's left undeleted. Instead of <code>xetex</code> , you can specify <code>dvipdfm</code> with XeLaTeX if you want to use specials of <code>dvipdfm</code> XeTeX supports.
<sup>†</sup> <code>vtex</code>	sets dimensions <code>\mediawidth</code> and <code>\mediaheight</code> for VTeX. When this driver is selected (explicitly or automatically), <code>geometry</code> 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.

<sup>†</sup> <code>verbose</code>	displays the parameter results on the terminal. <code>verbose=false</code> (default) still puts them into the log file.
<sup>†</sup> <code>reset</code>	sets back the layout dimensions and switches to the settings before <code>geometry</code> is loaded. Options given in <code>geometry.cfg</code> are also cleared. Note that this cannot reset <code>pass</code> and <code>mag</code> with <code>trueedimen</code> . <code>reset=false</code> has no effect and cannot cancel the previous <code>reset(=true)</code> if any. For example, when you go <pre> \documentclass[landscape]{article} \usepackage[twoside,reset,left=2cm]{geometry} </pre> with <code>\ExecuteOptions{scale=0.9}</code> in <code>geometry.cfg</code> , then as a result, <code>landscape</code> and <code>left=2cm</code> remain effective, and <code>scale=0.9</code> and <code>twoside</code> are ineffective.
<sup>†</sup> <code>mag</code>	sets magnification value ( <code>\mag</code> ) and automatically modifies <code>\hoffset</code> and <code>\voffset</code> according to the magnification. <code>mag=&lt;value&gt;</code> . Note that <code>&lt;value&gt;</code> should be an integer value with 1000 as a normal size. For example, <code>mag=1414</code> with <code>a4paper</code> provides an enlarged print fitting in <code>a3paper</code> , which is 1.414 ( $=\sqrt{2}$ ) times larger than <code>a4paper</code> . Font enlargement needs extra disk space. <b>Note that setting <code>mag</code> should precede any other settings with 'true' dimensions, such as <code>1.5truein</code>, <code>2truecm</code> and so on.</b> See also <code>trueedimen</code> option.
<sup>†</sup> <code>trueedimen</code>	changes all internal explicit dimension values into <i>true</i> dimensions, e.g., <code>1in</code> is changed to <code>1truein</code> . Typically this option will be used together with <code>mag</code> option. Note that this is ineffective against externally specified dimensions. For example, when you set <code>"mag=1440, margin=10pt, trueedimen"</code> , margins are not 'true' but magnified. If you want to set exact margins, you should set like <code>"mag=1440, margin=10truept, trueedimen"</code> instead.
<sup>†</sup> <code>pass</code>	disables all of the <code>geometry</code> options and calculations except <code>verbose</code> and <code>showframe</code> . It can be used for checking out the page layout of the <code>documentclass</code> , other packages and manual settings without <code>geometry</code> .
<sup>†</sup> <code>showframe</code>	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,

`[left=2cm, right=3cm]` is equivalent to `[right=3cm, left=2cm]`.

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

`[verbose=true, verbose=false]` results in `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 higher priority to the more concrete specification. Here is the priority rule for *body*.

priority:    low     $\longrightarrow$     high

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

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]`.

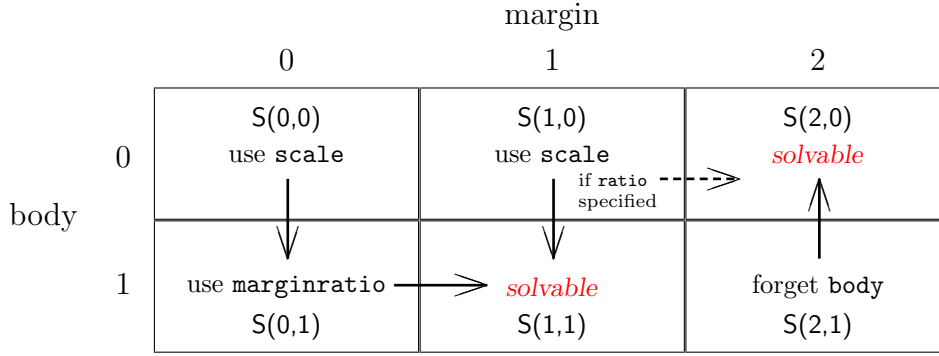


Figure 7: Specifications  $S(0,0)$  to  $S(2,1)$  and the completion rules (arrows). The horizontal and vertical denote the number of explicitly specified lengths for margin and body respectively.  $S(m,b)$  denote a specification with a set of the numbers  $(\text{margin}, \text{body}) = (m,b)$ .

## 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,

$$\text{top} : \text{bottom} = 2 : 3 \quad \text{default.} \quad (6)$$

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

$$\text{left (inner)} : \text{right (outer)} = \begin{cases} 1 : 1 & \text{default for oneside,} \\ 2 : 3 & \text{default for twoside.} \end{cases} \quad (7)$$

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  $0.7 \times \text{paper}$ )
- `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  $(\text{margin}, \text{body}) = (m,b)$ .

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  $0.7 \backslash \text{paperwidth}$ . 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

$$\text{top} : \text{bottom} = 2 : 3 \quad \text{default.} \quad (8)$$

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

$$\text{left (inner)} : \text{right (outer)} = \begin{cases} 1 : 1 & \text{default for oneside,} \\ 2 : 3 & \text{default for twoside.} \end{cases} \quad (9)$$

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

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

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

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

then `S(1,0)` goes to `S(1,1)`, in which `bottom` is calculated with `paperheight - (height + 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

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

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}

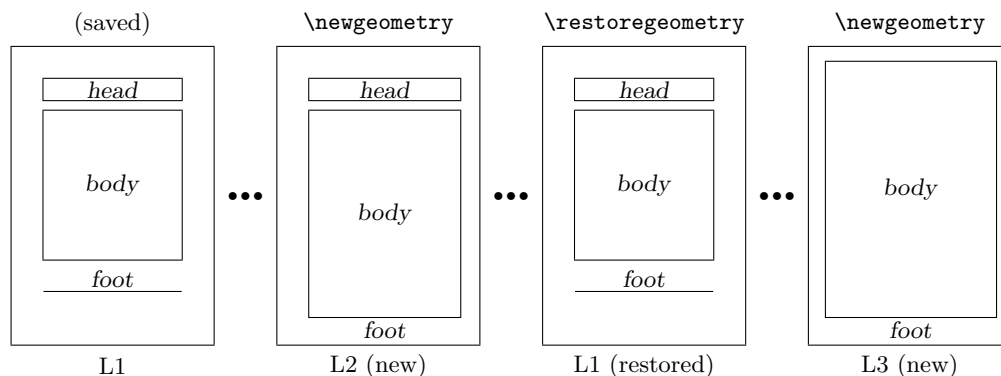


Layout L3 (new)



\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  $0.7 \times 1\text{cm}$  ( $=0.7\text{cm}$ ) 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

The author appreciates helpful suggestions and comments from many people including (in alphabetical order of family name): Jean-Bernard Addor, Frank Bennett, Alexis Dimitriadis, Friedrich Flender, Adrian Heathcote, Stephan Hennig, Morten Høgholm, Jonathan Kew, James Kilfiger, Jean-Marc Lasgouttes, Włodzimierz Macewicz, Frank Mittelbach, Eckhard Neber, Rolf Niepraschk, Hans Fr. Nordhaug, Heiko Oberdiek, Keith Reckdahl, Peter Riocreux, Will Robertson, Pablo Rodriguez, Nico Schlöemer, Perry C. Stearns, Frank Stengel, Plamen Tanovski, Petr Uher, Piet van Oostrum, Vladimir Volovich, Michael Vulis.

## 11 Implementation

1 (\*package)

This package requires three other packages: `keyval` in L<sup>A</sup>T<sub>E</sub>X 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` 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` An additional inner offset for binding.

```
23 \newdimen\Gm@bindingoffset
```

`\Gm@wd@mp` Correction lengths for `\textwidth`, `\oddsidemargin` and `\evensidemargin` in `includemp` mode.

```
\Gm@odd@mp 24 \newdimen\Gm@wd@mp
\Gm@even@mp 25 \newdimen\Gm@odd@mp
26 \newdimen\Gm@even@mp
```

`\Gm@layoutwidth` The dimensions for the layout.

```
\Gm@layoutheight 27 \newdimen\Gm@layoutwidth
\Gm@layouthoffset 28 \newdimen\Gm@layoutheight
\Gm@layoutvoffset 29 \newdimen\Gm@layouthoffset
30 \newdimen\Gm@layoutvoffset
```

`\Gm@dimlist` The token in which L<sup>A</sup>T<sub>E</sub>X Native dimensions are stored.

```
31 \newtoks\Gm@dimlist
```

`\Gm@warning` Macro for printing warning messages.

```
32 \def\Gm@warning#1{\PackageWarningNoLine{geometry}{#1}}%
```

`\ifGm@preamble` process the options only if it’s specified in the preamble.

```
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 the default value of horizontal *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.

`\Gm@Dvscale`

```

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

```

`\Gm@dvinfos` The driver names.

`\Gm@dviopdfm`

```

44 \def\Gm@dvinfos{dvips}%

```

`\Gm@pdftex`

```

45 \def\Gm@dviopdfm{dviopdfm}%

```

`\Gm@xetex`

```

46 \def\Gm@pdftex{pdftex}%

```

`\Gm@vtex`

```

47 \def\Gm@xetex{xetex}%
48 \def\Gm@vtex{vtex}%

```

`\Gm@true`

`\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`

```

53 \def\Gm@savelength#1{%
54   \edef\@tmp{\expandafter\the\csname #1\endcsname}%
55   \g@addto@macro\Gm@restore{\expandafter\noexpand\expandafter\csname
56     #1\endcsname\expandafter=\expandafter\the\csname #1\endcsname\relax}}

```

`\Gm@saveboolean`

```

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

```

`\Gm@restore`

```

63 \def\Gm@restore{%

```

`\Gm@save`

```

64 \def\Gm@save{%
65   \Gm@savelength{paperwidth}%
66   \Gm@savelength{paperheight}%
67   \Gm@savelength{textwidth}%
68   \Gm@savelength{textheight}%
69   \Gm@savelength{evensidemargin}%
70   \Gm@savelength{oddsidemargin}%
71   \Gm@savelength{topmargin}%
72   \Gm@savelength{headheight}%
73   \Gm@savelength{headsep}%
74   \Gm@savelength{topskip}%
75   \Gm@savelength{footskip}%
76   \Gm@savelength{baselineskip}%
77   \Gm@savelength{marginparwidth}%
78   \Gm@savelength{marginparsep}%

```

```

79 \Gm@savelength{columnsep}%
80 \Gm@savelength{hoffset}%
81 \Gm@savelength{voffset}
82 \Gm@savelength{Gm@layouthoffset}%
83 \Gm@savelength{Gm@layoutvoffset}%
84 \Gm@saveboolean{@twocolumn}%
85 \Gm@saveboolean{@twoside}%
86 \Gm@saveboolean{@mparswitch}%
87 \Gm@saveboolean{@reversemargin}}%

\Gm@initnewgm
88 \def\Gm@initnewgm{%
89 \Gm@dimlist={}
90 \Gm@hbodyfalse
91 \Gm@vbodyfalse
92 \Gm@heightroundedfalse
93 \Gm@includeheadfalse
94 \Gm@includefootfalse
95 \Gm@includempfalse
96 \let\Gm@width\@undefined
97 \let\Gm@height\@undefined
98 \let\Gm@textwidth\@undefined
99 \let\Gm@textheight\@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
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 set a boolean option.
\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** executes the third argument #3 if a boolean option #1 with its value #2 is 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 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
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@detiandiii 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@detiandiii#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@ratio\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@ratio\Gm@hmarginratio
219 \edef\Gm@Dmratio{\if@twoside\Gm@Dhratiotwo\else\Gm@Dhratio\fi}%
220 \else
221 \let\Gm@ratio\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@detiandiii{#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@detiandiii{#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@detiandiii{#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@detiandiii{#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}{#3}{#4}{#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 L<sup>A</sup>T<sub>E</sub>X 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@ansipaper}#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]j’ 383 \define@key{Gm}{a0paper}[true]{\Gm@setpaper@ifpre{a0paper}}%
384 \define@key{Gm}{a1paper}[true]{\Gm@setpaper@ifpre{a1paper}}%
‘ansi[a-e]paper’ 385 \define@key{Gm}{a2paper}[true]{\Gm@setpaper@ifpre{a2paper}}%
386 \define@key{Gm}{a3paper}[true]{\Gm@setpaper@ifpre{a3paper}}%
‘letterpaper’ 387 \define@key{Gm}{a4paper}[true]{\Gm@setpaper@ifpre{a4paper}}%
388 \define@key{Gm}{a5paper}[true]{\Gm@setpaper@ifpre{a5paper}}%
‘executivepaper’ 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}{ansipaper}[true]{\Gm@setpaper@ifpre{ansipaper}}%
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’ 413 \define@key{Gm}{paperwidth}{\ifGm@preamble{paperwidth}{%
‘papersize’ 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’ 419 \define@key{Gm}{layout}{\Gm@layouttrue\@nameuse{Gm@#1}{Gm@layout}}%
‘layoutheight’ 420 \let\KV@Gm@layoutname\KV@Gm@layout
‘layoutsizes’ 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}{layoutsizes}{\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’

```

432 \define@key{Gm}{hscale}{\Gm@hbodytrue\edef\Gm@hscale{#1}}%

```

‘scale’

```

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’

```

435 \define@key{Gm}{width}{\Gm@hbodytrue\Gm@defbylen{width}{#1}}%

```

‘totalwidth’

```

436 \define@key{Gm}{height}{\Gm@vbodytrue\Gm@defbylen{height}{#1}}%

```

‘totalheight’

```

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’

```

440 \define@key{Gm}{textwidth}{\Gm@hbodytrue\Gm@defbylen{textwidth}{#1}}%

```

‘text’

```

441 \define@key{Gm}{textheight}{\Gm@vbodytrue\Gm@defbylen{textheight}{#1}}%

```

‘body’

```

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’

```

445 \define@key{Gm}{includehead}[true]{\Gm@setbool{includehead}{#1}}%

```

‘includeheadfoot’

```

446 \define@key{Gm}{includefoot}[true]{\Gm@setbool{includefoot}{#1}}%

```

‘includemp’

```

447 \define@key{Gm}{includeheadfoot}[true]{\Gm@doifelse{includeheadfoot}{#1}%

```

‘includeall’

```

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’

```

454 \define@key{Gm}{ignorehead}[true]{%

```

‘ignoreheadfoot’

```

455   \Gm@setboolrev[ignorehead]{includehead}{#1}}%

```

‘ignoremp’

```

456 \define@key{Gm}{ignorefoot}[true]{%

```

‘ignoreall’

```

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’ 467 \define@key{Gm}{hdivide}{\Gm@parse@divide{#1}{lmargin}{width}{rmargin}}%  
‘divide’ 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 respectively.

‘tmargin’ 471 \define@key{Gm}{lmargin}{\Gm@defbylen{lmargin}{#1}}%  
‘bmargin’ 472 \define@key{Gm}{rmargin}{\Gm@defbylen{rmargin}{#1}}%  
‘left’ 473 \let\KV@Gm@left\KV@Gm@lmargin  
‘inner’ 474 \let\KV@Gm@inner\KV@Gm@lmargin  
‘innermargin’ 475 \let\KV@Gm@innermargin\KV@Gm@lmargin  
‘right’ 476 \let\KV@Gm@right\KV@Gm@rmargin  
‘outer’ 477 \let\KV@Gm@outer\KV@Gm@rmargin  
‘outermargin’ 478 \let\KV@Gm@outermargin\KV@Gm@rmargin  
‘top’ 479 \define@key{Gm}{tmargin}{\Gm@defbylen{tmargin}{#1}}%  
‘bottom’ 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’ 483 \define@key{Gm}{hmargin}{\Gm@branch{#1}{lmargin}{rmargin}}%  
‘margin’ 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’ 487 \define@key{Gm}{hmarginratio}{\edef\Gm@hmarginratio{#1}}%  
‘marginratio’ 488 \define@key{Gm}{vmarginratio}{\edef\Gm@vmarginratio{#1}}%  
‘hratio’ 489 \define@key{Gm}{marginratio}{\Gm@branch{#1}{hmarginratio}{vmarginratio}}%  
‘vratio’ 490 \let\KV@Gm@hratio\KV@Gm@hmarginratio  
‘ratio’ 491 \let\KV@Gm@vratio\KV@Gm@vmarginratio  
492 \let\KV@Gm@ratio\KV@Gm@marginratio

‘hcentering’ Useful shorthands to place *body* centered.

‘vcentering’ 493 \define@key{Gm}{hcentering}[true]{\Gm@doifelse{hcentering}{#1}}%  
‘centering’ 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’ 504 \define@key{Gm}{headheight}{\Gm@setlength\headheight{#1}}%  
‘footskip’ 505 \define@key{Gm}{headsep}{\Gm@setlength\headsep{#1}}%  
‘head’ 506 \define@key{Gm}{footskip}{\Gm@setlength\footskip{#1}}%  
‘foot’ 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 Opt.

‘nofoot’ 509 \define@key{Gm}{nohead}[true]{\Gm@doifelse{nohead}{#1}%

‘noheadfoot’ 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’ 517 \define@key{Gm}{marginparwidth}{\Gm@setlength\marginparwidth{#1}}%

‘marginparsep’ 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=Opt and \marginparsep=Opt.

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’ 526 \define@key{Gm}{layouthoffset}{\Gm@setlength\Gm@layouthoffset{#1}}%

‘layoutoffset’ 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}}%

‘reversemp’ The both options set \reversemargin.

‘reversemarginpar’ 531 \define@key{Gm}{reversemp}[true]{%

532 \Gm@doif{reversemp}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%

533 \define@key{Gm}{reversemarginpar}[true]{%

534 \Gm@doif{reversemarginpar}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%

‘ddiver’

535 \define@key{Gm}{driver}{\ifGm@preamble{driver}{%

536 \edef\@tempa{#1}\edef\@auto{auto}\edef\@none{none}%

537 \ifx\@tempa\@empty\let\Gm@driver\relax\else

538 \ifx\@tempa\@none\let\Gm@driver\relax\else

539 \ifx\@tempa\@auto\let\Gm@driver\@empty\else

540 \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’ 541 \define@key{Gm}{dvips}[true]{\ifGm@preamble{dvips}{%

‘pdftex’ 542 \Gm@doifelse{dvips}{#1}{\Gm@setdriver{dvips}}{\Gm@unsetdriver{dvips}}}}%

‘xetex’ 543 \define@key{Gm}{dvipdfm}[true]{\ifGm@preamble{dvipdfm}{%

‘vtex’ 544 \Gm@doifelse{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{\Gm@unsetdriver{dvipdfm}}}}%

545 \define@key{Gm}{pdftex}[true]{\ifGm@preamble{pdftex}{%



```

546 \Gm@doifelse{pdftex}{#1}{\Gm@setdriver{pdftex}}{\Gm@unsetdriver{pdftex}}}%
547 \define@key{Gm}{xetex}[true]{\ifGm@preamble{xetex}{%
548 \Gm@doifelse{xetex}{#1}{\Gm@setdriver{xetex}}{\Gm@unsetdriver{xetex}}}%
549 \define@key{Gm}{vtex}[true]{\ifGm@preamble{vtex}{%
550 \Gm@doifelse{vtex}{#1}{\Gm@setdriver{vtex}}{\Gm@unsetdriver{vtex}}}%

```

‘verbose’ The verbose mode.

```

551 \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.

```

552 \define@key{Gm}{reset}[true]{\ifGm@preamble{reset}{%
553 \Gm@doifelse{reset}{#1}{\Gm@restore@org\Gm@initall
554 \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.

```

555 \define@key{Gm}{resetpaper}[true]{\ifGm@preamble{resetpaper}{%
556 \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`.

```

557 \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`.

```

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

```

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

```

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

```

‘showframe’ The showframe option.

```

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

```

‘showcrop’ The showcrop option.

```

563 \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.

```

564 \def\Gm@setdefaultpaper{%
565 \ifx\Gm@paper\@undefined
566 \Gm@setsize{paper}(\strip@pt\paperwidth,\strip@pt\paperheight){pt}%
567 \Gm@setsize{Gm@layout}(\strip@pt\paperwidth,\strip@pt\paperheight){pt}%
568 \Gm@swap@papersizefalse
569 \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.

```

570 \def\Gm@adjustpaper{%
571 \ifdim\paperwidth>\p@else
572 \PackageError{geometry}{%
573 \string\paperwidth\space(\the\paperwidth) too short}{%
574 Set a paper type (e.g., ‘a4paper’).}%
575 \fi
576 \ifdim\paperheight>\p@else
577 \PackageError{geometry}{%
578 \string\paperheight\space(\the\paperheight) too short}{%

```

```

579   Set a paper type (e.g., 'a4paper').}%
580 \fi
581 \ifGm@swap@papersize
582   \setlength\@tempdima{\paperwidth}%
583   \setlength\paperwidth{\paperheight}%
584   \setlength\paperheight{\@tempdima}%
585 \fi
586 \ifGm@layout\else
587   \let\Gm@layoutwidth\paperwidth
588   \let\Gm@layoutheight\paperheight
589 \fi}%

```

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

```

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

```

**\Gm@adjustmp** The macro sets marginpar correction when includemp is set, which is used in \Gm@process. Local 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.

```

621 \def\Gm@adjustmp{%
622   \ifGm@includemp
623     \@tempdimb\marginparwidth
624     \advance\@tempdimb\marginparsep
625     \Gm@wd@mp\@tempdimb
626     \Gm@odd@mp\z@
627     \Gm@even@mp\z@
628     \if@twocolumn
629       \Gm@wd@mp2\@tempdimb
630       \Gm@odd@mp\@tempdimb
631       \Gm@even@mp\@tempdimb
632     \else

```

```

633     \if@reversemargin
634         \Gm@odd@mp\@tempdimb
635     \if@mparswitch\else
636         \Gm@even@mp\@tempdimb
637     \fi
638 \else
639     \if@mparswitch
640         \Gm@even@mp\@tempdimb
641     \fi
642 \fi
643 \fi
644 \fi}%

```

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

```

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

```

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

```

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

```

`\topskip` has to be adjusted so that the formula “`\textheight = (lines − 1) × \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`.

```

671         \ifdim\topskip<\ht\strutbox
672             \setlength\@tempdima{\topskip}%
673             \setlength\topskip{\ht\strutbox}%
674             \Gm@warning{\noexpand\topskip was changed from \the\@tempdima\space
675                 to \the\topskip}%
676         \fi
677         \setlength\@tempdima{\baselineskip}%
678         \multiply\@tempdima\Gm@lines
679         \addtolength\@tempdima{\topskip}%
680         \addtolength\@tempdima{-\baselineskip}%
681         \edef\Gm@textheight{\the\@tempdima}%
682     \fi
683     \ifx\Gm@textheight\@undefined\else
684         \setlength\@tempdima{\Gm@textheight}%
685     \ifGm@includehead

```

```

686      \addtolength\@tempdima{\headheight}%
687      \addtolength\@tempdima{\headsep}%
688      \fi
689      \ifGm@includefoot
690      \addtolength\@tempdima{\footskip}%
691      \fi
692      \edef\Gm@height{\the\@tempdima}%
693      \fi
694      \fi}%

```

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

```

695 \def\Gm@process{%

```

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

```

696   \ifGm@pass
697     \Gm@restore@org
698   \else
699     \Gm@@process
700   \fi}%

```

The main processing macro.

```

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

```

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

```

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

```

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

```

714   \ifGm@includemp
715     \advance\textwidth-\Gm@wd@mp
716     \advance\oddsidemargin\Gm@odd@mp
717   \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`.

```

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

```

The bindingoffset correction for `\oddsidemargin`.

```

727   \advance\oddsidemargin\Gm@bindingoffset
728   \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`.

```

729   \ifGm@includehead
730     \addtolength\textheight{-\headheight}%

```

```

731 \addtolength\textheight{-\headsep}%
732 \else
733 \addtolength\topmargin{-\headheight}%
734 \addtolength\topmargin{-\headsep}%
735 \fi

```

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

```

736 \ifGm@includefoot
737 \addtolength\textheight{-\footskip}%
738 \fi

```

If heightrounded is set, \textheight is rounded.

```

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

```

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

```

755 \advance\oddsidemargin\Gm@layouthoffset%
756 \advance\evensidemargin\Gm@layouthoffset%
757 \advance\topmargin\Gm@layoutvoffset%
758 \addtolength\Gm@layoutwidth{\Gm@bindingoffset}%
759 }% 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.

```

760 \def\Gm@detectdriver{%

```

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

```

761 \ifx\Gm@driver\@empty
762 \typeout{*geometry* driver: auto-detecting}%
763 \ifpdf is defined in ifpdf package in ‘oberdiek’ bundle.
764 \ifpdf
765 \Gm@setdriver{pdftex}%
766 \else
767 \Gm@setdriver{dvips}%
768 \fi

```

\ifvtex is defined in ifvtex package in ‘oberdiek’ bundle.

```

768 \ifvtex
769 \Gm@setdriver{vtex}%
770 \fi

```

xetex

```

771 \ifundefined{XeTeXversion}{\Gm@setdriver{xetex}}%

```

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

```

772 \else
773 \ifx\Gm@driver\Gm@xetex %%
774 \ifundefined{XeTeXversion}{\Gm@warning{%
775 Wrong driver setting: ‘xetex’; trying ‘pdftex’ driver}%

```

```

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

\Gm@showparams
790 \def\Gm@showparams#1{%
791     \ifGm@verbose\expandafter\typeout\else\expandafter\wlog\fi
792     {\Gm@logcontent{#1}}}%
793 \def\Gm@showbool#1{\@nameuse{ifGm@#1}#1\space\fi}%
794 \def\Gm@showdim#1{*gm \string#1=\the#1^^J}%

\Gm@logcontent The content of geometry parameters and native dimensions for the page layout.
795 \def\Gm@logcontent#1{%
796     *geometry* verbose: parameters #1^^J%
797     \ifGm@pass *gm: pass (ignores the geometry layout)^^J%
798     \else
799         \if\Gm@driver\relax *gm driver: <none>^^J\else *gm driver: \Gm@driver^^J\fi
800         \ifx\Gm@paper\@undefined *gm paper: \the\paperwidth, \the\paperheight^^J%
801         \else *gm paper: \Gm@paper^^J\fi
802         \ifGm@layout *gm layout: \the\Gm@layoutwidth, \the\Gm@layoutheight^^J%
803         \else *gm layout: (same as paper)^^J\fi
804         \@ifundefined{Gm@lines}{ }{*gm lines: \Gm@lines^^J}%
805         \@ifundefined{Gm@hmarginratio}{ }{*gm hratio: \Gm@hmarginratio^^J}%
806         \@ifundefined{Gm@vmarginratio}{ }{*gm vratio: \Gm@vmarginratio^^J}%
807         \ifdim\Gm@bindingoffset=\z@ \else *gm bindingoffset: \the\Gm@bindingoffset^^J\fi
808         *gm L,W,R: \Gm@lmargin, \Gm@width, \Gm@rmargin^^J%
809         *gm T,H,B: \Gm@tmargin, \Gm@height, \Gm@bmargin^^J%
810         *gm switch-on: %
811         \Gm@showbool{landscape}%
812         \Gm@showbool{includehead}%
813         \Gm@showbool{includefoot}%
814         \Gm@showbool{includemp}%
815         \if@twoside twoside\space\fi%
816         \if@mparswitch\else\if@twoside asymmetric\space\fi\fi%
817         \Gm@showbool{heightrounded}%
818         \ifx\Gm@truedimen\@empty\else *gm truedimen\space\fi%
819     ^^J%
820 \fi
821 \Gm@showdim{\paperwidth}%
822 \Gm@showdim{\paperheight}%
823 \Gm@showdim{\textwidth}%
824 \Gm@showdim{\textheight}%
825 \Gm@showdim{\oddsidemargin}%
826 \Gm@showdim{\evensidemargin}%
827 \Gm@showdim{\topmargin}%
828 \Gm@showdim{\headheight}%
829 \Gm@showdim{\headsep}%
830 \Gm@showdim{\footskip}%
831 \Gm@showdim{\marginparwidth}%
832 \Gm@showdim{\marginparsep}%

```

```

833 \Gm@showdim{\columnsep}%
834 *gm \string\skip\string\footins=\the\skip\footins^^J%
835 \Gm@showdim{\hoffset}%
836 \Gm@showdim{\voffset}%
837 \Gm@showdim{\mag}%
838 \if@twocolumn *gm\space\string\@twocolumntrue^^J\fi%
839 \if@twoside *gm\space\string\@twosidetrue^^J\fi%
840 \if@mparswitch *gm\space\string\@mparswitchtrue^^J\fi%
841 \if@reversemargin *gm\space\string\@reversemargintrue^^J\fi%
842 *gm (1in=72.27pt, 1cm=28.4528pt)}%

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

```

```

892      \Gm@vrules@mpii
893      \fi
894      \else
895      \Gm@vrules@mpi
896      \fi
897      \vb@xt@\z@\vss\hrule width\textwidth}%
898      \vskip\footskip\vb@xt@\z@\vss\hrule width\textwidth}%
899      \vss}%
900      \fi%
901  }}%
902  \let\Gm@shipout\shipout
903  \newcommand*{\gm@shipi}{%
904    \ifvoid@cclv\expandafter\aftergroup\fi\gm@shipii}%
905  \newcommand*{\gm@shipii}{%
906    \ifvoid@cclv
907      \Gm@shipout\box@cclv
908    \else
909      \ifGm@showframe
910        \Gm@shipout\ vbox{\Gm@pageframes\ifvbox@cclv\unvbox\else\box\fi@cclv}%
911      \else\ifGm@showcrop
912        \Gm@shipout\ vbox{\Gm@pageframes\ifvbox@cclv\unvbox\else\box\fi@cclv}%
913      \else
914        \Gm@shipout\box@cclv
915      \fi\fi
916    \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.

```

917 \def\ProcessOptionsKV{\@ifnextchar[%]
918   {\@ProcessOptionsKV}{\@ProcessOptionsKV[]}}%
919 \def\@ProcessOptionsKV[#1]#2{%
920   \let\@tempa\@empty
921   \@tempcnta\z@
922   \if#1p\@tempcnta\@ne\else\if#1c\@tempcnta\tw@\fi\fi
923   \ifodd\@tempcnta
924     \edef\@tempa{\@optionlist{\@currname.\@current}}%
925   \else
926     \@for\CurrentOption:=\@classoptionslist\do{%
927       \ifundefined{KV@#2\CurrentOption}%
928         {\edef\@tempa{\@tempa,\CurrentOption,}}}%
929     \ifnum\@tempcnta=\z@
930       \edef\@tempa{\@tempa,\@optionlist{\@currname.\@current}}%
931     \fi
932   \fi
933   \edef\@tempa{\noexpand\setkeys{#2}{\@tempa}}%
934   \@tempa
935   \AtEndOfPackage{\let\@unprocessedoptions\relax}}%
936 \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.

```

937 \def\Gm@processconfig{%
938   \let\Gm@origExecuteOptions\ExecuteOptions
939   \let\ExecuteOptions\Gm@setkeys
940   \InputIfFileExists{geometry.cfg}{-}{-}
941   \let\ExecuteOptions\Gm@origExecuteOptions}%

```

The original page layout before loading geometry is saved with \Gm@save and can be called by \Gm@restore@org. which will be used when reset or pass is specified.



```

942 \Gm@save
943 \edef\Gm@restore@org{\Gm@restore}%
944 \Gm@initall
    Processing config file.
945 \Gm@processconfig
    The optional arguments to \documentclass are processed here.
946 \ProcessOptionsKV[c]{Gm}%
    Paper dimensions given by class default are stored.
947 \Gm@setdefaultpaper
    The optional arguments to \usepackage are processed here.
948 \ProcessOptionsKV[p]{Gm}%
    Actual settings and calculation for layout dimensions are processed.
949 \Gm@process
    verbose, showframe and driver options are processed at \begin{document}.
950 \AtBeginDocument{%
    Paper size is temporally adjusted according to \mag for printing devices.
951 \ifGm@resetpaper
952     \edef\Gm@pw{\Gm@orgpw}%
953     \edef\Gm@ph{\Gm@orgph}%
954 \else
955     \edef\Gm@pw{\the\paperwidth}%
956     \edef\Gm@ph{\the\paperheight}%
957 \fi
    If pass is set to true, no adjustment for page dimensions is done.
958 \ifGm@pass\else
959     \ifnum\mag=\@m\else
960         \Gm@magtooffset
961         \divide\paperwidth\@m
962         \multiply\paperwidth\the\mag
963         \divide\paperheight\@m
964         \multiply\paperheight\the\mag
965     \fi
966 \fi
    Checking the driver options.
967 \Gm@detectdriver
    xetex
968 \ifx\Gm@driver\Gm@xetex
969     \ifundefined{pdfpagewidth}{\%
970         \setlength\pdfpagewidth{\Gm@pw}%
971         \setlength\pdfpageheight{\Gm@ph}}%
972     \ifnum\mag=\@m\else
973         \ifx\Gm@truedimen\Gm@true
974             \setlength\paperwidth{\Gm@pw}%
975             \setlength\paperheight{\Gm@ph}%
976         \fi
977     \fi
978 \fi
    If pdftex is set to true, pdf-commands are set properly. To avoid pdftex magnification problem,
    \pdfhorigin and \pdfvorigin are adjusted for \mag.
979 \ifx\Gm@driver\Gm@pdftex
980     \ifundefined{pdfpagewidth}{\%
981         \setlength\pdfpagewidth{\Gm@pw}%
982         \setlength\pdfpageheight{\Gm@ph}}%
983     \ifnum\mag=\@m\else

```

```

984 \tempdima=\mag sp%
985 \ifundefined{pdfhorigin}{\%
986 \divide\pdfhorigin\tempdima
987 \multiply\pdfhorigin\@m
988 \divide\pdfvorigin\tempdima
989 \multiply\pdfvorigin\@m}%
990 \ifx\Gm@truedimen\Gm@true
991 \setlength\paperwidth{\Gm@pw}%
992 \setlength\paperheight{\Gm@ph}%
993 \fi
994 \fi
995 \fi

```

With V<sub>T</sub>E<sub>X</sub> environment, V<sub>T</sub>E<sub>X</sub> variables are set here.

```

996 \ifx\Gm@driver\Gm@vtex
997 \ifundefined{mediawidth}{\%
998 \mediawidth=\paperwidth
999 \mediaheight=\paperheight}%
1000 \ifvtexdvi
1001 \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1002 \fi
1003 \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.

```

1004 \ifx\Gm@driver\Gm@dvips
1005 \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1006 \ifx\Gm@driver\Gm@dvips\ifGm@landscape
1007 \AtBeginDvi{\special{! /landplus90 true store}}%
1008 \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.

```

1009 \else\ifx\Gm@driver\Gm@dvipdfm
1010 \ifcase\ifx\AtBeginShipoutFirst\relax\@ne\else
1011 \ifx\AtBeginShipoutFirst\@undefined\@ne\else\z@\fi\fi
1012 \AtBeginShipoutFirst{\special{papersize=\the\paperwidth,\the\paperheight}}%
1013 \or
1014 \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1015 \fi
1016 \fi\fi

```

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

```

1017 \ifGm@showframe
1018 \renewcommand*{\shipout}{\afterassignment\gm@shipi\setbox\@cclv=}
1019 \else\ifGm@showcrop
1020 \renewcommand*{\shipout}{\afterassignment\gm@shipi\setbox\@cclv=}
1021 \fi\fi

```

Here the layout dimensions are saved. The layout can be restored by \restoregeometry command.

```

1022 \Gm@save
1023 \edef\Gm@restore@pkg{\Gm@restore}%

```

If verbose=true and pass=false, the system checks if marginpars overrun the page.

```

1024 \ifGm@verbose\ifGm@pass\else\Gm@checkmp\fi\fi

```

If verbose=true the parameter results are displayed on the terminal. verbose=false (default) still puts them into the log file.

```

1025 \setlength\tempdima{\Gm@width}\edef\Gm@width{\the\tempdima}%
1026 \setlength\tempdima{\Gm@height}\edef\Gm@height{\the\tempdima}%
1027 \Gm@showparams{in the preamble}%
1028 }% end of \AtBeginDocument

```

`\geometry` The user-interface macro `\geometry` is defined here. This command should be used in the preamble.

```
1029 \newcommand{\geometry}[1]{%
1030   \Gm@clean
1031   \setkeys{Gm}{#1}%
1032   \Gm@process}%
1033 \@onlypreamble\geometry
```

`\Gm@changelayout`

```
1034 \DeclareRobustCommand\Gm@changelayout{%
1035   \setlength{\@colht}{\textheight}\setlength{\@colroom}{\textheight}%
1036   \setlength{\vsize}{\textheight}\setlength{\columnwidth}{\textwidth}%
1037   \if@twocolumn%
1038     \advance\columnwidth-\columnsep \divide\columnwidth\tw@%
1039     \@firstcolumntrue%
1040   \fi%
1041   \setlength{\hsize}{\columnwidth}%
1042   \setlength{\linewidth}{\hsize}}%
```

`\newgeometry`

```
1043 \newcommand\newgeometry[1]{%
1044   \clearpage
1045   \Gm@restore@org
1046   \Gm@initnewgm
1047   \Gm@newgmtrue
1048   \setkeys{Gm}{#1}%
1049   \Gm@newgmfalse
1050   \Gm@process
1051   \Gm@changelayout
1052   \Gm@showparams{by \string\newgeometry}}%
```

`\restoregeometry`

```
1053 \newcommand\restoregeometry{%
1054   \clearpage
1055   \Gm@restore@pkg
1056   \Gm@changelayout}%
```

`\savegeometry`

```
1057 \newcommand*\savegeometry[1]{%
1058   \Gm@save
1059   \expandafter\edef\csname Gm@restore@@#1\endcsname{\Gm@restore}}%
```

`\loadgeometry`

```
1060 \newcommand*\loadgeometry[1]{%
1061   \clearpage
1062   \@ifundefined{Gm@restore@@#1}{%
1063     \PackageError{geometry}{%
1064       \string\loadgeometry : name ‘#1’ undefined}{%
1065       The name ‘#1’ should be predefined with \string\savegeometry}%
1066     }{\@nameuse{Gm@restore@@#1}}%
1067   \Gm@changelayout}}%
1068 \end{package}
```

## 12 Config file

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

```
1069 \*config>
1070 %<<SAVE_INTACT
```

```

1071
1072 % Uncomment and edit the line below to set default options.
1073 %\ExecuteOptions{a4paper}
1074
1075 %SAVE_INTACT
1076 </config>

```

## 13 Sample file

Here is an executable sample tex file.

```

1077 <samples>
1078 %<<SAVE_INTACT
1079 \documentclass{article}% uses letterpaper by default
1080 % \documentclass[a4paper]{article}% for A4 paper
1081 %-----
1082 % Edit and uncomment one of the settings below
1083 %-----
1084 % \usepackage{geometry}
1085 % \usepackage[centering]{geometry}
1086 % \usepackage[width=10cm,vscale=.7]{geometry}
1087 % \usepackage[margin=1cm, papersize={12cm,19cm}, resetpaper]{geometry}
1088 % \usepackage[margin=1cm,includeheadfoot]{geometry}
1089 \usepackage[margin=1cm,includeheadfoot,includemp]{geometry}
1090 % \usepackage[margin=1cm,bindingoffset=1cm,twoside]{geometry}
1091 % \usepackage[hmarginratio=2:1, vmargin=2cm]{geometry}
1092 % \usepackage[hscale=0.5,twoside]{geometry}
1093 % \usepackage[hscale=0.5,asymmetric]{geometry}
1094 % \usepackage[hscale=0.5,heightrounded]{geometry}
1095 % \usepackage[left=1cm,right=4cm,top=2cm,includefoot]{geometry}
1096 % \usepackage[lines=20,left=2cm,right=6cm,top=2cm,twoside]{geometry}
1097 % \usepackage[width=15cm, marginparwidth=3cm, includemp]{geometry}
1098 % \usepackage[hdivide={1cm,,2cm}, vdivide={3cm,8in,,}, nohead]{geometry}
1099 % \usepackage[headsep=20pt, head=40pt,foot=20pt,includeheadfoot]{geometry}
1100 % \usepackage[text={6in,8in}, top=2cm, left=2cm]{geometry}
1101 % \usepackage[centering,includemp,twoside,landscape]{geometry}
1102 % \usepackage[mag=1414,margin=2cm]{geometry}
1103 % \usepackage[mag=1414,margin=2truecm,truedimen]{geometry}
1104 % \usepackage[a5paper, landscape, twocolumn, twoside,
1105 %     left=2cm, hmarginratio=2:1, includemp, marginparwidth=43pt,
1106 %     bottom=1cm, foot=.7cm, includefoot, textheight=11cm, heightrounded,
1107 %     columnsep=1cm,verbose]{geometry}
1108 %-----
1109 % No need to change below
1110 %-----
1111 \geometry{verbose,showframe}% options appended.
1112 \newcommand\mynote{\marginpar%
1113 [\raggedright\rule{\marginparwidth}{.7pt}\\A left side note.]}%
1114 {\raggedright\rule{\marginparwidth}{.7pt}\\A side note.}}%
1115 \def\fox{A quick brown fox jumps over the lazy dog. }
1116 \def\fivefoxes{\fox\fox\fox\fox\fox}
1117 \def\manyfoxes{\fivefoxes\mynote\fivefoxes\par\fivefoxes\fivefoxes\par}
1118 % \let\mynote\relax % removes marginal notes.
1119 \begin{document}
1120 \manyfoxes\manyfoxes\manyfoxes\manyfoxes
1121 \manyfoxes\manyfoxes\manyfoxes\manyfoxes
1122 \manyfoxes\manyfoxes\manyfoxes\manyfoxes
1123 \end{document}
1124 %SAVE_INTACT
1125 </samples>

```