

The geometry package

Hideo Umeki
latexgeometry@gmail.com

2010/03/01 v5.2

Abstract

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

1 Preface to version 5

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

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

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

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

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

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

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

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

- **Changing default for underspecified margin.**

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

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

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

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

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

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

2 Introduction

To set dimensions for page layout in L^AT_EX is not straightforward. You need to adjust several L^AT_EX 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*.

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

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:

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

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

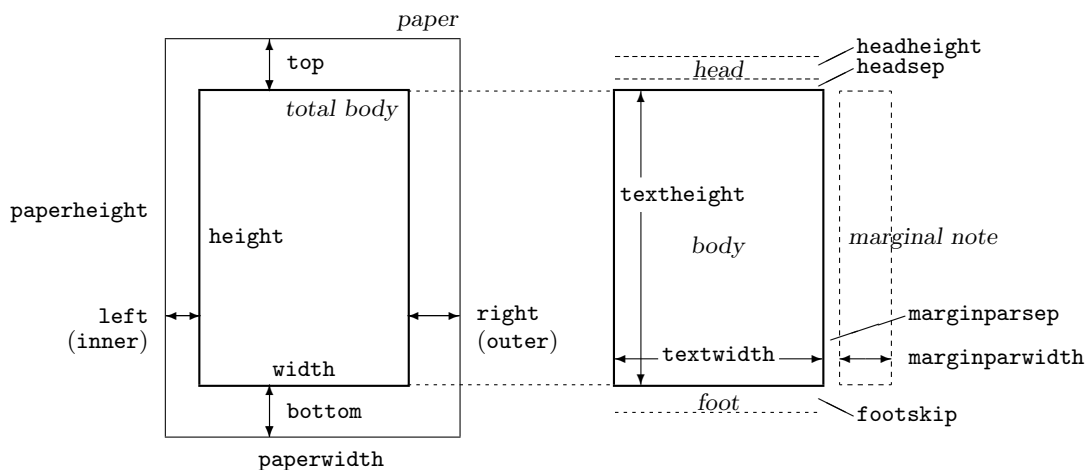


Figure 1: Dimension names used in the `geometry` package. `width=textwidth` and `height=textheight` by default. `left`, `right`, `top` and `bottom` are margins. If margins on verso pages are swapped by `twoside` option, margins specified by `left` and `right` options are used for the inside and outside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

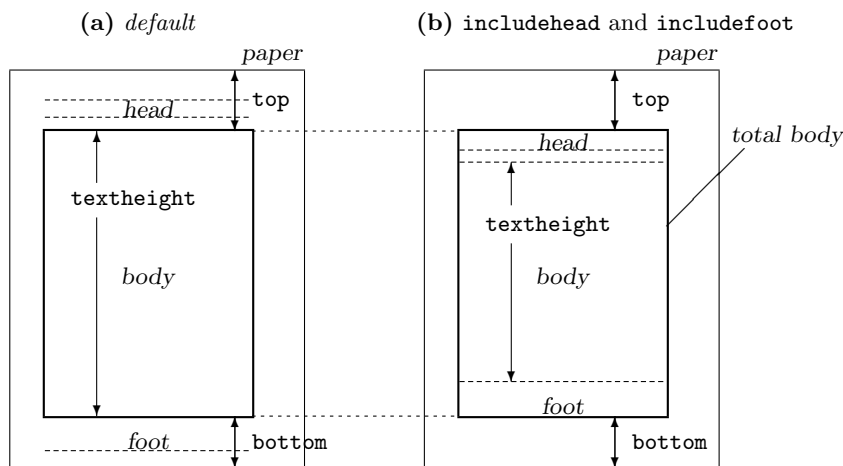


Figure 2: `includehead` and `includefoot` include the head and foot respectively into `total body`. (a) `height = textheight` (default). (b) `height = textheight + headheight + headsep + footskip` if `includehead` and `includefoot`. If the top and bottom margins are specified, `includehead` and `includefoot`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

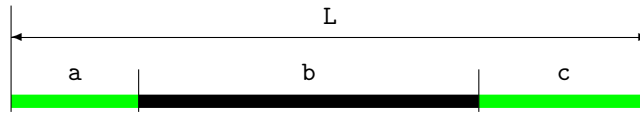


Figure 3: A simple model of page dimensions.

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*¹. For example,

```

\usepackage{calc}
\usepackage[texheight=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 [†] are not available as an argument of `\newgeometry` (See Section 7).

¹CTAN: macros/latex/required/tools

5.1 Paper size

The options below set paper/media size and orientation.

† <code>paper</code> <code>papername</code>	specifies the paper size by name. <code>paper=<paper-name></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> .
† <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>ansiapaper</code> , <code>ansibpaper</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.
† <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.
† <code>paperwidth</code>	width of the paper. <code>paperwidth=<length></code> .
† <code>paperheight</code>	height of the paper. <code>paperheight=<length></code> .
† <code>papersize</code>	width and height of the paper. <code>papersize={<width>,<height>}</code> or <code>papersize=<length></code> .
† <code>landscape</code>	switches the paper orientation to landscape mode.
† <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=<paper-name></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=<length></code> .
<code>layoutheight</code>	height of the layout. <code>layoutheight=<length></code> .
<code>layoutsizes</code>	width and height of the layout. <code>layoutsizes={<width>,<height>}</code> or <code>layoutsizes=<length></code> .
<code>layoutoffset</code>	specifies the horizontal offset from the left edge of the paper. <code>layoutoffset=<length></code> .
<code>layoutvoffset</code>	specifies the vertical offset from the top edge of the paper. <code>layoutvoffset=<length></code> .
<code>layoutoffset</code>	specifies both horizontal and vertical offsets. <code>layoutoffset={<hoffset>,<voffset>}</code> or <code>layoutsizes=<length></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=<h-scale></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=<v-scale></code> . (0.7 by default) <code>vscale=0.9</code> is equivalent to <code>height=0.9\paperheight</code> .

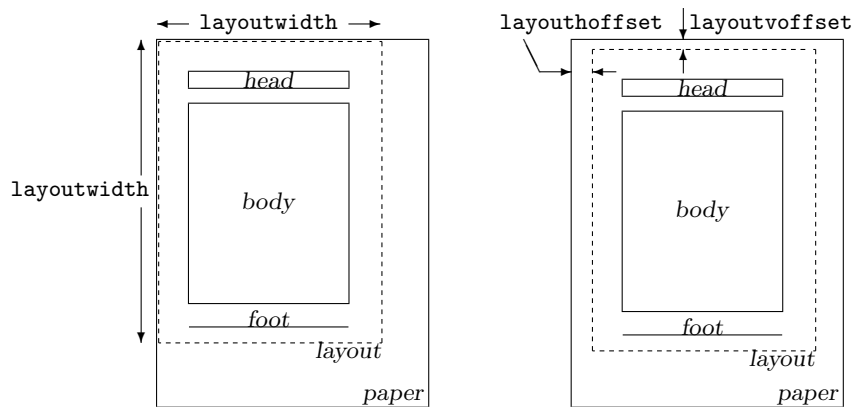


Figure 4: The dimensions related to the layout size. Note that the layout size defaults to the same size as the paper, so you don't have to specify layout-related options explicitly in most cases.

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

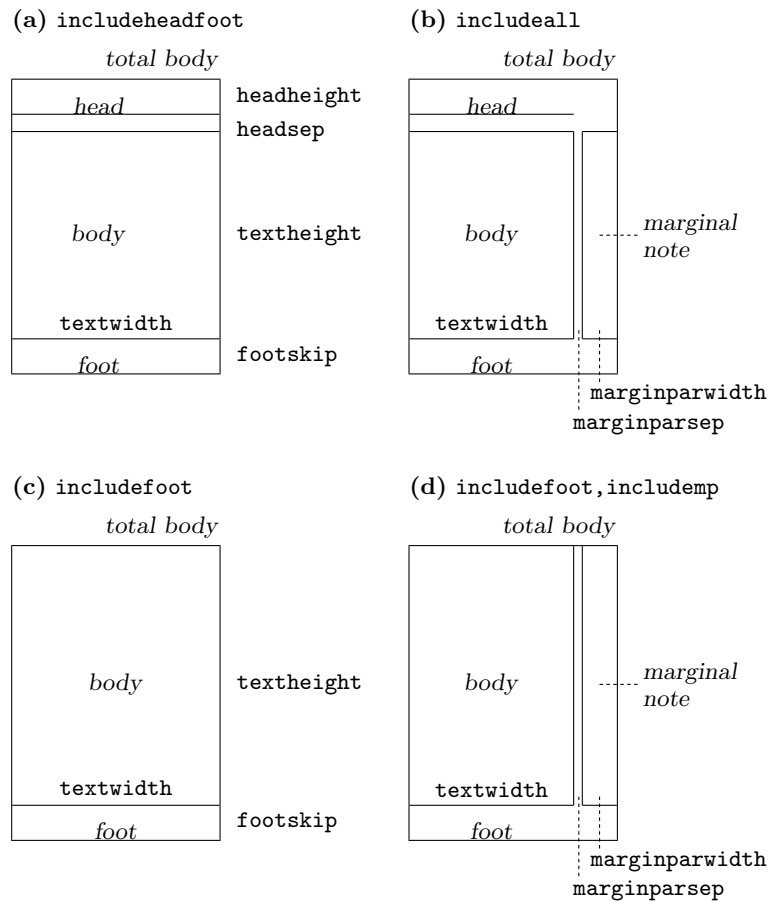


Figure 5: Sample layouts for *total body* with different switches. (a) `includeheadfoot`, (b) `includeall`, (c) `includefoot` and (d) `includefoot,includemp`. If `reversemp` is set to `true`, the location of the marginal notes are swapped on every page. Option `twoside` swaps both margins and marginal 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={\langle left margin \rangle, \langle width \rangle, \langle right margin \rangle}`. 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={\langle top margin \rangle, \langle height \rangle, \langle bottom margin \rangle}`.

`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 oneside) or inner margin (for twoside) 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 oneside, 2:3 for twoside.
<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 oneside. 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 twoside 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 twosided 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 <code>twoside</code> option. See also <code>twoside</code> .
<code>bindingoffset</code>	removes a specified space from the lefthand-side of the page for oneside or the inner-side for twoside. <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^AT_EX 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> .

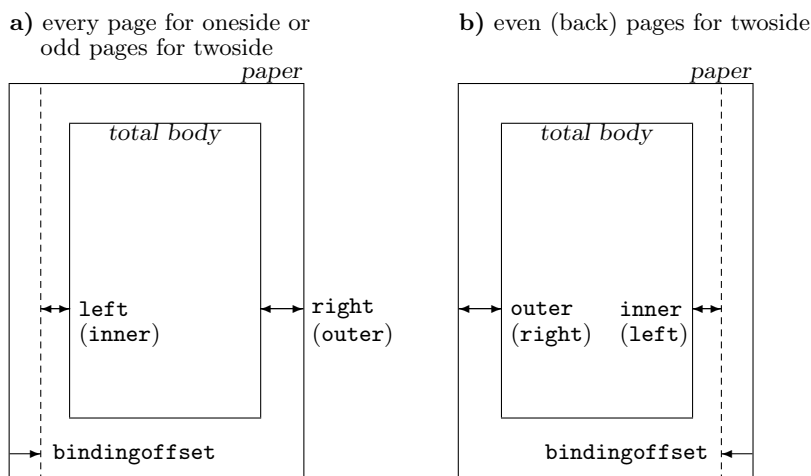


Figure 6: The option `bindingoffset` adds the specified length to the inner margin. Note that `twoside` option swaps the horizontal margins and the marginal notes together with `bindingoffset` on even pages (see **b**)), but `asymmetric` option suppresses the swap of the margins and marginal notes (but `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 <code>0pt</code> .
<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=<length></code> .
<code>marginparsep</code>	modifies <code>\marginparsep</code> , separation between body and marginal notes. <code>marginparsep=<length></code> .
<code>nomarginpar</code>	shrinks spaces for marginal notes to <code>0pt</code> , 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=<length></code> .
<code>voffset</code>	modifies <code>\voffset</code> . <code>voffset=<length></code> .
<code>offset</code>	horizontal and vertical offset. <code>offset={<hoffset>,<voffset>}</code> or <code>offset=<length></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> . Instead of <code>twocolumn=false</code> , you can specify <code>onecolumn</code> (which defaults to <code>true</code>)
<code>onecolumn</code>	works as <code>twocolumn=false</code> . On the other hand, <code>onecolumn=false</code> is equivalent to <code>twocolumn</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.

† <code>driver</code>	specifies the driver with <code>driver=<driver name></code> . <code>dvips</code> , <code>dvipdfm</code> , <code>pdftex</code> , <code>vtex</code> , <code>xetex</code> , <code>auto</code> and <code>none</code> are available as a driver name. The names except for <code>auto</code> and <code>none</code> can be specified directly with the name without <code>driver=</code> . <code>driver=auto</code> makes the auto-detection work whatever the previous setting is. <code>driver=none</code> 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 <code>crop</code> package with <code>geometry</code> , you should call <code>\usepackage[driver=none]{geometry}</code> before the <code>crop</code> package.
† <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> .
† <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.
† <code>pdftex</code>	sets <code>\pdfpagewidth</code> and <code>\pdfpageheight</code> internally.
† <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 ‘ <code>geometry.cfg</code> ’ 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.
† <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.

† <code>verbose</code>	displays the parameter results on the terminal. <code>verbose=false</code> (default) still puts them into the log file.
† <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>truedimen</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 style="margin-left: 40px;">\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.
† <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=<value></code> . Note that <code><value></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. 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. See also <code>truedimen</code> option.
† <code>truedimen</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, truedimen</code> ”, margins are not ‘true’ but magnified. If you want to set exact margins, you should set like “ <code>mag=1440, margin=10truept, truedimen</code> ” instead.
† <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 documentclass, other packages and manual settings without <code>geometry</code> .
† <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 \rightarrow high

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

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]`.

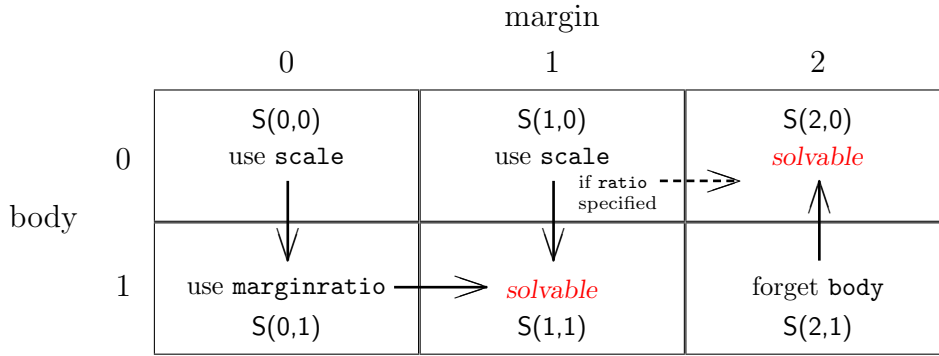


Figure 7: Specifications $S(0,0)$ to $S(2,1)$ and the completion rules (arrows). The horizontal and vertical denote the number of explicitly specified lengths for margin and body respectively. $S(m,b)$ denote a specification with a set of the numbers $(\text{margin}, \text{body}) = (m,b)$.

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,

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

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`.

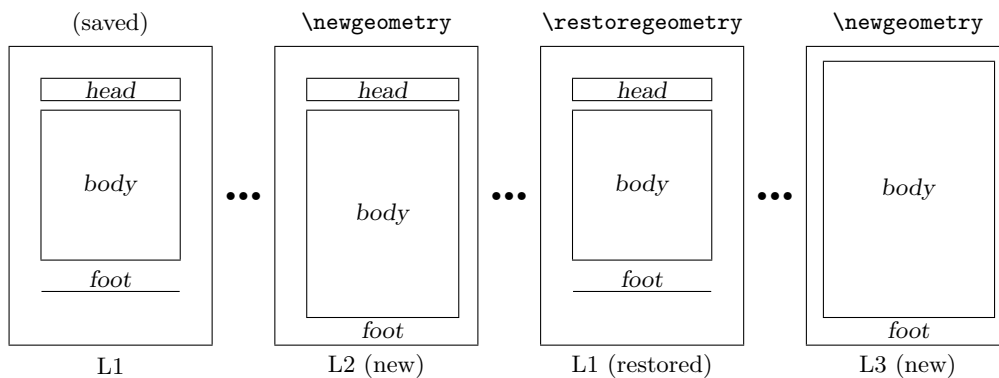
Note that both `\newgeometry` and `\restoregeometry` insert `\clearpage` where they are called.

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^AT_EX graphics bundle, `ifpdf` and `ifvtex` in ‘`oberdiek`’ bundle.

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

Internal switches are declared here.

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

`\Gm@cnth` The counters for horizontal and vertical partitioning patterns.

```
\Gm@cntv 20 \newcount\Gm@cnth
21 \newcount\Gm@cntv
```

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

```
22 \newcount\c@Gm@tempcnt
```

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

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

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

```
\Gm@odd@mp 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 area.

```
\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^AT_EX native dimensions can be stored.

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

`\Gm@warning` The macro to print warning messages.

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

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

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

`\Gm@Dhratio` The default values for the horizontal and vertical *marginalratio* are defined. `\Gm@Dhratio`two denotes the default value of horizontal *marginratio* for twoside page layout with left and right margins swapped

`\Gm@Dhratio`two on verso pages, which is set by `twoside`.

```

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

```

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

`\Gm@Dvscale`

```

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

```

`\Gm@dvinfos` The driver names.

`\Gm@dvipdfm`

```

44 \def\Gm@dvinfos{dvips}%

```

`\Gm@pdftex`

```

45 \def\Gm@dvipdfm{dvipdfm}%

```

`\Gm@xetex`

```

46 \def\Gm@pdftex{pdftex}%

```

`\Gm@vtex`

```

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

```

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

`\Gm@false`

```

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

```

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

`\Gm@orgph`

```

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

```

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

```

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

```

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

```

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

```

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

```

62 \def\Gm@restore{}%

```

`\Gm@save` The definition of the macro saving the real lengths L^AT_EX options.

```

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

```

```

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

```

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

```

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

```

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

```

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

```

`\Gm@setdriver` The macro sets the specified driver.

```

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

```

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

```

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

```

`\Gm@setbool` The macros for boolean option processing.

```

\Gm@setboolrev 126 \def\Gm@setbool{\@dblarg\Gm@@setbool}%
127 \def\Gm@setboolrev{\@dblarg\Gm@@setboolrev}%
128 \def\Gm@@setbool[#1]#2#3{\Gm@doif{#1}{#3}{\csname Gm@#2\Gm@bool\endcsname}}%
129 \def\Gm@@setboolrev[#1]#2#3{\Gm@doifelse{#1}{#3}%
130 {\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 true, and executes the fourth argument #4 if false.

```

131 \def\Gm@doif#1#2#3{%
132 \lowercase{\def\Gm@bool{#2}}%

```

```

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

```

`\Gm@reverse` The macro reverses a bool value.

```

150 \def\Gm@reverse#1{%
151   \csname ifGm@#1\endcsname
152   \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.

```

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

```

159 \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.

```

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

234 \ifcase\@tempcnta
235   \if#1h
236     \edef\Gm@width{\Gm@Dhscale\Gm@layoutwidth}%

```

```

237 \else
238 \edef\Gm@height{\Gm@Dvscale\Gm@layoutheight}%
239 \fi
240 \Gm@detiandiii{#2}{#3}{#4}%
Case 001 (=1) with right(bottom) fixed:
241 \or
242 \ifx\Gm@ratio\@undefined
243 \if#1h
244 \edef\Gm@width{\Gm@Dhscale\Gm@layoutwidth}%
245 \else
246 \edef\Gm@height{\Gm@Dvscale\Gm@layoutheight}%
247 \fi
248 \setlength\@tempdimc{\@nameuse{Gm@#4}}%
249 \Gm@detiandiii{#2}{#3}{#4}%
250 \expandafter\let\csname Gm@#2\endcsname\@undefined
251 \expandafter\edef\csname Gm@#4\endcsname{\the\@tempdimc}%
252 \else
253 \Gm@setbyratio[f]{#1}{#4}{#3}%
254 \fi
255 \Gm@detiv{#2}{#3}{#4}{#2}%
Case 010 (=2) with width(height) fixed:
256 \or\Gm@detiandiii{#2}{#3}{#4}%
Case 011 (=3) with both width(height) and right(bottom) fixed:
257 \or\Gm@detiv{#2}{#2}{#4}{#3}%
Case 100 (=4) with left(top) fixed:
258 \or
259 \ifx\Gm@ratio\@undefined
260 \if#1h
261 \edef\Gm@width{\Gm@Dhscale\Gm@layoutwidth}%
262 \else
263 \edef\Gm@height{\Gm@Dvscale\Gm@layoutheight}%
264 \fi
265 \setlength\@tempdimc{\@nameuse{Gm@#3}}%
266 \Gm@detiandiii{#2}{#4}{#3}%
267 \expandafter\let\csname Gm@#2\endcsname\@undefined
268 \expandafter\edef\csname Gm@#3\endcsname{\the\@tempdimc}%
269 \else
270 \Gm@setbyratio[b]{#1}{#3}{#4}%
271 \fi
272 \Gm@detiv{#2}{#3}{#4}{#2}%
Case 101 (=5) with both left(top) and right(bottom) fixed:
273 \or\Gm@detiv{#2}{#3}{#4}{#2}%
Case 110 (=6) with both left(top) and width(height) fixed:
274 \or\Gm@detiv{#2}{#2}{#3}{#4}%
Case 111 (=7) with all fixed though it is over-specified:
275 \or\Gm@warning{Over-specification in '#1'-direction.%
276 ^J\@spaces '#2' (\@nameuse{Gm@#2}) is ignored}%
277 \Gm@detiv{#2}{#3}{#4}{#2}%
278 \else\fi}%

```

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

`\Gm@setlength` This macro stores L^AT_EX native dimensions, which are stored and set afterwards.

```

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

```

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

```

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

```


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

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

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

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

Various paper size are defined here.

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

‘`paper`’ `paper` takes a paper name as its value.

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

‘`a[0-6]paper`’ The following paper names are available.

```
‘b[0-6]paper’ 382 \define@key{Gm}{a0paper}[true]{\Gm@setpaper@ifpre{a0paper}}%
‘b[0-6]j’ 383 \define@key{Gm}{a1paper}[true]{\Gm@setpaper@ifpre{a1paper}}%
‘ansi[a-e]paper’ 384 \define@key{Gm}{a2paper}[true]{\Gm@setpaper@ifpre{a2paper}}%
‘letterpaper’ 385 \define@key{Gm}{a3paper}[true]{\Gm@setpaper@ifpre{a3paper}}%
‘legalpaper’ 386 \define@key{Gm}{a4paper}[true]{\Gm@setpaper@ifpre{a4paper}}%
‘executivepaper’ 387 \define@key{Gm}{a5paper}[true]{\Gm@setpaper@ifpre{a5paper}}%
388 \define@key{Gm}{a6paper}[true]{\Gm@setpaper@ifpre{a6paper}}%
‘screen’ 389 \define@key{Gm}{b0paper}[true]{\Gm@setpaper@ifpre{b0paper}}%
390 \define@key{Gm}{b1paper}[true]{\Gm@setpaper@ifpre{b1paper}}%
391 \define@key{Gm}{b2paper}[true]{\Gm@setpaper@ifpre{b2paper}}%
392 \define@key{Gm}{b3paper}[true]{\Gm@setpaper@ifpre{b3paper}}%
393 \define@key{Gm}{b4paper}[true]{\Gm@setpaper@ifpre{b4paper}}%
394 \define@key{Gm}{b5paper}[true]{\Gm@setpaper@ifpre{b5paper}}%
395 \define@key{Gm}{b6paper}[true]{\Gm@setpaper@ifpre{b6paper}}%
396 \define@key{Gm}{b0j}[true]{\Gm@setpaper@ifpre{b0j}}%
397 \define@key{Gm}{b1j}[true]{\Gm@setpaper@ifpre{b1j}}%
398 \define@key{Gm}{b2j}[true]{\Gm@setpaper@ifpre{b2j}}%
```

```

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

```

‘paperwidth’ Direct specification for paper size is also possible.

```

‘paperheight’ 412 \define@key{Gm}{paperwidth}{\ifGm@preamble{paperwidth}{%
‘papersize’ 413 \Gm@setlength{paperwidth}{#1}}}%
414 \define@key{Gm}{paperheight}{\ifGm@preamble{paperheight}{%
415 \Gm@setlength{paperheight}{#1}}}%
416 \define@key{Gm}{papersize}{\ifGm@preamble{papersize}{%
417 \Gm@branch{#1}{paperwidth}{paperheight}}}%

```

‘layout’ Direct specification for layout size is also possible.

```

‘layoutwidth’ 418 \define@key{Gm}{layout}{\Gm@layouttrue\@nameuse{Gm#1}{Gm@layout}}%
‘layoutheight’ 419 \let\KV@Gm@layoutname\KV@Gm@layout
‘layoutsizes’ 420 \define@key{Gm}{layoutwidth}{\Gm@layouttrue\Gm@setlength\Gm@layoutwidth{#1}}%
421 \define@key{Gm}{layoutheight}{\Gm@layouttrue\Gm@setlength\Gm@layoutheight{#1}}%
422 \define@key{Gm}{layoutsizes}{\Gm@branch{#1}{layoutwidth}{layoutheight}}%

```

‘landscape’ Paper orientation setting.

```

‘portrait’ 423 \define@key{Gm}{landscape}[true]{\ifGm@preamble{landscape}{%
424 \Gm@doifelse{landscape}{#1}}%
425 {\ifGm@landscape\else\Gm@landscapetrue\Gm@reverse{swap@papersize}\fi}%
426 {\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}}%
427 \define@key{Gm}{portrait}[true]{\ifGm@preamble{portrait}{%
428 \Gm@doifelse{portrait}{#1}}%
429 {\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}%
430 {\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’ 431 \define@key{Gm}{hscale}{\Gm@hbodytrue\edef\Gm@hscale{#1}}%
‘scale’ 432 \define@key{Gm}{vscale}{\Gm@vbodytrue\edef\Gm@vscale{#1}}%
433 \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’ 434 \define@key{Gm}{width}{\Gm@hbodytrue\Gm@defbylen{width}{#1}}%
‘totalwidth’ 435 \define@key{Gm}{height}{\Gm@vbodytrue\Gm@defbylen{height}{#1}}%
‘totalheight’ 436 \define@key{Gm}{total}{\Gm@branch{#1}{width}{height}}%
437 \let\KV@Gm@totalwidth\KV@Gm@width
438 \let\KV@Gm@totalheight\KV@Gm@height

```

‘textwidth’ These options directly sets the dimensions *\textwidth* and *\textheight*. *body* is an alias of *text*.

```

‘textheight’ 439 \define@key{Gm}{textwidth}{\Gm@hbodytrue\Gm@defbylen{textwidth}{#1}}%
‘text’ 440 \define@key{Gm}{textheight}{\Gm@vbodytrue\Gm@defbylen{textheight}{#1}}%
‘body’ 441 \define@key{Gm}{text}{\Gm@branch{#1}{textwidth}{textheight}}%
442 \let\KV@Gm@body\KV@Gm@text

```

‘lines’ The option sets *\textheight* with the number of lines.

```

443 \define@key{Gm}{lines}{\Gm@vbodytrue\Gm@defbycnt{lines}{#1}}%

```

‘includehead’ The options take the corresponding dimensions as part of *body*.

```

‘includefoot’ 444 \define@key{Gm}{includehead}[true]{\Gm@setbool{includehead}{#1}}%
‘includeheadfoot’ 445 \define@key{Gm}{includefoot}[true]{\Gm@setbool{includefoot}{#1}}%
‘includemp’
‘includeall’

```

```

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

‘ignorehead’ These options exclude head, foot and marginpars when determining body.
‘ignorefoot’
‘ignoreheadfoot’ 453 \define@key{Gm}{ignorehead}[true]{%
‘ignoreemp’ 454   \Gm@setboolrev[ignorehead]{includehead}{#1}}%
‘ignoreall’ 455 \define@key{Gm}{ignorefoot}[true]{%
456   \Gm@setboolrev[ignorefoot]{includefoot}{#1}}%
457 \define@key{Gm}{ignoreheadfoot}[true]{\Gm@doifelse{ignoreheadfoot}{#1}}%
458   {\Gm@includeheadfalse\Gm@includefootfalse}%
459   {\Gm@includeheadtrue\Gm@includefoottrue}}%
460 \define@key{Gm}{ignoreemp}[true]{%
461   \Gm@setboolrev[ignoreemp]{includemp}{#1}}%
462 \define@key{Gm}{ignoreall}[true]{\Gm@doifelse{ignoreall}{#1}}%
463   {\Gm@includeheadfalse\Gm@includefootfalse\Gm@includempfalse}%
464   {\Gm@includeheadtrue\Gm@includefoottrue\Gm@includemptrue}}%

‘heightrounded’ The option rounds \textheight to n-times of \baselineskip plus \topskip.
465 \define@key{Gm}{heightrounded}[true]{\Gm@setbool{heightrounded}{#1}}%

‘hdivide’ The options are useful to specify partitioning in each direction of the paper.
‘vdivide’
‘divide’ 466 \define@key{Gm}{hdivide}{\Gm@parse@divide{#1}{lmargin}{width}{rmargin}}%
467 \define@key{Gm}{vdivide}{\Gm@parse@divide{#1}{tmargin}{height}{bmargin}}%
468 \define@key{Gm}{divide}{\Gm@parse@divide{#1}{lmargin}{width}{rmargin}}%
469   \Gm@parse@divide{#1}{tmargin}{height}{bmargin}}%

‘lmargin’ These options set margins. left, inner, innermargin are aliases of lmargin. right, outer,
‘rmargin’ outermargin are aliases of rmargin. top and bottom are aliases of tmargin and bmargin respec-
‘tmargin’ tively.
‘bmargin’ 470 \define@key{Gm}{lmargin}{\Gm@defbylen{lmargin}{#1}}%
‘left’ 471 \define@key{Gm}{rmargin}{\Gm@defbylen{rmargin}{#1}}%
‘inner’ 472 \let\KV@Gm@left\KV@Gm@lmargin
‘innermargin’ 473 \let\KV@Gm@inner\KV@Gm@lmargin
‘right’ 474 \let\KV@Gm@innermargin\KV@Gm@lmargin
‘outer’ 475 \let\KV@Gm@right\KV@Gm@rmargin
‘outermargin’ 476 \let\KV@Gm@outer\KV@Gm@rmargin
‘top’ 477 \let\KV@Gm@outermargin\KV@Gm@rmargin
‘bottom’ 478 \define@key{Gm}{tmargin}{\Gm@defbylen{tmargin}{#1}}%
479 \define@key{Gm}{bmargin}{\Gm@defbylen{bmargin}{#1}}%
480 \let\KV@Gm@top\KV@Gm@tmargin
481 \let\KV@Gm@bottom\KV@Gm@bmargin

‘hmargin’ These options are shorthands for setting margins.
‘vmargin’ 482 \define@key{Gm}{hmargin}{\Gm@branch{#1}{lmargin}{rmargin}}%
‘margin’ 483 \define@key{Gm}{vmargin}{\Gm@branch{#1}{tmargin}{bmargin}}%
484 \define@key{Gm}{margin}{\Gm@branch{#1}{lmargin}{tmargin}}%
485   \Gm@branch{#1}{rmargin}{bmargin}}%

‘hmarginratio’ Options specifying the margin ratios.
‘vmarginratio’ 486 \define@key{Gm}{hmarginratio}{\edef\Gm@hmarginratio{#1}}%
‘marginratio’ 487 \define@key{Gm}{vmarginratio}{\edef\Gm@vmarginratio{#1}}%
‘hratio’ 488 \define@key{Gm}{marginratio}{\Gm@branch{#1}{hmarginratio}{vmarginratio}}%
‘vratio’ 489 \let\KV@Gm@hratio\KV@Gm@hmarginratio
‘ratio’ 490 \let\KV@Gm@vratio\KV@Gm@vmarginratio
491 \let\KV@Gm@ratio\KV@Gm@marginratio

‘hcentering’ Useful shorthands to place body centered.
‘vcentering’ 492 \define@key{Gm}{hcentering}[true]{\Gm@doifelse{hcentering}{#1}}%
‘centering’ 493   {\def\Gm@hmarginratio{1:1}}}%

```

```

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

‘twoside’ If twoside=true, \@twoside and \@mparswitch is set to true.
498 \define@key{Gm}{twoside}[true]{\Gm@doifelse{twoside}{#1}%
499   {\@twosidetrue\@mparswitchtrue}{\@twosidefalse\@mparswitchfalse}}%

‘asymmetric’ asymmetric sets \@mparswitchfalse and \@twosidetrue A asymmetric=false has no effect.
500 \define@key{Gm}{asymmetric}[true]{\Gm@doifelse{asymmetric}{#1}%
501   {\@twosidetrue\@mparswitchfalse}{}}%

‘bindingoffset’ The macro adds the specified space to the inner margin.
502 \define@key{Gm}{bindingoffset}{\Gm@setlength\Gm@bindingoffset{#1}}%

‘headheight’ The direct settings of head and/or foot dimensions.
‘headsep’ 503 \define@key{Gm}{headheight}{\Gm@setlength\headheight{#1}}%
‘footskip’ 504 \define@key{Gm}{headsep}{\Gm@setlength\headsep{#1}}%
‘head’ 505 \define@key{Gm}{footskip}{\Gm@setlength\footskip{#1}}%
‘foot’ 506 \let\KV@Gm@head\KV@Gm@headheight
507 \let\KV@Gm@foot\KV@Gm@footskip

‘nohead’ They are only shorthands to set head and/or foot to be 0pt.
‘nofoot’ 508 \define@key{Gm}{nohead}[true]{\Gm@doifelse{nohead}{#1}%
‘noheadfoot’ 509   {\Gm@setlength\headheight\z@\Gm@setlength\headsep\z@}{}}%
510 \define@key{Gm}{nofoot}[true]{\Gm@doifelse{nofoot}{#1}%
511   {\Gm@setlength\footskip\z@}{}}%
512 \define@key{Gm}{noheadfoot}[true]{\Gm@doifelse{noheadfoot}{#1}%
513   {\Gm@setlength\headheight\z@\Gm@setlength\headsep
514     \z@\Gm@setlength\footskip\z@}{}}%

‘footnotesep’ The option directly sets a native dimension \footnotesep.
515 \define@key{Gm}{footnotesep}{\Gm@setlength{\skip\footins}{#1}}%

‘marginparwidth’ They directly set native dimensions \marginparwidth and \marginparsep.
‘marginpar’ 516 \define@key{Gm}{marginparwidth}{\Gm@setlength\marginparwidth{#1}}%
‘marginparsep’ 517 \let\KV@Gm@marginpar\KV@Gm@marginparwidth
518 \define@key{Gm}{marginparsep}{\Gm@setlength\marginparsep{#1}}%

‘nomarginpar’ The macro is a shorthand for \marginparwidth=0pt and \marginparsep=0pt.
519 \define@key{Gm}{nomarginpar}[true]{\Gm@doifelse{nomarginpar}{#1}%
520   {\Gm@setlength\marginparwidth\z@\Gm@setlength\marginparsep\z@}{}}%

‘columnsep’ The option sets a native dimension \columnsep.
521 \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’ 522 \define@key{Gm}{hoffset}{\Gm@setlength\hoffset{#1}}%
523 \define@key{Gm}{voffset}{\Gm@setlength\voffset{#1}}%
524 \define@key{Gm}{offset}{\Gm@branch{#1}{hoffset}{voffset}}%

‘layoutoffset’
‘layoutvoffset’ 525 \define@key{Gm}{layoutoffset}{\Gm@setlength\Gm@layoutoffset{#1}}%
‘layoutoffset’ 526 \define@key{Gm}{layoutvoffset}{\Gm@setlength\Gm@layoutvoffset{#1}}%
527 \define@key{Gm}{layoutoffset}{\Gm@branch{#1}{layoutoffset}{layoutvoffset}}%

‘twocolumn’ The option sets \twocolumn switch.
528 \define@key{Gm}{twocolumn}[true]{%
529   \Gm@doif{twocolumn}{#1}{\csname @twocolumn\Gm@bool\endcsname}}%

```

‘onecolumn’ This option has the reverse effect of twocolumn option.

```

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

‘reversemp’ The both options set \reversemargin.

‘reversemarginpar’

```

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

‘dvriver’

```

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

```

542 \define@key{Gm}{dvips}[true]{\ifGm@preamble{dvips}{%
543 \Gm@doifelse{dvips}{#1}{\Gm@setdriver{dvips}}{\Gm@unsetdriver{dvips}}}}%
```

‘pdftex’

```

544 \define@key{Gm}{dvipdfm}[true]{\ifGm@preamble{dvipdfm}{%
545 \Gm@doifelse{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{\Gm@unsetdriver{dvipdfm}}}}%
```

‘xetex’

```

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

‘vtex’

```

550 \define@key{Gm}{vtex}[true]{\ifGm@preamble{vtex}{%
551 \Gm@doifelse{vtex}{#1}{\Gm@setdriver{vtex}}{\Gm@unsetdriver{vtex}}}}%
```

‘verbose’ The verbose mode.

```

552 \define@key{Gm}{verbose}[true]{\ifGm@preamble{verbose}{\Gm@setbool{verbose}{#1}}}%
553 \define@key{Gm}{reset}[true]{\ifGm@preamble{reset}{%
554 \Gm@doifelse{reset}{#1}{\Gm@restore@org\Gm@initall
555 \ProcessOptionsKV[c]{Gm}\Gm@setdefaultpaper}{}}}%
556 \define@key{Gm}{resetpaper}[true]{\ifGm@preamble{resetpaper}{%
557 \Gm@setbool{resetpaper}{#1}}}%
558 \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.

```

559 \define@key{Gm}{truedimen}[true]{\ifGm@preamble{truedimen}{%
560 \Gm@doifelse{truedimen}{#1}{\let\Gm@truedimen\Gm@true}%
561 {\let\Gm@truedimen\empty}}}%
562 \define@key{Gm}{pass}[true]{\ifGm@preamble{pass}{\Gm@setbool{pass}{#1}}}%
563 \define@key{Gm}{showframe}[true]{\Gm@setbool{showframe}{#1}}%
564 \define@key{Gm}{showcrop}[true]{\Gm@setbool{showcrop}{#1}}%
```

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

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

‘showcrop’ The showcrop option prints cropmarks at the corners.

`\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.

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

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

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

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

```
621 \fi}%
```

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

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

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

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

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

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

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

```

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

```

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

```

696 \def\Gm@process{%
    If pass is set, the original dimensions and switches are restored and process is ended here.
697   \ifGm@pass
698     \Gm@restore@org
699   \else
700     \Gm@@process
701   \fi}%

```

The main processing macro.

```

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

```

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

```

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

```

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

```

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

```

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

```


The bindingoffset correction for \oddsidemargin.

```
728 \advance\oddsidemargin\Gm@bindingoffset
729 \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.

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

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

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

If heightrounded is set, \textheight is rounded.

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

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

```
756 \advance\oddsidemargin\Gm@layouthoffset%
757 \advance\evensidemargin\Gm@layouthoffset%
758 \advance\topmargin\Gm@layoutvoffset%
759 \addtolength\Gm@layoutwidth{\Gm@bindingoffset}%
760 }% 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.

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

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

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

```
769 \ifvtex
770 \Gm@setdriver{vtex}%
771 \fi
xetex
772 \@ifundefined{XeTeXversion}{}{\Gm@setdriver{xetex}}%
```

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

Macros for the page frames and cropmarks.

```

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

```

```

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

```

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

```

944 \def\Gm@processconf{%
945 \let\Gm@origExecuteOptions\ExecuteOptions
946 \let\ExecuteOptions\Gm@setkeys
947 \InputIfFileExists{geometry.cfg}{-}{-}
948 \let\ExecuteOptions\Gm@origExecuteOptions}%

```

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

```

949 \Gm@save

```

```

950 \edef\Gm@restore@org{\Gm@restore}%
951 \Gm@initall
    Processing config file.
952 \Gm@processconfig
    The optional arguments to \documentclass are processed here.
953 \ProcessOptionsKV[c]{Gm}%
    Paper dimensions given by class default are stored.
954 \Gm@setdefaultpaper
    The optional arguments to \usepackage are processed here.
955 \ProcessOptionsKV[p]{Gm}%
    Actual settings and calculation for layout dimensions are processed.
956 \Gm@process

```

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

```

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

```

The original paper size is used if `resetpaper`.

```

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

```

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

```

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

```

Checking the driver options.

```

977   \Gm@detectdriver

```

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

```

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

```

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

```

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

```

```

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

```

With $\text{V}\text{T}_{\text{E}}\text{X}$ environment, $\text{V}\text{T}_{\text{E}}\text{X}$ variables are set here.

```

1006 \ifx\Gm@driver\Gm@vtex
1007 \@ifundefined{mediawidth}{}{%
1008 \mediawidth=\paperwidth
1009 \mediaheight=\paperheight}%
1010 \ifvtexdvi
1011 \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1012 \fi
1013 \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.

```

1014 \ifx\Gm@driver\Gm@dvips
1015 \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1016 \ifx\Gm@driver\Gm@dvips\ifGm@landscape
1017 \AtBeginDvi{\special{! /landplus90 true store}}%
1018 \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.

```

1019 \else\ifx\Gm@driver\Gm@dvipdfm
1020 \ifcase\ifx\AtBeginShipoutFirst\relax\@ne\else
1021 \ifx\AtBeginShipoutFirst\@undefined\@ne\else\z@\fi\fi
1022 \AtBeginShipoutFirst{\special{papersize=\the\paperwidth,\the\paperheight}}%
1023 \or
1024 \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1025 \fi
1026 \fi\fi

```

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

```

1027 \ifGm@showframe
1028 \renewcommand*{\shipout}{\afterassignment\gm@shipi\setbox\@cclv=}
1029 \else\ifGm@showcrop
1030 \renewcommand*{\shipout}{\afterassignment\gm@shipi\setbox\@cclv=}
1031 \fi\fi

```

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

```

1032 \Gm@save
1033 \edef\Gm@restore@pkg{\Gm@restore}%

```

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

```

1034 \ifGm@verbose\ifGm@pass\else\Gm@checkmp\fi\fi

```

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

```

1035 \Gm@showparams{preamble}%

```

The following lines free the memories no longer needed.

```

1036 \let\Gm@pw\relax
1037 \let\Gm@ph\relax
1038 }% end of \AtBeginDocument

```

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

```

1039 \newcommand{\geometry}[1]{%
1040   \Gm@clean
1041   \setkeys{Gm}{#1}%
1042   \Gm@process}%
1043 \@onlypreamble\geometry

```

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

```

1044 \DeclareRobustCommand\Gm@changelayout{%
1045   \setlength{\@colht}{\textheight}
1046   \setlength{\@colroom}{\textheight}%
1047   \setlength{\vsize}{\textheight}
1048   \setlength{\columnwidth}{\textwidth}%
1049   \if@twocolumn%
1050     \advance\columnwidth-\columnsep
1051     \divide\columnwidth\tw@%
1052     \@firstcolumntrue%
1053   \fi%
1054   \setlength{\hsize}{\columnwidth}%
1055   \setlength{\linewidth}{\hsize}}%

```

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

```

1056 \newcommand\newgeometry[1]{%
1057   \clearpage
1058   \Gm@restore@org
1059   \Gm@initnewgm
1060   \Gm@newgmtrue
1061   \setkeys{Gm}{#1}%
1062   \Gm@newgmfalse
1063   \Gm@process
1064   \ifnum\mag=@\else\Gm@magtooffset\fi
1065   \Gm@changelayout
1066   \Gm@showparams{newgeometry}}%

```

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

```

1067 \newcommand\restoregeometry{%
1068   \clearpage
1069   \Gm@restore@pkg
1070   \Gm@changelayout}%

```

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

```

1071 \newcommand*\savegeometry[1]{%
1072   \Gm@save
1073   \expandafter\edef\csname Gm@restore@@#1\endcsname{\Gm@restore}}%

```

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

```

1074 \newcommand*\loadgeometry[1]{%
1075   \clearpage
1076   \@ifundefined{Gm@restore@@#1}{%
1077     \PackageError{geometry}{%
1078       \string\loadgeometry : name ‘#1’ undefined}{%
1079       The name ‘#1’ should be predefined with \string\savegeometry}%
1080   }{\@nameuse{Gm@restore@@#1}}%
1081   \Gm@changelayout}}%
1082 \end{package}

```

12 Config file

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

```
1083 (*config)
1084 %<<SAVE_INTACT
1085
1086 % Uncomment and edit the line below to set default options.
1087 \ExecuteOptions{a4paper}
1088
1089 %SAVE_INTACT
1090 </config>
```

13 Sample file

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

```
1091 (*samples)
1092 %<<SAVE_INTACT
1093 \documentclass[12pt]{article}% uses letterpaper by default
1094 % \documentclass[12pt,a4paper]{article}% for A4 paper
1095 %-----
1096 % Edit and uncomment one of the settings below
1097 %-----
1098 % \usepackage{geometry}
1099 % \usepackage[centering]{geometry}
1100 % \usepackage[width=10cm,vscale=.7]{geometry}
1101 % \usepackage[margin=1cm, papersize={12cm,19cm}, resetpaper]{geometry}
1102 % \usepackage[margin=1cm,includeheadfoot]{geometry}
1103 % \usepackage[margin=1cm,includeheadfoot,includemp]{geometry}
1104 % \usepackage[margin=1cm,bindingoffset=1cm,twoside]{geometry}
1105 % \usepackage[hmarginratio=2:1, vmargin=2cm]{geometry}
1106 % \usepackage[hscale=0.5,twoside]{geometry}
1107 % \usepackage[hscale=0.5,asymmetric]{geometry}
1108 % \usepackage[hscale=0.5,heightrounded]{geometry}
1109 % \usepackage[left=1cm,right=4cm,top=2cm,includefoot]{geometry}
1110 % \usepackage[lines=20,left=2cm,right=6cm,top=2cm,twoside]{geometry}
1111 % \usepackage[width=15cm, marginparwidth=3cm, includemp]{geometry}
1112 % \usepackage[hdivide={1cm,,2cm}, vdivide={3cm,8in,}, nohead]{geometry}
1113 % \usepackage[headsep=20pt, head=40pt,foot=20pt,includeheadfoot]{geometry}
1114 % \usepackage[text={6in,8in}, top=2cm, left=2cm]{geometry}
1115 % \usepackage[centering,includemp,twoside,landscape]{geometry}
1116 % \usepackage[mag=1414,margin=2cm]{geometry}
1117 % \usepackage[mag=1414,margin=2truecm,truedimen]{geometry}
1118 % \usepackage[a5paper, landscape, twocolumn, twoside,
1119 % left=2cm, hmarginratio=2:1, includemp, marginparwidth=43pt,
1120 % bottom=1cm, foot=.7cm, includefoot, textheight=11cm, heightrounded,
1121 % columnsep=1cm,verbose]{geometry}
1122 %-----
1123 % No need to change below
1124 %-----
1125 \geometry{verbose,showframe}% the options appended.
1126 \usepackage{lipsum}% for dummy text of 150 paragraphs
1127 \newcommand\mynote{\marginpar[\raggedright
1128 A sample margin note in the left side.]}
1129 {\raggedright A sample margin note.}}%
1130 \newcommand\myfootnote{\footnote{This is a sample footnote text.}}
1131 \begin{document}
1132 \lipsum[1-2]\mynote\lipsum[3-4]\mynote
1133 \lipsum[5-11]\mynote\lipsum[12]\myfootnote
1134 \lipsum[13-22]\mynote\lipsum[23-32]
1135 \end{document}
1136 %SAVE_INTACT
1137 </samples>
```