The geometry package

Hideo Umeki
latexgeometry@gmail.com

2008/12/21 v4.2

Abstract

This package provides a flexible and easy interface to page dimensions. You can set 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}`.

Contents

1 Preface to version 4 1

2 Preface to version 3 2

3 Introduction 3

4 Page geometry 4
- 4.1 Layout dimensions 4
- 4.2 Auto-completion scheme 5

5 User interface 7
- 5.1 General features 7
- 5.2 Option types 8

6 Option specification 9
- 6.1 Paper size 9
- 6.2 Body size 9
- 6.3 Margin size 11
- 6.4 Native dimensions 11
- 6.5 drivers 13
- 6.6 Other options 13

7 Default settings 14
- 7.1 Default layout 14
- 7.2 Configuration file 14

8 Relations between options 14
- 8.1 Order dependence 14
- 8.2 Priority 15

9 Examples 15

10 Known problems 17

11 Acknowledgments 17

12 Implementation 17

13 Config file 37

14 Sample file 37

1 Preface to version 4

Many improvements to the code and documentation were made according to suggestions and comments from users. Main changes are listed below.

- **More robust driver detection.**

  The driver detection method has been totally rewritten so that it can automatically detect the driver appropriate for the typesetting program in use. Therefore, explicit driver setting is no longer needed in most cases, except for the driver `dvipdfm`. This improvement makes geometry work more robustly for typesetting programs under eTEX, XeTEX and VTEX as well as normal TEX environment. The packages ifpdf and ifvtex are used, which are available in CTAN. See Section 6.5 for details. Note that ifvtex package v1.3 (2007/09/09) had a bug (a typo) that made the detection of VTeX wrong. So make sure ifvtex v1.4 or later is being used.

- **New option: `resetpaper`.**

  This option disables explicit paper setting in geometry and uses the paper size specified before geometry. This option may be useful to print nonstandard sized documents with normal printers and papers.

- **Added adjustment to `topskip`.**

  When `lines` option and large font sizes are specified, `\topskip` can be adjusted so that the formula "`\textheight` = (*lines* − 1) × `\baselineskip` + `\topskip`" to be correct. To do this, `\topskip` is set to `\ht\strutbox`, if `\topskip` is smaller than `\ht\strutbox`.

- **Added ANSI paper sizes.**

  New paper size definitions for ANSI A to E are added.

- **Fixed wrong ISO paper sizes.**

  The paper sizes for A1,A2,A5 and A6 were wrong (by 1mm).

- **Fixed pdfTeX magnification problem.**

  PDF paper offset is adjusted properly when magnification is set by `mag` option with pdfTeX.

- **Changed package source organization.**

  Files `geometry.ins` and `geometry-samples.tex` as well as `geometry.sty` are integrated into `geometry.dtx` so that they can be generated from `geometry.dtx` by 'tex' command. Documentation can be also generated directly from `geometry.dtx` by '(pdf)latex' command.

2 Preface to version 3

The geometry package becomes even more flexible and powerful with the release of version 3. This new release contains major changes and enhancements in user interface, calculation schemes and the default settings of the page dimensions.

- **New default layout.**

  The 'automatic' centering is no longer default layout. Instead of centering, the idea of margin ratio and common values for default settings are introduced: the ratio of left (inner) margin to right (outer) margin is set 1:1 (2:3 for twoside), and the ratio of top to bottom is set 2:3. The margin ratios can be specified by newly introduced options, e.g. `marginratio` (see Section 4.2 and 6.3 for the detail). In addition, the spaces for the head and foot of the page are disregarded in calculating the placement of the text area by default. Furthermore the default `scale` of the type area is set to `0.7` with 70% of the width and height of the paper. If you want to use the old default layout of version 2.3 or earlier, add `compat2` as a first option, e.g., `\usepackage[compat2,left=1.5in]{geometry}`, which sets the old default options `[scale={0.8,0.9}, centering, includeheadfoot]` and allows the subsequent options to behave as if they are used in the old version. See also Section 7.1 for the detail of the default layout.

- **Option `twosideshift` is obsoleted.**

  `twoside` and other geometry options can substitute for it. A new option `bindingoffset` might be also helpful to control margins for oneside/twoside. For the detail, see Section 6.3.

- **Option `includemp` becomes independent of `marginparwidth` and `marginparsep`.**

  In the previous version, `marginparwidth` or `marginparsep` automatically set `includemp=true`. Now if you want `includemp` mode, `includemp` should be set explicitly.

- **Options `nohead`, `nofoot` and `noheadfoot` become order-dependent and overwritable**

  In the previous version, these options was order-independent: `nohead,headsep=10pt` resulted in just `nohead` (`\headsep=0pt`, `\headheight=0pt`), for example. But now they are overwritable by subsequent options. The above case results in `\headheight=0pt` and `\headsep=10pt`.

- **A complete set of options `ignore*` and `include*` for head, foot and marginpar.**

  The previous version has only `includemp`, which denotes that the width of marginpar is included in the total body width. Now `ignore{head, foot, headfoot, mp, all}` and `include{head, foot, headfoot, all}` are newly added. If one of these `ignore*` is set, the corresponding space(s) are disregarded in auto-completion calculation. In version 3, `ignoreall` is set by default. So if you need to include the spaces for the head, foot and marginpar, the corresponding `include*` should be set explicitly. In addition, unlike the previous version, neither `reversemp`, `marginparwidth` nor `marginparsep` sets `includemp` automatically.

- **New option `lines`.**

  The option enables users to specify `\textheight` by the number of lines included in `\textheight`, e.g., `lines=20`.

- **New option `heightrounded`.**

  The option rounds `\textheight` to n-times (n: an integer) of `\baselineskip` plus `\topskip` to avoid "underfull vbox" in some cases.

- **New option `screen`.**

  To make presentation with PC and video projector, geometry option `screen,centering` with 'slide' documentclass would be the best choice.

- **New option `asymmetric`.**

  The option implements a twosided layout in which margins are not swapped on alternate pages and the marginal notes stay always on the same side.

- **New option `showframe`.**

  The option displays visible frames for the text area and page, and lines for the head and foot to check layout in detail. Therefore `showframe.sty` is excluded from the geometry package distribution.

- **New option `pass`.**

  The option disables auto-layout and all of the geometry settings except `verbose` and `showframe`. It can be used for checking out the page layout of the documentclass, other packages and manual settings without geometry.

See the text for the detail. All the new and modified options in this release are marked with '⋆3' and '†3' respectively.

3 Introduction

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

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

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

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

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

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

In both cases, the unspecified dimensions are automatically determined. The 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

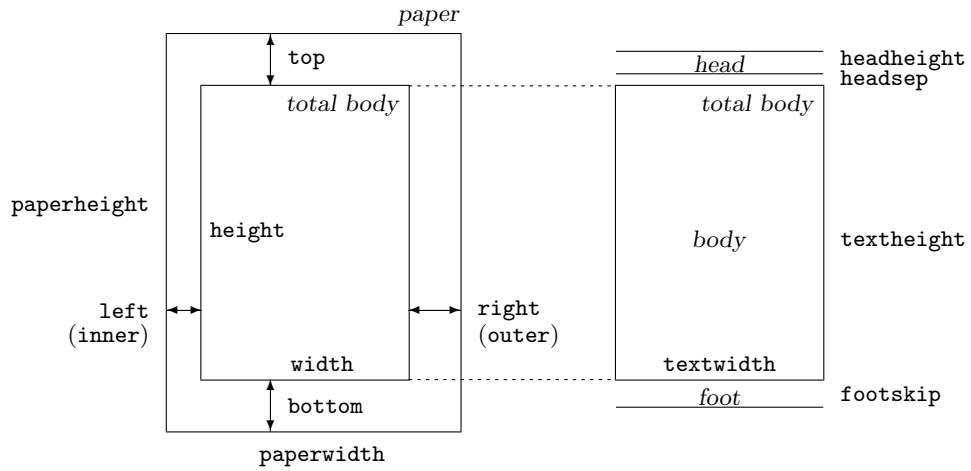

Figure 1: Dimension names used in the geometry package. `width`=`textwidth` and `height`=`textheight` by default. `left`, `right`, `top` and `bottom` are margins. If margins on verso pages are swapped by `twoside` option, margins specified by `left` and `right` options are used for the inside and outside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

```
\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.

4 Page geometry

4.1 Layout dimensions

To realize a straightforward setting for page layout, the following page structure is introduced: A paper contains a total body (printable area) and margins. The total body consists of a body (text area) with optional a header, a footer and marginal notes (marginpar). There are four margins: the left, right, top and bottom margins. For twosided documents, horizontal margins should be called the inner and outer margins.

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 layout parts and the corresponding dimension names used in this package are showed schematically in Figure 1. The dimensions for paper, total body and margins have the following relations.

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

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

The dimensions of the total body, `width` and `height`, are defined as follows:

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

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

In Equation (3), `width:=textwidth` by default, but `marginparsep` and `marginparwidth` are included in `width` if `includemp` option is set `true`. In Equation (4), `height:=textheight` by default. If

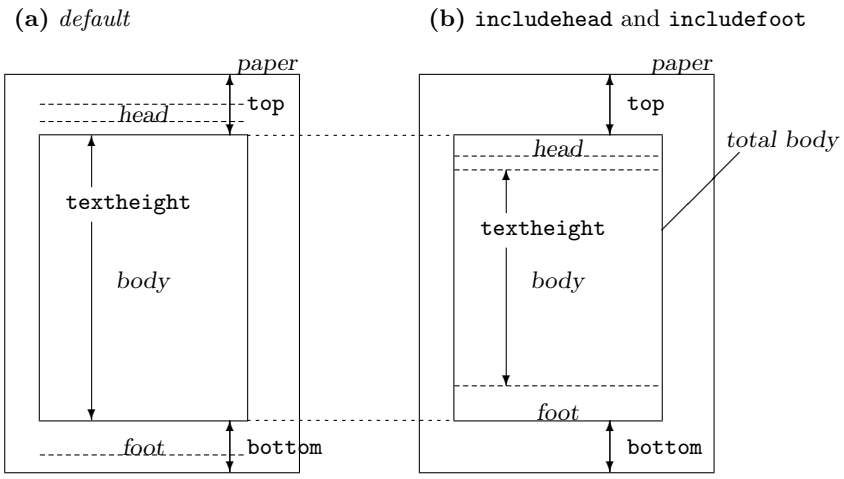

Figure 2: `includehead` and `includefoot` include the head and foot respectively into *total body*. **(a)** `height` = `textheight` (default). **(b)** `height` = `textheight` + `headheight` + `headsep` + `footskip` if `includehead` and `includefoot`. If the top and bottom margins are fixed, `includehead`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

`includehead` is set to `true`, `headheight` and `headsep` are considered as a part of `height` in the the vertical completion calculation. In the same way, `includefoot` includes `footskip`. Note that options `ignore*` just exclude the corresponding spaces from `textheight`, but do not change those lengths themselves. Figure 2 shows how these options work. Each of the seven dimensions in the right-hand side of Equations (3) and (4) corresponds to the ordinary LaTeX control sequence with the same name.

Figure 3 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 `ignore*` do *not* change the corresponding native dimensions.

4.2 Auto-completion scheme

Suppose that the paper size is pre-defined in Equation (1) or (2), if two dimensions out of the three dimensions in the right-hand side of each equation are specified, the rest of the dimensions can be determined by the specified ones. However, when none or only one of the three dimensions is specified, the rest of the dimensions can't generally be determined without some assumptions.

The geometry package has an auto-completion scheme with some default parameters to determine the unspecified dimensions independently for each direction. If the size of *total body* (i.e., `width` in the horizontal direction) is specified, the margins (`left` and `right`) can be determined with a default ratio of one margin to the other (`left/right`). If one margin is specified, the rest of dimensions can also be determined by the default margin ratio. Page margin setting by margin ratio was introduced in KOMA script[1].

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

$$\texttt{top} : \texttt{bottom} = 2 : 3 \qquad \textit{default.} \tag{5}$$

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

$$\texttt{left (inner)} : \texttt{right (outer)} = \begin{cases} 1 : 1 & \textit{default for oneside,} \\ 2 : 3 & \textit{default for twoside.} \end{cases} \tag{6}$$

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

For example, if one specifies `right=2.4cm` with a *twosided* layout in A4 paper (21.0cm×29.7cm), unspecified `left` and `width` are automatically determined using the default horizontal margin ratio (2/3) as follows:

$$\begin{aligned} \texttt{left} &= \langle \text{horizontal-margin-ratio} \rangle \times \texttt{right} \\ &= \texttt{2/3} \times \texttt{2.4cm} = \texttt{1.6cm} \end{aligned} \tag{7}$$

[1] CTAN: `macros/latex/contrib/koma-script` by Frank Neukam and Markus Kohm.

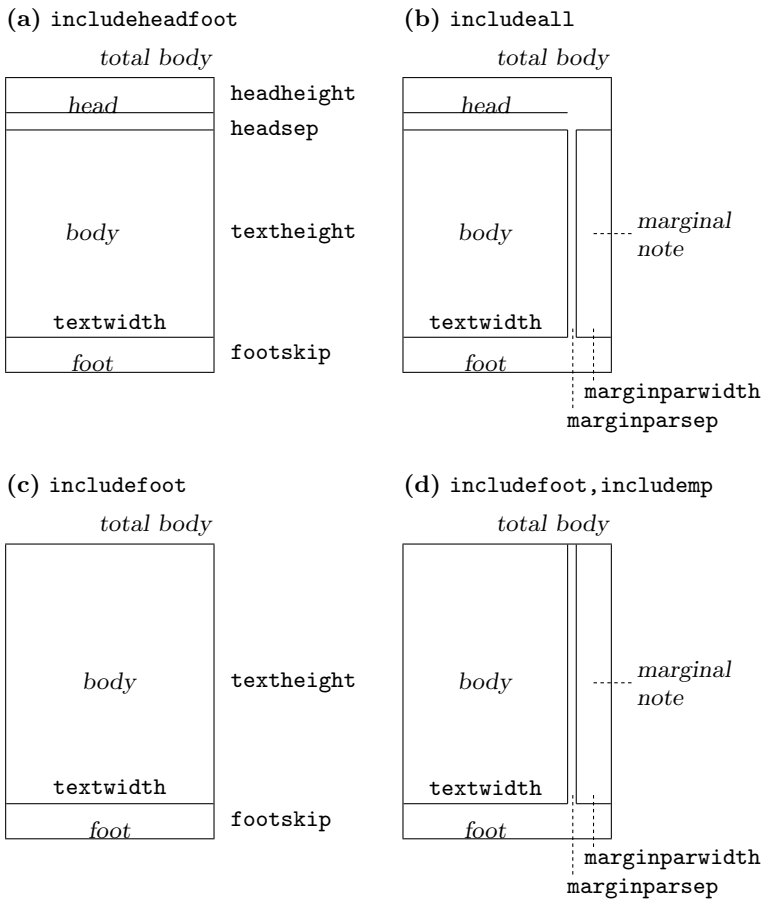

Figure 3: Sample layouts for *total body* with different switches. (a) `includeheadfoot`, (b) `includeall`, (c) `includefoot` and (d) `includefoot,includemp`. If `reversemp` is set to `true`, the location of the marginal notes are swapped on every page. Option `twoside` swaps both margins and marginal notes on verso pages. Note that the marginal notes are printed on the page, even when `ignoremp` or `includemp=false`, but can fall off the page in some cases.

Settings				Results		
left	width	right		left	width	right
top	height	bottom		top	height	bottom
$*$	$*$	$*$		$\sigma\mathcal{M}(0.7L)$	$0.7L$	$\mathcal{M}(0.7L)$
$*$	A	$*$		$\sigma\mathcal{M}(A)$	A	$\mathcal{M}(A)$
A	$*$	$*$		A	$\mathcal{R}(A+A/\sigma)$	A/σ
$*$	$*$	A	$\Longrightarrow$	σA	$\mathcal{R}(A+\sigma A)$	A
A	B	$*$		A	B	$\mathcal{R}(A+B)$
$*$	A	B		$\mathcal{R}(A+B)$	A	B
A	$*$	B		A	$\mathcal{R}(A+B)$	B
A	C	B		A	$\mathcal{R}(A+B)$	B

Table 1: Auto-completion rules. The mark '$*$' in each row (left table) denotes the dimensions not specified explicitly, which can be determined as the corresponding Results (right table). σ denotes the value of margin ratio. Functions $\mathcal{R}(x)$ and $\mathcal{M}(x)$ are defined in Equation (12). The bottom case shows over-specification, which gives in the same result as the A-$*$-B case.

$$\begin{aligned} \texttt{width} &= \texttt{paperwidth} - \texttt{left} - \texttt{right} \\ &= \texttt{21.0cm} - \texttt{1.6cm} - \texttt{2.4cm} = \texttt{17.0cm}. \end{aligned} \tag{8}$$

In this case, the vertical dimensions `top`, `height` and `bottom` are determined by the default vertical margin ratio with 2:3 and the default size of *total body* with 70% of the paper height:

$$\begin{aligned} \texttt{height} &= 0.7 \times \texttt{paperheight} \\ &= \texttt{0.7} \times \texttt{29.7cm} = \texttt{20.79cm} \end{aligned} \tag{9}$$

$$\begin{aligned} \texttt{top} &= \frac{\langle\text{vertical-margin-ratio}\rangle}{1+\langle\text{vertical-margin-ratio}\rangle} \times (\texttt{paperheight} - \texttt{height}) \\ &= \frac{2}{2+3} \times (\texttt{29.7cm} - \texttt{20.79cm}) \\ &= 0.4 \times \texttt{8.91cm} = \texttt{3.564cm} \end{aligned} \tag{10}$$

$$\texttt{bottom} = 0.6 \times \texttt{8.91cm} = \texttt{5.346cm} \tag{11}$$

The auto-completion rules are shown in Table 1 and Equation (12). A, B and C in Table 1 are user-specified values, $*$ denotes unspecified ones. The right-hand side table shows the corresponding results of auto-completion. The unspecified values can be determined by A, B and L (`paperwidth` or `paperheight`). In Table 1, functions $\mathcal{R}(x)$ and $\mathcal{M}(x)$ are defined as follows:

$$\begin{aligned} \mathcal{R}(x) &= L - x \\ \mathcal{M}(x) &= \mathcal{R}(x) \;/\; (1+\sigma) \end{aligned} \tag{12}$$

Here σ denotes the ratio of left margin (inner) to right margin (outer) or the ratio of top to bottom. To set σ as a geometry option, you can use `{h,v}marginratio` options with `a:b`-type value, for example, `hmarginratio=2:3`.

$$\texttt{hmarginratio} = \texttt{left} : \texttt{right} \tag{13}$$

$$\texttt{vmarginratio} = \texttt{top} : \texttt{bottom} \tag{14}$$

By default, σ is 1/1 (=1) for oneside and 2/3 for twoside in the horizontal direction, and 2/3 in the vertical. If none of three dimensions is specified in each direction, the default setting is used: width and height is set to 70% of the paper width and height respectively. If all the three dimensions would be specified, margins remain and width or height is ignored.

5 User interface

5.1 General features

The geometry options using the keyval interface '⟨*key*⟩=⟨*value*⟩' can be set either in the optional argument to the `\usepackage` command, or in the argument of the `\geometry` macro. This macro, if

necessary, should be used only in the preamble, i.e., before `\begin{document}`. In either case, the argument consists of a list of comma-separated keyval options. The main features of setting options are listed below.

- Multiple lines are allowed. (But blank lines are not allowed.)
- Any spaces between words are ignored.
- Options are basically order-independent.
  (There are some exceptions. See Section 8.1 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*[2]. For example,

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

5.2 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.

   ⟨*key*⟩`=true` | `false`.
   ⟨*key*⟩ with no value is equivalent to ⟨*key*⟩`=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.

   ⟨*key*⟩`=`⟨*value*⟩.

   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.

   ⟨*key*⟩`={`⟨*value1*⟩`,`⟨*value2*⟩`}`.
   ⟨*key*⟩`=`⟨*value*⟩ is equivalent to ⟨*key*⟩`={`⟨*value*⟩`,`⟨*value*⟩`}`.

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

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

4. **Triple-valued type**

   takes three mandatory, comma-separated values in braces.

   ⟨*key*⟩`={`⟨*value1*⟩`,`⟨*value2*⟩`,`⟨*value3*⟩`}`

   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}`.

6 Option specification

This section describes all the options provided by geometry.

6.1 Paper size

The options below set paper/media size and orientation.

`paper` | `papername`
: specifies a paper name. The paper names available in geometry. `paper=`⟨*paper-name*⟩. For example `paper=a4paper`, which is equivalent to just `a4paper`.

`a0paper, a1paper, a2paper, a3paper, a4paper, a5paper, a6paper`
`b0paper, b1paper, b2paper, b3paper, b4paper, b5paper, b6paper`
`ansiapaper, ansibpaper, ansicpaper, ansidpaper, ansiepaper`
`letterpaper, executivepaper, legalpaper`
: specifies paper name. They can typically be used with no values. Note that whatever value (even `false`) is given to this option, the value will be ignored. For example, the followings have the same effect: `a5paper`, `a5paper=true`, `a5paper=false` and `a5paper=XXXX`.

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

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

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

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

`landscape`
: switches the paper orientation to landscape mode.

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

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 6.4).

6.2 Body size

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

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

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

`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 should not be confused with `textwidth`. Generally, `width` ≥ `textwidth` because `width` includes the width of the marginal notes if `includemp` is set to `true`. If `textwidth` and `width` are specified at the same time, `width` is ignored.

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

`textheight`
: modifies `\textheight`, the height of *body*. `textheight=`⟨*length*⟩.

`text` | `body`
: sets 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.

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

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

`includemp`
: includes the margin notes, `\marginparwidth` and `\marginparsep`, into *body* when calculating horizontal calculation. In version 3, `includemp` is independent of options `marginparwidth` and `marginparsep`, and set to `false` by default.

`includeall`
: sets both `includeheadfoot` and `includemp` to `true`. See Figure 2 and Figure 3.

`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 is set to `true` by default. See also `includefoot`.

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

`ignoremp`
: disregards the marginal notes in determining the horizontal margins (`true` is set by default). If marginal notes fall off the page, the warning message will be displayed when `verbose=true`. See also Figure 3 and `includemp`.

`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.

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

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

`vdivide`
: vertical partitions (top,height,bottom). `vdivide={`⟨*top margin*⟩`,`⟨*height*⟩`,`⟨*bottom margin*⟩`}`.

`divide`
: `divide={`A`,`B`,`C`}` is interpreted as `hdivide={`A`,`B`,`C`}` and `vdivide={`A`,`B`,`C`}`.

6.3 Margin size

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`bindingoffset`
: removes a specified space from the lefthand-side of the page for oneside or the inner-side for twoside. `bindingoffset=`⟨*length*⟩. This is useful if pages are bound by a press binding (glued, stitched, stapled ...). See Figure 4.

`hdivide`
: See description in Section 6.2.

`vdivide`
: See description in Section 6.2.

`divide`
: See description in Section 6.2.

6.4 Native dimensions

The options below specify LaTeX native dimensions and switches for page layout. See Figure 1. Note that unlike version 2.3, `nohead`, `nofoot` and `noheadfoot` become overwritable, in other words, just shorthand for setting the corresponding LaTeX dimensions (`\headheight`, `\headsep` and `\footskip`) to 0pt.

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

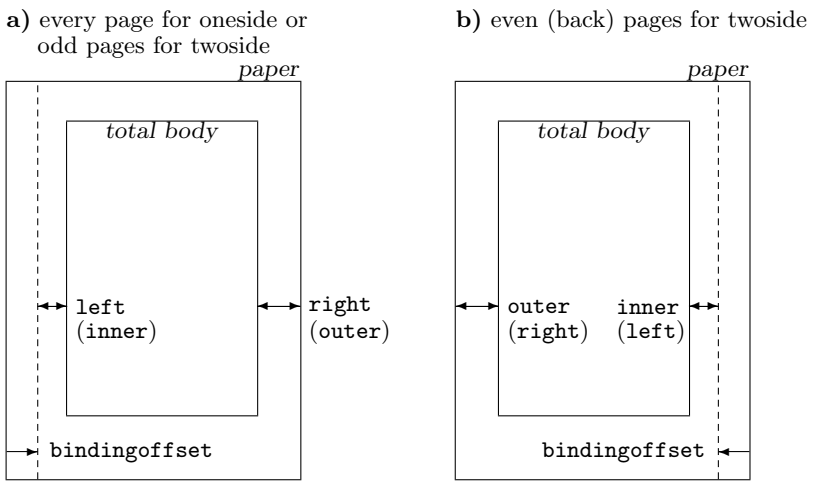

Figure 4: `bindingoffset` option. Note that `twoside` option swaps the horizontal margins and the marginal notes together with `bindingoffset` on even pages (see **b)**), but `asymmetric` option suppresses the swap of the margins and marginal notes (but `bindingoffset` is still swapped).

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

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

`nohead` eliminates spaces for the head of the page, which is equivalent to both `\headheight=0pt` and `\headsep=0pt`.

`nofoot` eliminates spaces for the foot of the page, which is equivalent to `\footskip=0pt`.

`noheadfoot` equivalent to `nohead` and `nofoot`, which means that `\headheight`, `\headsep` and `\footskip` are all set to `0pt`.

`footnotesep` changes the dimension `\skip\footins`, separation between the bottom of text body and the top of footnote text.

`marginparwidth` | `marginpar` modifies `\marginparwidth`, width of the marginal notes. `marginparwidth=`⟨*length*⟩. Unlike version 2.3, it does *not* set `includemp=true`.

`marginparsep` modifies `\marginparsep`, separation between body and marginal notes. `marginparsep=`⟨*length*⟩. Unlike version 2.3, it does *not* set `includemp=true`.

`nomarginpar` shrinks spaces for marginal notes to 0pt, which is equivalent to `\marginparwidth=0pt` and `\marginparsep=0pt`.

`columnsep` modifies `\columnsep`, the separation between two columns in `twocolumn` mode.

`hoffset` modifies `\hoffset`. `hoffset=`⟨*length*⟩.

`voffset` modifies `\voffset`. `voffset=`⟨*length*⟩.

`offset` horizontal and vertical offset. `offset={`⟨*hoffset*⟩`,`⟨*voffset*⟩`}` or `offset=`⟨*length*⟩.

`twocolumn` sets `twocolumn` mode with `\@twocolumntrue`. `twocolumn=false` denotes onecolumn mode with`\@twocolumnfalse`.

`twoside` sets both `\@twosidetrue` and `\@mparswitchtrue`. See Section 6.3.

`textwidth` sets `\textwidth` directly. See Section 6.2.

`textheight` sets `\textheight` directly. See Section 6.2.

`reversemp` | `reversemarginpar` makes the marginal notes appear in the left (inner) margin with `\@reversemargintrue`. Unlike version 2.3 or earlier, it does *not* change `includemp` mode. This is `false` by default.

6.5 drivers

Package geometry supports `dvips`, `dvipdfm` including its derivatives dvipdfmx and xdvipdfmx, `pdftex` for pdflatex, and `vtex` for VTEX environment. These driver options are exclusive. The driver can be set by either `driver=`⟨*driver name*⟩ or any of the drivers directly like `pdftex`. A driver auto-detection mechanism is introduced in version 4. Therefore, you don't have to set a driver in most cases, except for `dvipdfm`. Setting `driver=auto` makes the auto-detection work whatever the previous setting is. Setting `driver=none` does nothing for driver.

`driver` sets driver. `driver=`⟨*driver name*⟩. `dvips`, `dvipdfm`, `pdftex`, `vtex`, `auto` and `none` are available as a driver name.

The options below can be set directly instead of `driver=`⟨*value*⟩.

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

`dvipdfm` works like `dvips` except landscape correction.

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

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

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

6.6 Other options

The other useful options are described here.

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

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

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

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

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

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

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

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

`compat2` sets all kind of options so that `\usepackage[compat2]{geometry}` would behave as if one is using the old version (v2.3) with the old default layout: `[scale={0.8,0.9}, centering, includeheadfoot]`, which is here expressed by options available in version 3. Note this option should be set as a first option.

7 Default settings

7.1 Default layout

Let us recapitulate the default layout here. The geometry package has the following default page layout for onesided documents:

```
scale=0.7, marginratio={1:1, 2:3}, ignoreall
```

For twoside, the horizontal margin ratio is also set `2:3`,

```
scale=0.7, marginratio=2:3, ignoreall.
```

Of course, you don't need to set them explicitly. `\usepackage{geometry}` will internally set the above options. Additional options will overwrite the layout dimensions. For example,

```
\usepackage[hmargin=2cm]{geometry}
```

will overwrite horizontal dimensions, but use the default for vertical layout. Page dimensions specified by the documentclass being used and other direct settings before geometry is loaded are passed down to geometry.

Note version 2.3 or earlier had default layout different from the version 3. The old default options can be expressed with options available in the current version:

```
scale={0.8,0.9}, centering, includeheadfoot.
```

Adding `compat2` as a first option sets those options so that, for example,

```
\usepackage[compat2, width=10cm]{geometry}
```

would behave as if one is using the old version (v2.3).

7.2 Configuration file

One can set up a configuration file to make default options. To do this, produce a file `geometry.cfg` containing an `\ExecuteOptions` macro, for example,

```
\ExecuteOptions{a4paper,dvips}
```

and install it somewhere TEX can find it.

The options specified in the `geometry.cfg` can be cleared by option `reset`.

8 Relations between options

This section shows how complexity is solved when options are over-specified.

8.1 Order dependence

The geometry options are basically order-independent, but there are some exceptions. For multiple specification of the same option, the last setting is adopted. For example,

```
verbose=true, verbose=false
```

obviously results in `verbose=false`. If you set

```
hmargin={3cm,2cm}, left=1cm
```

the left(or inner) margin is overwritten by `left=1cm`. As a result, it is equivalent to `hmargin={1cm,2cm}`.

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. As a result, this case is 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.

8.2 Priority

There are several ways to set dimensions of the printable area: `scale`, `total`, `text` and `lines`. Basically specification with the more concrete dimension has the higher priority:

$$\text{low} \quad \longrightarrow \quad \text{high} \quad \text{(priority)}$$

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

For example,

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

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

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

results in `[lines=30, textwidth=7in]`.

Options determining margin size also have priority rule: margin ratios versus margin length. For example, if both `marginratio=1:2` and `margin=1cm` are set at the same time, `margin=1cm` wins because `margin=1cm` is more concrete dimension than ratios. That is why normal margin options work well with default margin ratios (`marginratio={1:1, 2:3}` for oneside).

$$\text{low} \quad \longrightarrow \quad \text{high} \quad \text{(priority)}$$

$$\left\{\begin{array}{l}\texttt{hmarginratio}\\ \texttt{vmarginratio}\\ \texttt{marginratio}\end{array}\right\} < \left\{\begin{array}{l}\texttt{hmargin}\ \textit{or}\ \texttt{left}\ \&\ \texttt{right}\\ \texttt{vmargin}\ \textit{or}\ \texttt{top}\ \&\ \texttt{bottom}\\ \texttt{margin}\end{array}\right\}.$$

9 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 the case without `bindingoffset=1cm` by $0.7 \times$ `1cm` (=`0.7cm`).
- 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 6.2 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`.

- An old style setting with v2.3 or earlier

```
\usepackage[a4paper,mag=1200,truedimen,margin=2cm,
    twosideshift=10pt,
    headsep=7pt,headheight=14.5pt,
    marginparwidth=30pt]{geometry}
```

can be rewritten with options in version 3 without `compat2`:

```
\usepackage{calc}
\usepackage[a4paper,mag=1200,truedimen,margin=2cm,
    twoside, left=2cm+10pt, right=2cm-10pt,
    includeheadfoot, headsep=7pt,headheight=14.5pt,
    includemp, marginparwidth=30pt]{geometry}
```

In this case, `includeall` can be used instead of `includeheadfoot` and `includemp`.

- 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. `:-)`

10 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.

11 Acknowledgments

The author appreciates helpful suggestions and comments from Jean-Bernard Addor, Frank Bennett, Alexis Dimitriadis, Friedrich Flender, Stephan Hennig, Morten Høgholm, Jonathan Kew, James Kilfiger, Jean-Marc Lasgouttes, Wlodzimierz Macewicz, Rolf Niepraschk, Hans Fr. Nordhaug, Keith Reckdahl, Peter Riocreux, Will Robertson, Nico Schlöemer, Perry C. Stearns, Frank Stengel, Plamen Tanovski, Petr Uher, Piet van Oostrum, Vladimir Volovich, and Michael Vulis.

The author is deeply grateful to Frank Mittelbach for checking the codes patiently and providing extremely helpful insight and suggestions for version 3.

12 Implementation

1 ⟨∗package⟩

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

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

Internal switches are declared here.

```
5 \newif\ifGm@verbose
```

```
\newif\ifGm@landscape
\newif\ifGm@includehead
\newif\ifGm@includefoot
\newif\ifGm@includemp
\newif\ifGm@hbody
\newif\ifGm@vbody
\newif\ifGm@heightrounded
\newif\ifGm@showframe
\newif\ifGm@compatii
\newif\ifGm@sworient\Gm@sworientfalse
\newif\ifGm@pass\Gm@passfalse
\newif\ifGm@resetpaper
```

`\Gm@cnth` `\Gm@cntv` Counters for horizontal and vertical partitioning patterns.

```
\newcount\Gm@cnth
\newcount\Gm@cntv
```

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

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

`\Gm@bindingoffset` An additional inner offset for binding.

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

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

```
\newdimen\Gm@wd@mp
\newdimen\Gm@odd@mp
\newdimen\Gm@even@mp
```

`\Gm@dimlist` Native dimension setting list.

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

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

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

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

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

`\Gm@Dhscale` `\Gm@Dvscale` The default values for the horizontal and vertical *scale* are defined. In version 3 the default scale has been changed from {0.8, 0.9} to {0.7, 0.7} in each direction.

```
\def\Gm@Dhscale{0.7}%
\def\Gm@Dvscale{0.7}%
\@onlypreamble\Gm@Dhscale
\@onlypreamble\Gm@Dvscale
```

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

```
\def\Gm@dvips{dvips}%
\def\Gm@dvipdfm{dvipdfm}%
\def\Gm@pdftex{pdftex}%
\def\Gm@vtex{vtex}%
\@onlypreamble\Gm@dvips
\@onlypreamble\Gm@dvipdfm
\@onlypreamble\Gm@pdftex
\@onlypreamble\Gm@vtex
```

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

\Gm@orgpw
\Gm@orgph
These macros keep original paper (media) size intact.
```
\edef\Gm@orgpw{\the\paperwidth}%
\edef\Gm@orgph{\the\paperheight}%
```

\Gm@dorg
The macro saves LaTeX native dimensions and switches before processing geometry options, and is called when `reset` or `pass` is set.
```
\edef\Gm@dorg{%
  \noexpand\setlength{\paperwidth}{\the\paperwidth}%
  \noexpand\setlength{\paperheight}{\the\paperheight}%
  \noexpand\setlength{\textheight}{\the\textheight}%
  \noexpand\setlength{\textwidth}{\the\textwidth}%
  \noexpand\setlength{\oddsidemargin}{\the\oddsidemargin}%
  \noexpand\setlength{\evensidemargin}{\the\evensidemargin}%
  \noexpand\setlength{\topmargin}{\the\topmargin}%
  \noexpand\setlength{\headsep}{\the\headsep}%
  \noexpand\setlength{\headheight}{\the\headheight}%
  \noexpand\setlength{\footskip}{\the\footskip}%
  \noexpand\setlength{\marginparwidth}{\the\marginparwidth}%
  \noexpand\setlength{\marginparsep}{\the\marginparsep}%
  \noexpand\setlength{\columnsep}{\the\columnsep}%
  \noexpand\setlength{\skip\footins}{\the\skip\footins}%
  \noexpand\setlength{\hoffset}{\the\hoffset}%
  \noexpand\setlength{\voffset}{\the\voffset}%
  \expandafter\noexpand\csname @twocolumn\if@twocolumn
    \Gm@true\else\Gm@false\fi\endcsname
  \expandafter\noexpand\csname @twoside\if@twoside
    \Gm@true\else\Gm@false\fi\endcsname
  \expandafter\noexpand\csname @mparswitch\if@mparswitch
    \Gm@true\else\Gm@false\fi\endcsname
  \expandafter\noexpand\csname @reversemargin\if@reversemargin
    \Gm@true\else\Gm@false\fi\endcsname
  \noexpand\mag=\the\mag}%
\@onlypreamble\Gm@dorg
```

\Gm@init
The macro for initializing modes and flags is defined here. This macro is called at the beginning of the package and when `reset` is specified.
```
\def\Gm@init{%
  \Gm@hbodyfalse\Gm@vbodyfalse
  \Gm@includeheadfalse\Gm@includefootfalse\Gm@includempfalse
  \Gm@landscapefalse\Gm@compatiifalse\Gm@heightroundedfalse
  \Gm@verbosefalse\Gm@showframefalse\Gm@resetpaperfalse
  \let\Gm@paper\@undefined
  \let\Gm@width\@undefined\let\Gm@height\@undefined
  \let\Gm@textwidth\@undefined\let\Gm@textheight\@undefined
  \let\Gm@hscale\@undefined\let\Gm@vscale\@undefined
  \let\Gm@hmarginratio\@undefined\let\Gm@vmarginratio\@undefined
  \let\Gm@lmargin\@undefined\let\Gm@rmargin\@undefined
  \let\Gm@tmargin\@undefined\let\Gm@bmargin\@undefined
  \let\Gm@driver\@empty\let\Gm@truedimen\@empty
  \Gm@bindingoffset\z@\Gm@dimlist={}}%
\@onlypreamble\Gm@init
```

\Gm@setdriver
The macro sets the specified driver.
```
\def\Gm@setdriver#1{%
  \expandafter\let\expandafter\Gm@driver\csname Gm@#1\endcsname}%
```

\Gm@unsetdriver
The macro unsets the specified driver if it has been set.
```
\def\Gm@unsetdriver#1{%
  \expandafter\ifx\csname Gm@#1\endcsname\Gm@driver
```

```
    \let\Gm@driver\@empty
  \fi}%
```

`\Gm@setbool` `\Gm@setboolrev` The macros set a boolean option.

```
\def\Gm@setbool{\@dblarg\Gm@@setbool}%
\def\Gm@setboolrev{\@dblarg\Gm@@setboolrev}%
\def\Gm@@setbool[#1]#2#3{\Gm@doif{#1}{#3}{\csname Gm@#2\Gm@bool\endcsname}}%
\def\Gm@@setboolrev[#1]#2#3{\Gm@doifelse{#1}{#3}%
  {\csname Gm@#2\Gm@false\endcsname}{\csname Gm@#2\Gm@true\endcsname}}%
\@onlypreamble\Gm@setbool
\@onlypreamble\Gm@setboolrev
\@onlypreamble\Gm@@setbool
\@onlypreamble\Gm@@setboolrev
```

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

```
\def\Gm@doif#1#2#3{%
  \lowercase{\def\Gm@bool{#2}}%
  \ifx\Gm@bool\@empty
    \let\Gm@bool\Gm@true
  \fi
  \ifx\Gm@bool\Gm@true
  \else
    \ifx\Gm@bool\Gm@false
    \else
      \let\Gm@bool\relax
    \fi
  \fi
  \ifx\Gm@bool\relax
    \Gm@warning{‘#1’ should be set to ‘true’ or ‘false’}%
  \else
    #3
  \fi}%
\def\Gm@doifelse#1#2#3#4{%
  \Gm@doif{#1}{#2}{\ifx\Gm@bool\Gm@true #3\else #4\fi}}%
\@onlypreamble\Gm@doif
\@onlypreamble\Gm@doifelse
```

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

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

`\Gm@checkbool` The macro is used in `\Gm@showparams` to print `true` or nothing.

```
\def\Gm@checkbool#1{#1: \csname ifGm@#1\endcsname true\else --\fi^^J}%
\@onlypreamble\Gm@checkbool
```

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

```
\def\Gm@defbylen#1#2{%
  \setlength\@tempdima{#2}%
  \expandafter\edef\csname Gm@#1\endcsname{\the\@tempdima}}%
\def\Gm@defbycnt#1#2{%
  \setcounter{Gm@tempcnt}{#2}%
  \expandafter\edef\csname Gm@#1\endcsname{\the\value{Gm@tempcnt}}}%
\@onlypreamble\Gm@defbylen
\@onlypreamble\Gm@defbycnt
```

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

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

`\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.

```
\def\Gm@setbyratio[#1]#2#3#4{% determine #4 by ratio
  \expandafter\Gm@sep@ratio\Gm@mratio\relax
  \if#1b
    \edef\@@tempa{\the\@tempcnta}%
    \@tempcnta=\@tempcntb
    \@tempcntb=\@@tempa\relax
  \fi
  \expandafter\setlength\expandafter\@tempdimb\expandafter
    {\csname Gm@#3\endcsname}%
  \ifnum\@tempcntb>\z@
    \multiply\@tempdimb\@tempcnta
    \divide\@tempdimb\@tempcntb
  \fi
  \expandafter\edef\csname Gm@#4\endcsname{\the\@tempdimb}}%
\@onlypreamble\Gm@setbyratio
```

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

```
\def\Gm@detiv#1#2#3#4{% determine #4.
  \expandafter\setlength\expandafter\@tempdima\expandafter
    {\csname paper#1\endcsname}%
  \expandafter\setlength\expandafter\@tempdimb\expandafter
    {\csname Gm@#2\endcsname}%
  \addtolength\@tempdima{-\@tempdimb}%
  \expandafter\setlength\expandafter\@tempdimb\expandafter
    {\csname Gm@#3\endcsname}%
  \addtolength\@tempdima{-\@tempdimb}%
  \ifdim\@tempdima<\z@
    \Gm@warning{‘#4’ results in NEGATIVE (\the\@tempdima).%
    ^^J\@spaces ‘#2’ or ‘#3’ should be shortened in length}%
  \fi
  \expandafter\edef\csname Gm@#4\endcsname{\the\@tempdima}}%
\@onlypreamble\Gm@detiv
```

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

```
\def\Gm@detiiandiii#1#2#3{% determine #2 and #3.
  \expandafter\setlength\expandafter\@tempdima\expandafter
    {\csname paper#1\endcsname}%
  \expandafter\setlength\expandafter\@tempdimb\expandafter
    {\csname Gm@#1\endcsname}%
  \addtolength\@tempdima{-\@tempdimb}%
  \ifdim\@tempdima<\z@
    \Gm@warning{‘#2’ and ‘#3’ result in NEGATIVE (\the\@tempdima).%
                  ^^J\@spaces ‘#1’ should be shortened in length}%
  \fi
  \ifx\Gm@mratio\@undefined
    \divide\@tempdima\tw@
    \@tempdimb=\@tempdima
  \else
    \@tempdimb=\@tempdima
    \expandafter\Gm@sep@ratio\Gm@mratio\relax
    \advance\@tempcntb\@tempcnta
    \ifnum\@tempcntb>\z@
      \divide\@tempdima\@tempcntb
      \multiply\@tempdima\@tempcnta
      \advance\@tempdimb-\@tempdima
    \else
      \divide\@tempdima\tw@
      \@tempdimb=\@tempdima
```

```
    \fi
  \fi
  \expandafter\edef\csname Gm@#2\endcsname{\the\@tempdima}%
  \expandafter\edef\csname Gm@#3\endcsname{\the\@tempdimb}}%
\@onlypreamble\Gm@detiiandiii
```

`\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`.

```
\def\Gm@detall#1#2#3#4{%
  \@tempcnta\z@
  \edef\Gm@mratio{\@nameuse{Gm@#1marginratio}}%
```

`\@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.

```
  \if#1h
    \ifx\Gm@lmargin\@undefined\else\advance\@tempcnta4\relax\fi
    \ifGm@hbody\advance\@tempcnta2\relax\fi
    \ifx\Gm@rmargin\@undefined\else\advance\@tempcnta1\relax\fi
    \Gm@cnth\@tempcnta
  \else
    \ifx\Gm@tmargin\@undefined\else\advance\@tempcnta4\relax\fi
    \ifGm@vbody\advance\@tempcnta2\relax\fi
    \ifx\Gm@bmargin\@undefined\else\advance\@tempcnta1\relax\fi
    \Gm@cntv\@tempcnta
  \fi
```

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

```
  \ifcase\@tempcnta
    \if#1h
      \edef\Gm@width{\Gm@Dhscale\paperwidth}%
    \else
      \edef\Gm@height{\Gm@Dvscale\paperheight}%
    \fi
    \Gm@detiiandiii{#2}{#3}{#4}%
```

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

```
  \or\Gm@setbyratio[f]{#1}{#4}{#3}\Gm@detiv{#2}{#3}{#4}{#2}%
```

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

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

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

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

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

```
  \or\Gm@setbyratio[b]{#1}{#3}{#4}\Gm@detiv{#2}{#3}{#4}{#2}%
```

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

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

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

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

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

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

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

```
\def\Gm@clean{%
  \ifnum\Gm@cnth<4\let\Gm@lmargin\@undefined\fi
  \ifodd\Gm@cnth\else\let\Gm@rmargin\@undefined\fi
  \ifnum\Gm@cntv<4\let\Gm@tmargin\@undefined\fi
  \ifodd\Gm@cntv\else\let\Gm@bmargin\@undefined\fi
  \ifGm@hbody\else
```

```
    \let\Gm@hscale\@undefined
    \let\Gm@width\@undefined
    \let\Gm@textwidth\@undefined
  \fi
  \ifGm@vbody\else
    \let\Gm@vscale\@undefined
    \let\Gm@height\@undefined
    \let\Gm@textheight\@undefined
  \fi
  \if@twoside
    \ifx\Gm@hmarginratio\Gm@Dhratiotwo
      \let\Gm@hmarginratio\@undefined
    \fi
  \else
    \ifx\Gm@hmarginratio\Gm@Dhratio
      \let\Gm@hmarginratio\@undefined
    \fi
  \fi}%
\@onlypreamble\Gm@clean
```

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

```
\def\Gm@parse@divide#1#2#3#4{%
  \def\Gm@star{*}%
  \@tempcnta\z@
  \@for\Gm@tmp:=#1\do{%
    \expandafter\KV@@sp@def\expandafter\Gm@frag\expandafter{\Gm@tmp}%
    \edef\Gm@value{\Gm@frag}%
    \ifcase\@tempcnta\relax\edef\Gm@key{#2}%
      \or\edef\Gm@key{#3}%
      \else\edef\Gm@key{#4}%
    \fi
    \@nameuse{Gm@set\Gm@key false}%
    \ifx\empty\Gm@value\else
    \ifx\Gm@star\Gm@value\else
      \setkeys{Gm}{\Gm@key=\Gm@value}%
    \fi\fi
    \advance\@tempcnta\@ne}%
  \let\Gm@star\relax}%
\@onlypreamble\Gm@parse@divide
```

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

```
\def\Gm@branch#1#2#3{%
  \@tempcnta\z@
  \@for\Gm@tmp:=#1\do{%
    \KV@@sp@def\Gm@frag{\Gm@tmp}%
    \edef\Gm@value{\Gm@frag}%
    \ifcase\@tempcnta\relax% cnta == 0
      \setkeys{Gm}{#2=\Gm@value}%
    \or% cnta == 1
      \setkeys{Gm}{#3=\Gm@value}%
    \else\fi
    \advance\@tempcnta\@ne}%
  \ifnum\@tempcnta=\@ne
    \setkeys{Gm}{#3=\Gm@value}%
  \fi}%
\@onlypreamble\Gm@branch
```

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

```
\def\Gm@magtooffset{%
  \@tempdima=\mag\Gm@truedimen sp%
  \@tempdimb=1\Gm@truedimen in%
  \divide\@tempdimb\@tempdima
  \multiply\@tempdimb\@m
  \addtolength{\hoffset}{1\Gm@truedimen in}%
```

```
  \addtolength{\voffset}{1\Gm@truedimen in}%
  \addtolength{\hoffset}{-\the\@tempdimb}%
  \addtolength{\voffset}{-\the\@tempdimb}}%
\@onlypreamble\Gm@magtooffset
```

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

```
\def\Gm@setafter#1#2{%
  \let\Gm@len=\relax\let\Gm@td=\relax
  \edef\addtolist{\noexpand\Gm@dimlist=%
  {\the\Gm@dimlist \Gm@len{#1}{#2}}}\addtolist}%
\@onlypreamble\Gm@setafter
```

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

```
\def\Gm@processdimlist{%
  \def\Gm@td{\Gm@truedimen}%
  \def\Gm@len##1##2{\setlength{##1}{##2}}%
  \the\Gm@dimlist}%
\@onlypreamble\Gm@processdimlist
```

`\Gm@setpaper` The macro sets `paperwidth` and `paperheight` dimensions using `\Gm@setafter` macro.

```
\def\Gm@setpaper(#1,#2)#3{%
  \let\Gm@td\relax
  \Gm@setafter\paperwidth{#1\Gm@td #3}%
  \Gm@setafter\paperheight{#2\Gm@td #3}%
  \ifGm@landscape\Gm@sworienttrue\else\Gm@sworientfalse\fi}%
\@onlypreamble\Gm@setpaper
```

`\Gm@chpaper` The macro changes the paper size.

```
\def\Gm@chpaper{\@nameuse{Gm@\Gm@paper}}%
\@onlypreamble\Gm@chpaper
```

Various paper size are defined here.

```
\@namedef{Gm@a0paper}{\Gm@setpaper(841,1189){mm}}%
\@namedef{Gm@a1paper}{\Gm@setpaper(594,841){mm}}%
\@namedef{Gm@a2paper}{\Gm@setpaper(420,594){mm}}%
\@namedef{Gm@a3paper}{\Gm@setpaper(297,420){mm}}%
\@namedef{Gm@a4paper}{\Gm@setpaper(210,297){mm}}%
\@namedef{Gm@a5paper}{\Gm@setpaper(148,210){mm}}%
\@namedef{Gm@a6paper}{\Gm@setpaper(105,148){mm}}%
\@namedef{Gm@b0paper}{\Gm@setpaper(1000,1414){mm}}%
\@namedef{Gm@b1paper}{\Gm@setpaper(707,1000){mm}}%
\@namedef{Gm@b2paper}{\Gm@setpaper(500,707){mm}}%
\@namedef{Gm@b3paper}{\Gm@setpaper(353,500){mm}}%
\@namedef{Gm@b4paper}{\Gm@setpaper(250,353){mm}}%
\@namedef{Gm@b5paper}{\Gm@setpaper(176,250){mm}}%
\@namedef{Gm@b6paper}{\Gm@setpaper(125,176){mm}}%
\@namedef{Gm@ansiapaper}{\Gm@setpaper(8.5,11){in}}%
\@namedef{Gm@ansibpaper}{\Gm@setpaper(11,17){in}}%
\@namedef{Gm@ansicpaper}{\Gm@setpaper(17,22){in}}%
\@namedef{Gm@ansidpaper}{\Gm@setpaper(22,34){in}}%
\@namedef{Gm@ansiepaper}{\Gm@setpaper(34,44){in}}%
\@namedef{Gm@letterpaper}{\Gm@setpaper(8.5,11){in}}%
\@namedef{Gm@legalpaper}{\Gm@setpaper(8.5,14){in}}%
\@namedef{Gm@executivepaper}{\Gm@setpaper(7.25,10.5){in}}%
\@namedef{Gm@screen}{\Gm@setpaper(225,180){mm}}%
```

All the available options are defined below.

`'paper'` `paper` takes paper name as its value. Available paper names are listed below.

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

`'a[0-6]paper'`
`'b[0-6]paper'`
`'ansi[a-e]paper'`
`'letterpaper'`
`'legalpaper'`
`'executivepaper'`
`'screen'`

The following paper names are available. `screen` and ANSI paper sizes have been introduced in ver.3, but of course they can't be used as a documentclass option.

```
\define@key{Gm}{a0paper}[true]{\def\Gm@paper{a0paper}\Gm@chpaper}%
\define@key{Gm}{a1paper}[true]{\def\Gm@paper{a1paper}\Gm@chpaper}%
\define@key{Gm}{a2paper}[true]{\def\Gm@paper{a2paper}\Gm@chpaper}%
\define@key{Gm}{a3paper}[true]{\def\Gm@paper{a3paper}\Gm@chpaper}%
\define@key{Gm}{a4paper}[true]{\def\Gm@paper{a4paper}\Gm@chpaper}%
\define@key{Gm}{a5paper}[true]{\def\Gm@paper{a5paper}\Gm@chpaper}%
\define@key{Gm}{a6paper}[true]{\def\Gm@paper{a6paper}\Gm@chpaper}%
\define@key{Gm}{b0paper}[true]{\def\Gm@paper{b0paper}\Gm@chpaper}%
\define@key{Gm}{b1paper}[true]{\def\Gm@paper{b1paper}\Gm@chpaper}%
\define@key{Gm}{b2paper}[true]{\def\Gm@paper{b2paper}\Gm@chpaper}%
\define@key{Gm}{b3paper}[true]{\def\Gm@paper{b3paper}\Gm@chpaper}%
\define@key{Gm}{b4paper}[true]{\def\Gm@paper{b4paper}\Gm@chpaper}%
\define@key{Gm}{b5paper}[true]{\def\Gm@paper{b5paper}\Gm@chpaper}%
\define@key{Gm}{b6paper}[true]{\def\Gm@paper{b6paper}\Gm@chpaper}%
\define@key{Gm}{ansiapaper}[true]{\def\Gm@paper{ansiapaper}\Gm@chpaper}%
\define@key{Gm}{ansibpaper}[true]{\def\Gm@paper{ansibpaper}\Gm@chpaper}%
\define@key{Gm}{ansicpaper}[true]{\def\Gm@paper{ansicpaper}\Gm@chpaper}%
\define@key{Gm}{ansidpaper}[true]{\def\Gm@paper{ansidpaper}\Gm@chpaper}%
\define@key{Gm}{ansiepaper}[true]{\def\Gm@paper{ansiepaper}\Gm@chpaper}%
\define@key{Gm}{letterpaper}[true]{\def\Gm@paper{letterpaper}\Gm@chpaper}%
\define@key{Gm}{legalpaper}[true]{\def\Gm@paper{legalpaper}\Gm@chpaper}%
\define@key{Gm}{executivepaper}[true]{\def\Gm@paper{executivepaper}%
  \Gm@chpaper}%
\define@key{Gm}{screen}[true]{\def\Gm@paper{screen}\Gm@chpaper}%
```

'paperwidth' 'paperheight' 'papersize' Direct specification for paper size is also possible.

```
\define@key{Gm}{paperwidth}{%
  \Gm@setafter\paperwidth{#1}\def\Gm@paper{user defined}}%
\define@key{Gm}{paperheight}{%
  \Gm@setafter\paperheight{#1}\def\Gm@paper{user defined}}%
\define@key{Gm}{papersize}{\Gm@branch{#1}{paperwidth}{paperheight}}%
```

'landscape' 'portrait' Paper orientation setting is also available.

```
\define@key{Gm}{landscape}[true]{\Gm@doifelse{landscape}{#1}%
  {\ifGm@landscape\else\Gm@landscapetrue\Gm@reverse{sworient}\fi}%
  {\ifGm@landscape\Gm@landscapefalse\Gm@reverse{sworient}\fi}}%
\define@key{Gm}{portrait}[true]{\Gm@doifelse{portrait}{#1}%
  {\ifGm@landscape\Gm@landscapefalse\Gm@reverse{sworient}\fi}%
  {\ifGm@landscape\else\Gm@landscapetrue\Gm@reverse{sworient}\fi}}%
```

'hscale' 'vscale' 'scale' These options can determine the length(s) of *total body* giving *scale(s)* against the paper size.

```
\define@key{Gm}{hscale}{\Gm@hbodytrue\edef\Gm@hscale{#1}}%
\define@key{Gm}{vscale}{\Gm@vbodytrue\edef\Gm@vscale{#1}}%
\define@key{Gm}{scale}{\Gm@branch{#1}{hscale}{vscale}}%
```

'width' 'height' 'total' 'totalwidth' 'totalheight' These options give concrete dimension(s) of *total body*. `totalwidth` and `totalheight` are aliases of `width` and `height` respectively.

```
\define@key{Gm}{width}{\Gm@hbodytrue\Gm@defbylen{width}{#1}}%
\define@key{Gm}{height}{\Gm@vbodytrue\Gm@defbylen{height}{#1}}%
\define@key{Gm}{total}{\Gm@branch{#1}{width}{height}}%
\let\KV@Gm@totalwidth\KV@Gm@width
\let\KV@Gm@totalheight\KV@Gm@height
```

'textwidth' 'textheight' 'text' 'body' These options directly sets the dimensions `\textwidth` and `\textheight`. `body` is an alias of `text`.

```
\define@key{Gm}{textwidth}{\Gm@hbodytrue\Gm@defbylen{textwidth}{#1}}%
\define@key{Gm}{textheight}{\Gm@vbodytrue\Gm@defbylen{textheight}{#1}}%
\define@key{Gm}{text}{\Gm@branch{#1}{textwidth}{textheight}}%
\let\KV@Gm@body\KV@Gm@text
```

'lines' The option sets `\textheight` with the number of lines.

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

'includehead' 'includefoot' 'includeheadfoot' 'includemp' 'includeall'

include* options include the corresponding part(s) in *total body*.

```
\define@key{Gm}{includehead}[true]{\Gm@setbool{includehead}{#1}}%
\define@key{Gm}{includefoot}[true]{\Gm@setbool{includefoot}{#1}}%
\define@key{Gm}{includeheadfoot}[true]{\Gm@doifelse{includeheadfoot}{#1}%
  {\Gm@includeheadtrue\Gm@includefoottrue}%
  {\Gm@includeheadfalse\Gm@includefootfalse}}%
\define@key{Gm}{includemp}[true]{\Gm@setbool{includemp}{#1}}%
\define@key{Gm}{includeall}[true]{\Gm@doifelse{includeall}{#1}%
  {\Gm@includeheadtrue\Gm@includefoottrue\Gm@includemptrue}%
  {\Gm@includeheadfalse\Gm@includefootfalse\Gm@includempfalse}}%
```

'ignorehead' 'ignorefoot' 'ignoreheadfoot' 'ignoremp' 'ignoreall'

ignore* options disregard *head*, *foot* and *marginpars* in determining the location of *total body*.

```
\define@key{Gm}{ignorehead}[true]{%
  \Gm@setboolrev[ignorehead]{includehead}{#1}}%
\define@key{Gm}{ignorefoot}[true]{%
  \Gm@setboolrev[ignorefoot]{includefoot}{#1}}%
\define@key{Gm}{ignoreheadfoot}[true]{\Gm@doifelse{ignoreheadfoot}{#1}%
  {\Gm@includeheadfalse\Gm@includefootfalse}%
  {\Gm@includeheadtrue\Gm@includefoottrue}}%
\define@key{Gm}{ignoremp}[true]{%
  \Gm@setboolrev[ignoremp]{includemp}{#1}}%
\define@key{Gm}{ignoreall}[true]{\Gm@doifelse{ignoreall}{#1}%
  {\Gm@includeheadfalse\Gm@includefootfalse\Gm@includempfalse}%
  {\Gm@includeheadtrue\Gm@includefoottrue\Gm@includemptrue}}%
```

'heightrounded'

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

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

'hdivide' 'vdivide' 'divide'

The options are useful to specify partitioning in each direction of the paper.

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

'lmargin' 'rmargin' 'tmargin' 'bmargin' 'left' 'inner' 'innermargin' 'right' 'outer' 'outermargin' 'top' 'bottom'

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

```
\define@key{Gm}{lmargin}{\Gm@defbylen{lmargin}{#1}}%
\define@key{Gm}{rmargin}{\Gm@defbylen{rmargin}{#1}}%
\let\KV@Gm@left\KV@Gm@lmargin
\let\KV@Gm@inner\KV@Gm@lmargin
\let\KV@Gm@innermargin\KV@Gm@lmargin
\let\KV@Gm@right\KV@Gm@rmargin
\let\KV@Gm@outer\KV@Gm@rmargin
\let\KV@Gm@outermargin\KV@Gm@rmargin
\define@key{Gm}{tmargin}{\Gm@defbylen{tmargin}{#1}}%
\define@key{Gm}{bmargin}{\Gm@defbylen{bmargin}{#1}}%
\let\KV@Gm@top\KV@Gm@tmargin
\let\KV@Gm@bottom\KV@Gm@bmargin
```

'hmargin' 'vmargin' 'margin'

These options are shorthands for setting *margins*.

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

'hmarginratio' 'vmarginratio' 'marginratio' 'hratio' 'vratio' 'ratio'

Options specifying the margin ratios.

```
\define@key{Gm}{hmarginratio}{\edef\Gm@hmarginratio{#1}}%
\define@key{Gm}{vmarginratio}{\edef\Gm@vmarginratio{#1}}%
\define@key{Gm}{marginratio}{\Gm@branch{#1}{hmarginratio}{vmarginratio}}%
\let\KV@Gm@hratio\KV@Gm@hmarginratio
\let\KV@Gm@vratio\KV@Gm@vmarginratio
\let\KV@Gm@ratio\KV@Gm@marginratio
```

'hcentering' Useful shorthands to make *body* centered.

'vcentering'
'centering'

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

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

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

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

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

'bindingoffset' The macro specifies a white space added to the left or inner margin.

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

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

'headsep'
'footskip'
'head'
'foot'

```
\define@key{Gm}{headheight}{\Gm@setafter\headheight{#1}}%
\define@key{Gm}{headsep}{\Gm@setafter\headsep{#1}}%
\define@key{Gm}{footskip}{\Gm@setafter\footskip{#1}}%
\let\KV@Gm@head\KV@Gm@headheight
\let\KV@Gm@foot\KV@Gm@footskip
```

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

'nofoot'
'noheadfoot'

```
\define@key{Gm}{nohead}[true]{\Gm@doifelse{nohead}{#1}%
  {\Gm@setafter\headheight\z@\Gm@setafter\headsep\z@}{}}%
\define@key{Gm}{nofoot}[true]{\Gm@doifelse{nofoot}{#1}%
  {\Gm@setafter\footskip\z@}{}}%
\define@key{Gm}{noheadfoot}[true]{\Gm@doifelse{noheadfoot}{#1}%
  {\Gm@setafter\headheight\z@\Gm@setafter\headsep
  \z@\Gm@setafter\footskip\z@}{}}%
```

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

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

'marginparwidth' They directly set native dimensions `\marginparwidth` and `\marginparsep`. For compatibility, `includemp` is set to `true` if `compat2` is set.

'marginpar'
'marginparsep'

```
\define@key{Gm}{marginparwidth}{\ifGm@compatii\Gm@includemptrue\fi
  \Gm@setafter\marginparwidth{#1}}%
\let\KV@Gm@marginpar\KV@Gm@marginparwidth
\define@key{Gm}{marginparsep}{\ifGm@compatii\Gm@includemptrue\fi
  \Gm@setafter\marginparsep{#1}}%
```

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

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

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

```
\define@key{Gm}{columnsep}{\Gm@setafter\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'

```
\define@key{Gm}{hoffset}{\Gm@setafter\hoffset{#1}}%
\define@key{Gm}{voffset}{\Gm@setafter\voffset{#1}}%
\define@key{Gm}{offset}{\Gm@branch{#1}{hoffset}{voffset}}%
```

'twocolumn' The option sets `\twocolumn` switch.

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

'reversemp' The both options set \reversemargin.

'reversemarginpar'

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

'dviver'

```
\define@key{Gm}{driver}{\edef\@@tempa{#1}\edef\@@auto{auto}\edef\@@none{none}%
  \ifx\@@tempa\@empty\let\Gm@driver\relax\else
  \ifx\@@tempa\@@none\let\Gm@driver\relax\else
  \ifx\@@tempa\@@auto\let\Gm@driver\@empty\else
  \setkeys{Gm}{#1}\fi\fi\fi\let\@@auto\relax\let\@@none\relax}%
```

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

'dvipdfm'

'pdftex'

'vtex'

```
\define@key{Gm}{dvips}[true]{%
  \Gm@doifelse{dvips}{#1}{\Gm@setdriver{dvips}}{\Gm@unsetdriver{dvips}}}%
\define@key{Gm}{dvipdfm}[true]{%
  \Gm@doifelse{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{\Gm@unsetdriver{dvipdfm}}}%
\define@key{Gm}{pdftex}[true]{%
  \Gm@doifelse{pdftex}{#1}{\Gm@setdriver{pdftex}}{\Gm@unsetdriver{pdftex}}}%
\define@key{Gm}{vtex}[true]{%
  \Gm@doifelse{vtex}{#1}{\Gm@setdriver{vtex}}{\Gm@unsetdriver{vtex}}}%
```

'verbose' The verbose mode.

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

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

```
\define@key{Gm}{reset}[true]{\Gm@doifelse{reset}{#1}%
  {\Gm@init\Gm@dorg\ProcessOptionsKV[c]{Gm}\Gm@setdefaultpaper}{}}%
```

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

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

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

```
\define@key{Gm}{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.

```
\define@key{Gm}{truedimen}[true]{\Gm@doifelse{truedimen}{#1}%
  {\let\Gm@truedimen\Gm@true}{\let\Gm@truedimen\@empty}}%
```

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

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

'showframe' The showframe option.

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

'compat2' The option sets the old default options for compatibility with version 2. compat2=false does nothing.

```
\define@key{Gm}{compat2}[true]{%
  \Gm@doifelse{compat2}{#1}{\Gm@compatiitrue
  \setkeys{Gm}{scale={0.8,0.9},centering,includeheadfoot}}{}}%
```

Option twosideshift has been obsoleted. But for compatibility with version 2, one can use twosideshift when compat2 is set to true.

```
\define@key{Gm}{twosideshift}{%
  \ifGm@compatii\@twosidetrue\@mparswitchtrue\Gm@defbylen{twosideshift}{#1}%
  \else\Gm@warning{'twosideshift' is obsolete}%
  \fi}%
```

`\Gm@setdefaultpaper` The macro stores paper dimensions. This macro should be called after `\ProcessOptionsKV[c]{Gm}`.

```
\def\Gm@setdefaultpaper{%
  \ifx\Gm@paper\@undefined
    \Gm@setpaper(\strip@pt\paperwidth,\strip@pt\paperheight){pt}%
    \Gm@sworientfalse
  \fi}%
\@onlypreamble\Gm@setdefaultpaper
```

`\Gm@checkpaper` The macro checks if paperwidth/height is larger than 0pt, which is used in `\Gm@process`.

```
\def\Gm@checkpaper{%
  \ifdim\paperwidth>\p@\else
    \PackageError{geometry}{%
    You must set \string\paperwidth\space properly}{%
    Set your paper type (e.g., 'a4paper' for A4) as a class option^^J%
    or as a geometry package option.}%
  \fi
  \ifdim\paperheight>\p@\else
    \PackageError{geometry}{%
    You must set \string\paperheight\space properly}{%
    Set your paper type (e.g., 'a4paper' for A4) as a class option^^J%
    or as a geometry package option.}%
  \fi}%
```

`\Gm@checkmp` The macro checks if marginpars fall off the page.

```
\def\Gm@checkmp{%
  \ifGm@includemp\else
    \@tempcnta\z@\@tempcntb\@ne
    \if@twocolumn
      \@tempcnta\@ne
    \else
      \if@reversemargin
        \@tempcnta\@ne\@tempcntb\z@
      \fi
    \fi
    \@tempdima\marginparwidth
    \advance\@tempdima\marginparsep
    \ifnum\@tempcnta=\@ne
      \@tempdimc\@tempdima
      \setlength\@tempdimb{\Gm@lmargin}%
      \advance\@tempdimc-\@tempdimb
      \ifdim\@tempdimc>\z@
        \Gm@warning{The marginal notes would fall off the page.^^J
           \@spaces Add \the\@tempdimc\space and more to the left margin}%
      \fi
    \fi
    \ifnum\@tempcntb=\@ne
      \@tempdimc\@tempdima
      \setlength\@tempdimb{\Gm@rmargin}%
      \advance\@tempdimc-\@tempdimb
      \ifdim\@tempdimc>\z@
        \Gm@warning{The marginal notes would fall off the page.^^J
           \@spaces Add \the\@tempdimc\space and more to the right margin}%
      \fi
    \fi
  \fi}%
\@onlypreamble\Gm@checkmp
```

`\Gm@checkdrivers` 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 version 4 with packages ifpdf and ifvtex.

```
\def\Gm@checkdrivers{%
```

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

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

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

```
    \ifpdf
      \Gm@setdriver{pdftex}%
    \else
      \Gm@setdriver{dvips}%
    \fi
```

XeTEX supports the same page size parameter as pdfTEX.

```
    \@ifundefined{XeTeXrevision}{}{\Gm@setdriver{pdftex}}%
```

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

```
    \ifvtex
      \Gm@setdriver{vtex}%
    \fi
```

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

```
  \else
    \ifx\Gm@driver\Gm@pdftex
      \ifpdf\else
         \@ifundefined{XeTeXrevision}{\Gm@warning{%
            Wrong driver setting: 'pdftex'; using default driver}%
            \Gm@setdriver{dvips}}{}%
      \fi
    \fi
    \ifx\Gm@driver\Gm@vtex
      \ifvtex\else
        \Gm@warning{Wrong driver setting: 'vtex'; using default driver}%
        \Gm@setdriver{dvips}%
      \fi
    \fi
  \fi}%
\@onlypreamble\Gm@checkdrivers
```

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

```
\def\Gm@mpfix{%
  \@tempdimb\marginparwidth
  \advance\@tempdimb\marginparsep
  \Gm@wd@mp\@tempdimb
  \Gm@odd@mp\z@
  \Gm@even@mp\z@
  \if@twocolumn
    \Gm@wd@mp2\@tempdimb
    \Gm@odd@mp\@tempdimb
    \Gm@even@mp\@tempdimb
  \else
    \if@reversemargin
      \Gm@odd@mp\@tempdimb
      \if@mparswitch\else
        \Gm@even@mp\@tempdimb
      \fi
    \else
      \if@mparswitch
        \Gm@even@mp\@tempdimb
      \fi
    \fi
  \fi}%
\@onlypreamble\Gm@mpfix
```

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

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

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

```
  \ifGm@pass
```

```
    \Gm@dorg
  \else
```
The stored native dimension settings are processed here.
```
  \Gm@processdimlist
```
The margin ratios are set to the default if not specified.
```
  \ifx\Gm@hmarginratio\@undefined
    \if@twoside
      \edef\Gm@hmarginratio{\Gm@Dhratiotwo}%
    \else
      \edef\Gm@hmarginratio{\Gm@Dhratio}%
    \fi
  \fi
  \ifx\Gm@vmarginratio\@undefined
    \edef\Gm@vmarginratio{\Gm@Dvratio}%
  \fi
```
The paper size is checked here.
```
  \Gm@checkpaper
```
The paper dimensions can be swapped when paper orientation is changed over by `landscape` and `portrait` options.
```
  \ifGm@sworient
    \setlength\@tempdima{\paperwidth}%
    \setlength\paperwidth{\paperheight}%
    \setlength\paperheight{\@tempdima}%
    \Gm@setpaper(\strip@pt\paperwidth,\strip@pt\paperheight){pt}%
    \Gm@sworientfalse
  \fi
```
The bindingoffset value is removed from the paper width, which will be set back after auto-completion calculation.
```
  \addtolength\paperwidth{-\Gm@bindingoffset}%
```
The local variables are set here for marginpar correction `\Gm@wd@mp`, `\Gm@odd@mp` and `\Gm@even@mp` when `includemp` is set.
```
  \ifGm@includemp
    \Gm@mpfix
  \fi
```
If the horizontal dimension of *body* is specified by user, `\Gm@width` is set properly here.
```
  \ifGm@hbody
    \ifx\Gm@width\@undefined
      \ifx\Gm@hscale\@undefined
        \edef\Gm@width{\Gm@Dhscale\paperwidth}%
      \else
        \edef\Gm@width{\Gm@hscale\paperwidth}%
      \fi
    \fi
    \ifx\Gm@textwidth\@undefined\else
      \setlength\@tempdima{\Gm@textwidth}%
      \ifGm@includemp
        \advance\@tempdima\Gm@wd@mp
      \fi
      \edef\Gm@width{\the\@tempdima}%
    \fi
  \fi
```
If the vertical dimension of *body* is specified by user, `\Gm@height` is set properly here.
```
  \ifGm@vbody
    \ifx\Gm@height\@undefined
      \ifx\Gm@vscale\@undefined
        \edef\Gm@height{\Gm@Dvscale\paperheight}%
      \else
        \edef\Gm@height{\Gm@vscale\paperheight}%
      \fi
```

```
    \fi
    \ifx\Gm@lines\@undefined\else
```

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

```
      \ifdim\topskip<\ht\strutbox
        \setlength\@tempdima{\topskip}%
        \setlength\topskip{\ht\strutbox}%
        \Gm@warning{\noexpand\topskip was changed from \the\@tempdima\space
        to \the\topskip}%
      \fi
      \setlength\@tempdima{\baselineskip}%
      \multiply\@tempdima\Gm@lines
      \addtolength\@tempdima{\topskip}%
      \addtolength\@tempdima{-\baselineskip}%
      \edef\Gm@textheight{\the\@tempdima}%
    \fi
    \ifx\Gm@textheight\@undefined\else
      \setlength\@tempdima{\Gm@textheight}%
      \ifGm@includehead
        \addtolength\@tempdima{\headheight}%
        \addtolength\@tempdima{\headsep}%
      \fi
      \ifGm@includefoot
        \addtolength\@tempdima{\footskip}%
      \fi
      \edef\Gm@height{\the\@tempdima}%
    \fi
  \fi
```

The auto-completion calculation is executed for each direction.

```
  \Gm@detall{h}{width}{lmargin}{rmargin}%
  \Gm@detall{v}{height}{tmargin}{bmargin}%
```

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

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

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

```
  \ifGm@includemp
    \advance\textwidth-\Gm@wd@mp
    \advance\oddsidemargin\Gm@odd@mp
  \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.

```
  \if@mparswitch
    \setlength\evensidemargin{\Gm@rmargin}%
    \addtolength\evensidemargin{-1\Gm@truedimen in}%
    \ifGm@includemp
      \advance\evensidemargin\Gm@even@mp
    \fi
    \ifGm@compatii
      \ifx\Gm@twosideshift\@undefined
        \def\Gm@twosideshift{20\Gm@truedimen pt}%
      \fi
      \addtolength\oddsidemargin{\Gm@twosideshift}%
      \addtolength\evensidemargin{-\Gm@twosideshift}%
    \fi
  \else
    \evensidemargin\oddsidemargin
  \fi
```

The bindingoffset correction for `\oddsidemargin`.

```
  \advance\oddsidemargin\Gm@bindingoffset
```

`\topmargin` is adjusted here.

```
  \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`.

```
  \ifGm@includehead
    \addtolength\textheight{-\headheight}%
    \addtolength\textheight{-\headsep}%
  \else
    \addtolength\topmargin{-\headheight}%
    \addtolength\topmargin{-\headsep}%
  \fi
```

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

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

If `heightrounded` is set, `\textheight` is rounded.

```
  \ifGm@heightrounded
    \setlength\@tempdima{\textheight}%
    \addtolength\@tempdima{-\topskip}%
    \@tempcnta\@tempdima
    \@tempcntb\baselineskip
    \divide\@tempcnta\@tempcntb
    \setlength\@tempdimb{\baselineskip}%
    \multiply\@tempdimb\@tempcnta
    \advance\@tempdima-\@tempdimb
    \multiply\@tempdima\tw@
    \ifdim\@tempdima>\baselineskip
      \addtolength\@tempdimb{\baselineskip}%
    \fi
    \addtolength\@tempdimb{\topskip}%
    \textheight\@tempdimb
  \fi
```

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

```
  \addtolength\paperwidth{\Gm@bindingoffset}%
 \fi}%
\@onlypreamble\Gm@process
```

`\Gm@showparam` The macro for typeout of geometry status and native dimensions for page layout.

```
\def\Gm@showparams{%
  -------------------- Geometry parameters^^J%
  \ifGm@pass
  'pass' is specified!! (disables the geometry layouter)^^J%
  \else
  paper: \ifx\Gm@paper\@undefined class default\else\Gm@paper\fi^^J%
  \Gm@checkbool{landscape}%
  twocolumn: \if@twocolumn\Gm@true\else--\fi^^J%
  twoside: \if@twoside\Gm@true\else--\fi^^J%
  asymmetric: \if@mparswitch --\else\if@twoside\Gm@true\else --\fi\fi^^J%
  h-parts: \Gm@lmargin, \Gm@width, \Gm@rmargin%
  \ifnum\Gm@cnth=\z@\space(default)\fi^^J%
  v-parts: \Gm@tmargin, \Gm@height, \Gm@bmargin%
  \ifnum\Gm@cntv=\z@\space(default)\fi^^J%
  hmarginratio: \ifnum\Gm@cnth<5 \ifnum\Gm@cnth=3--\else%
    \Gm@hmarginratio\fi\else--\fi^^J%
  vmarginratio: \ifnum\Gm@cntv<5 \ifnum\Gm@cntv=3--\else%
    \Gm@vmarginratio\fi\else--\fi^^J%
  lines: \@ifundefined{Gm@lines}{--}{\Gm@lines}^^J%
  \Gm@checkbool{heightrounded}%
```

```
  bindingoffset: \the\Gm@bindingoffset^^J%
  truedimen: \ifx\Gm@truedimen\@empty --\else\Gm@true\fi^^J%
  \Gm@checkbool{includehead}%
  \Gm@checkbool{includefoot}%
  \Gm@checkbool{includemp}%
  driver: \if\Gm@driver\relax --\else\Gm@driver\fi^^J%
  \fi
  -------------------- Page layout dimensions and switches^^J%
  \string\paperwidth\space\space\the\paperwidth^^J%
  \string\paperheight\space\the\paperheight^^J%
  \string\textwidth\space\space\the\textwidth^^J%
  \string\textheight\space\the\textheight^^J%
  \string\oddsidemargin\space\space\the\oddsidemargin^^J%
  \string\evensidemargin\space\the\evensidemargin^^J%
  \string\topmargin\space\space\the\topmargin^^J%
  \string\headheight\space\the\headheight^^J%
  \string\headsep\@spaces\the\headsep^^J%
  \string\footskip\space\space\space\the\footskip^^J%
  \string\marginparwidth\space\the\marginparwidth^^J%
  \string\marginparsep\space\space\space\the\marginparsep^^J%
  \string\columnsep\space\space\the\columnsep^^J%
  \string\skip\string\footins\space\space\the\skip\footins^^J%
  \string\hoffset\space\the\hoffset^^J%
  \string\voffset\space\the\voffset^^J%
  \string\mag\space\the\mag^^J%
  \if@twocolumn\string\@twocolumntrue\space\fi%
  \if@twoside\string\@twosidetrue\space\fi%
  \if@mparswitch\string\@mparswitchtrue\space\fi%
  \if@reversemargin\string\@reversemargintrue\space\fi^^J%
  (1in=72.27pt, 1cm=28.45pt)^^J%
  ----------------------}%
\@onlypreamble\Gm@showparams
```

`\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.

```
\def\ProcessOptionsKV{\@ifnextchar[%]
  {\@ProcessOptionsKV}{\@ProcessOptionsKV[]}}%
\def\@ProcessOptionsKV[#1]#2{%
  \let\@tempa\@empty
  \@tempcnta\z@
  \if#1p\@tempcnta\@ne\else\if#1c\@tempcnta\tw@\fi\fi
  \ifodd\@tempcnta
   \edef\@tempa{\@ptionlist{\@currname.\@currext}}%
  \else
    \@for\CurrentOption:=\@classoptionslist\do{%
      \@ifundefined{KV@#2@\CurrentOption}%
      {}{\edef\@tempa{\@tempa,\CurrentOption,}}}%
    \ifnum\@tempcnta=\z@
      \edef\@tempa{\@tempa,\@ptionlist{\@currname.\@currext}}%
    \fi
  \fi
  \edef\@tempa{\noexpand\setkeys{#2}{\@tempa}}%
  \@tempa
  \AtEndOfPackage{\let\@unprocessedoptions\relax}}%
\@onlypreamble\ProcessOptionsKV
\@onlypreamble\@ProcessOptionsKV
```

Geometry parameters are initialized here. `\Gm@init` can be called by `reset` or `pass` options.

```
\Gm@init
```

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

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

Paper dimensions given by class default are stored.

```
\Gm@setdefaultpaper
```

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

```
\def\Gm@setkeys{\setkeys{Gm}}%
\@onlypreamble\Gm@setkeys
\let\Gm@origExecuteOptions\ExecuteOptions
\let\ExecuteOptions\Gm@setkeys
```

A local configuration file may define more options. To set A4 paper as default, `geometry.cfg` gg to contain `\ExecuteOptions{a4paper}`.

```
\InputIfFileExists{geometry.cfg}{}{}%
```

The original definition for `\ExecuteOptions` macro is restored.

```
\let\ExecuteOptions\Gm@origExecuteOptions
```

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

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

Actual settings and calculation for layout dimensions are processed.

```
\Gm@process
```

`verbose`, `showframe` and driver options are processed at `\begin{document}`.

```
\AtBeginDocument{%
```

Paper size is temporally adjusted according to `\mag` for printing devices.

```
  \ifGm@resetpaper
    \edef\Gm@pw{\Gm@orgpw}%
    \edef\Gm@ph{\Gm@orgph}%
  \else
    \edef\Gm@pw{\the\paperwidth}%
    \edef\Gm@ph{\the\paperheight}%
  \fi
```

If `pass` is set to `true`, no adjustment for page dimensions is done.

```
  \ifGm@pass\else
    \ifnum\mag=\@m\else
      \Gm@magtooffset
      \divide\paperwidth\@m
      \multiply\paperwidth\the\mag
      \divide\paperheight\@m
      \multiply\paperheight\the\mag
    \fi
  \fi
```

Checking the driver options.

```
  \Gm@checkdrivers
  \ifx\Gm@driver\relax
    \typeout{*geometry detected driver: <none>*}%
  \else
    \typeout{*geometry detected driver: \Gm@driver*}%
  \fi
```

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

```
  \ifx\Gm@driver\Gm@pdftex
    \setlength\pdfpagewidth{\Gm@pw}%
    \setlength\pdfpageheight{\Gm@ph}%
    \ifnum\mag=\@m\else
      \@tempdima=\mag sp%
      \divide\pdfhorigin\@tempdima
      \multiply\pdfhorigin\@m
      \divide\pdfvorigin\@tempdima
      \multiply\pdfvorigin\@m
      \ifx\Gm@truedimen\Gm@true
        \setlength\paperwidth{\Gm@pw}%
        \setlength\paperheight{\Gm@ph}%
      \fi
    \fi
  \fi
```

With VTEX environment, VTEX variables are set here.

```
\ifx\Gm@driver\Gm@vtex
  \mediawidth=\paperwidth
  \mediaheight=\paperheight
  \ifvtexdvi
    \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
  \fi
\fi
```

If `dvips` or `dvipdfm` is set to `true`, 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.

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

When `dvipdfm` option is set and atbegshi package in 'oberdiek' bundle is loaded, `\AtBeginShipoutFirst` is used instead of `\AtBeginDvi` for compatibility with hyperref and `dvipdfm` program.

```
\else\ifx\Gm@driver\Gm@dvipdfm
  \ifcase\ifx\AtBeginShipoutFirst\relax\@ne\else
      \ifx\AtBeginShipoutFirst\@undefined\@ne\else\z@\fi\fi
    \AtBeginShipoutFirst{\special{papersize=\the\paperwidth,\the\paperheight}}%
  \or
    \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
  \fi
\fi\fi
```

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

```
\ifGm@showframe
  \AtBeginDvi{%
    \moveright\@themargin%
    \vbox to\z@{\baselineskip\z@skip\lineskip\z@skip\lineskiplimit\z@%
    \vskip\topmargin\vbox to\z@{\vss\hrule width\textwidth}%
    \vskip\headheight\vbox to\z@{\vss\hrule width\textwidth}%
    \vskip\headsep\vbox to\z@{\vss\hrule width\textwidth}%
    \hbox to\textwidth{\llap{\vrule height\textheight}\hfil%
    \vrule height\textheight}%
    \vbox to\z@{\vss\hrule width\textwidth}%
    \vskip\footskip\vbox to\z@{\vss\hrule width\textwidth}%
    \vss}}%
  \AtBeginDvi{%
    \vbox to\z@{\baselineskip\z@skip\lineskip\z@skip\lineskiplimit\z@%
    \vskip-1\Gm@truedimen in\rlap{\hskip-1\Gm@truedimen in%
    \vbox to\z@{\vbox to\z@{\vss\hrule width\paperwidth}%
    \hbox to \paperwidth{\llap{\vrule height\paperheight}\hfil%
    \vrule height\paperheight}%
    \vbox to\z@{\vss\hrule width\paperwidth}%
    \vss}}\vss}}%
\fi
```

If `verbose=true` and `pass=false`, the system checks if marginpars fall off the page.

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

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

```
\ifGm@verbose\expandafter\typeout\else\expandafter\wlog\fi
{\Gm@showparams}%
```

save memory.

```
\let\Gm@cnth\relax
\let\Gm@cntv\relax
\let\c@Gm@tempcnt\relax
\let\Gm@bindingoffset\relax
\let\Gm@wd@mp\relax
```

```
  \let\Gm@odd@mp\relax
  \let\Gm@even@mp\relax
  \let\Gm@orgpw\relax
  \let\Gm@orgph\relax
  \let\Gm@pw\relax
  \let\Gm@ph\relax
  \let\Gm@dimlist\relax}%
```

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

```
\def\geometry#1{%
  \Gm@clean
  \setkeys{Gm}{#1}%
  \Gm@process}%
\@onlypreamble\geometry
⟨/package⟩
```

13 Config file

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

```
⟨*config⟩
%<<SAVE_INTACT

%  Uncomment and edit the line below to set default options.
%\ExecuteOptions{a4paper}

%SAVE_INTACT
⟨/config⟩
```

14 Sample file

Here is an executable sample tex file.

```
⟨*samples⟩
%<<SAVE_INTACT
\documentclass{article}% uses letterpaper by default
% \documentclass[a4paper]{article}% for A4 paper
%---------------------------------------------------------------
% Edit and uncomment one of the settings below
%---------------------------------------------------------------
% \usepackage{geometry}
% \usepackage[centering]{geometry}
% \usepackage[width=10cm,vscale=.7]{geometry}
% \usepackage[margin=1cm, papersize={12cm,19cm}, resetpaper]{geometry}
% \usepackage[margin=1cm,includeheadfoot]{geometry}
\usepackage[margin=1cm,includeheadfoot,includemp]{geometry}
% \usepackage[margin=1cm,bindingoffset=1cm,twoside]{geometry}
% \usepackage[hmarginratio=2:1, vmargin=2cm]{geometry}
% \usepackage[hscale=0.5,twoside]{geometry}
% \usepackage[hscale=0.5,asymmetric]{geometry}
% \usepackage[hscale=0.5,heightrounded]{geometry}
% \usepackage[left=1cm,right=4cm,top=2cm,includefoot]{geometry}
% \usepackage[lines=20,left=2cm,right=6cm,top=2cm,twoside]{geometry}
% \usepackage[width=15cm, marginparwidth=3cm, includemp]{geometry}
% \usepackage[hdivide={1cm,,2cm}, vdivide={3cm,8in,}, nohead]{geometry}
% \usepackage[headsep=20pt, head=40pt,foot=20pt,includeheadfoot]{geometry}
% \usepackage[text={6in,8in}, top=2cm, left=2cm]{geometry}
% \usepackage[centering,includemp,twoside,landscape]{geometry}
% \usepackage[mag=1414,margin=2cm]{geometry}
% \usepackage[mag=1414,margin=2truecm,truedimen]{geometry}
% \usepackage[compat2,marginpar=50pt,twosideshift=50pt]{geometry}
% \usepackage[a5paper, landscape, twocolumn, twoside,
```

```
%    left=2cm, hmarginratio=2:1, includemp, marginparwidth=43pt,
%    bottom=1cm, foot=.7cm, includefoot, textheight=11cm, heightrounded,
%    columnsep=1cm,verbose]{geometry}
%---------------------------------------------------------------
% No need to change below
%---------------------------------------------------------------
\geometry{verbose,showframe}% options appended.
\newcommand\mynote{\marginpar%
[\raggedright\rule{\marginparwidth}{.7pt}\\A left side note.]%
{\raggedright\rule{\marginparwidth}{.7pt}\\A side note.}}%
\def\fox{A quick brown fox jumps over the lazy dog. }
\def\fivefoxes{\fox\fox\fox\fox\fox}
\def\manyfoxes{\fivefoxes\mynote\fivefoxes\par\fivefoxes\fivefoxes\par}
% \let\mynote\relax % removes marginal notes.
\begin{document}
\manyfoxes\manyfoxes\manyfoxes\manyfoxes
\manyfoxes\manyfoxes\manyfoxes\manyfoxes
\manyfoxes\manyfoxes\manyfoxes\manyfoxes
\end{document}
%SAVE_INTACT
```
⟨/samples⟩