The geometry package

Hideo Umeki
latexgeometry@gmail.com

2018/04/16 v5.8

Abstract

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

1 Preface to version 5

- Changing page layout mid-document.

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

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

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

- A new driver option xetex.

  The new driver option xetex is added. The driver auto-detection routine has been revised so as to avoid an error with undefined control sequences. Note that ‘geometry.cfg’ in T\LaTeX Live, which disables the auto-detection routine and sets pdftex, is no longer necessary and has no problem even though it still exists. To set xetex is strongly recommended with Xe\LaTeX.

- New paper size presets for JIS B-series and ISO C-series.

  The papersize presets b0j to b6j for JIS (Japanese Industrial Standards) B-series and c0paper to c6paper for ISO C-series (v5.4∼) are added.

- Changing default for underspecified margin.

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

- The option showframe and showcrop works on every page.

  With showframe option, the page frames are shown on every page. In addition, a new option showcrop prints crop marks at each corner of layout area on every page. Note that the marks would be invisible without specifying the layout size smaller than paper size. Version 5.4 introduced a new \shipout overloading process using atbegshi package, so the atbegshi package is required when showframe or showcrop option is specified.

- Loading geometry.cfg precedes processing class options.

  The previous version loaded geometry.cfg after processing the document class options. Now that the config file is loaded before processing the class options, you can change the behavior specified in geometry.cfg by adding options into \documentclass as well as \usepackage and \geometry.
• **Deleted options:** compat2 and twosideshift. The version 5 has no longer compatibility with the previous ones. compat2 and twosideshift are gone for simplicity.

## 2 Introduction

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

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

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

\begin{verbatim}
\usepackage[text={7in,10in},centering]{geometry}.
\end{verbatim}

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 to 1.5in, you can type

\begin{verbatim}
\usepackage[margin=1.5in]{geometry}.
\end{verbatim}

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

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

In this case, using geometry you can type

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

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

## 3 Page geometry

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

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

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

\begin{align*}
\text{paperwidth} &= \text{left} + \text{width} + \text{right} \quad (1) \\
\text{paperheight} &= \text{top} + \text{height} + \text{bottom} \quad (2)
\end{align*}
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.

(a) default
(b) includehead and includefoot

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

The total body width and height would be defined:

\[
\text{width} := \text{textwidth} + (\text{marginparsep} + \text{marginparwidth}) \tag{3}
\]
\[
\text{height} := \text{textheight} + (\text{headheight} + \text{headsep} + \text{footskip}) \tag{4}
\]

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

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

\[
L = a + b + c \tag{5}
\]

The specification with two of the three (a, b and c) fixed explicitly is solvable. If two or more are left unspecified or ‘underspecified’, Equation (5) cannot be solved without any other relation between them. If all of them are specified, then it needs to check whether or not they satisfy Equation (5), that is too much specification or ‘overspecified’.
The `geometry` package has auto-completion mechanism that saves the trouble of specifying the page layout dimensions. For example, you can set

\usepackage[width=14cm, left=3cm]{geometry}

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

4 User interface

4.1 Commands

The `geometry` package provides the following commands:

- \geometry{(options)}
- \newgeometry{(options)} and \restoregeometry
- \savegeometry{(name)} and \loadgeometry{(name)}

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

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

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

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

4.2 Optional argument

The `geometry` package adopts keyval interface ‘(key)=(value)’ for the optional argument to \usepackage, \geometry and \newgeometry.

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

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

For example,

\usepackage[\textwidth=14cm, left=3cm,\hspace{0.8in}, 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. \langle3cm,0.8in\rangle. Note that the order of values in the sub-list is significant. The above setting is also equivalent to the followings:
\usepackage{geometry}
\geometry{height=10in,a5paper,hmargin={3cm,0.8in}}

or

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

Thus, multiple use of \geometry just appends options.
\geometry supports package calc\(^1\). For example,

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

4.3 Option types

\geometry options are categorized into four types:

1. **Boolean type**
   takes a boolean value (true or false). If no value, true is set by default.

   \langle key\rangle=true \mid false.
   \langle key\rangle with no value is equivalent to \langle key\rangle=true.

   Examples: \texttt{verbose=true}, \texttt{includehead}, \texttt{twoside=false}.

2. **Single-valued type**
   takes a mandatory value.

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

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

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

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

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

4. **Triple-valued type**
   takes three mandatory, comma-separated values in braces.

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

   Each value must be a dimension or null. When you give an empty value or ‘*’, it means null and
   leaves the appropriate value to the auto-completion mechanism. You need to specify at least one
   dimension, typically two dimensions. You can set nulls for all the values, but it makes no sense.

   Examples:
   \texttt{hdivide={2cm,*,1cm}}, \texttt{vdivide={3cm,19cm}}, \texttt{divide={1in,*,1in}}.

5 Option details

This section describes all options available in \geometry. Options with a dagger \(\dagger\) are not available as arguments of \texttt{newgeometry} (See Section 7).

\(^1\)CTAN: macros/latex/required/tools
5.1 Paper size

The options below set paper/media size and orientation.

† paper | papername

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

† a0paper, a1paper, a2paper, a3paper, a4paper, a5paper, a6paper,
b0paper, b1paper, b2paper, b3paper, b4paper, b5paper, b6paper,
c0paper, c1paper, c2paper, c3paper, c4paper, c5paper, c6paper,
b0j, b1j, b2j, b3j, b4j, b5j, b6j,
ansiapaper, ansibpaper, ansicpaper, ansidpaper, ansiepaper,
letterpaper, executivepaper, legalpaper

  specifies paper name. The value part is ignored even if any. For example, the followings
  have the same effect:
  a5paper,
a5paper=true,
a5paper=false

  and so forth.

a[0-6]paper, b[0-6]paper and c[0-6]paper are ISO A, B and C series of paper sizes
respectively. The JIS (Japanese Industrial Standards) A-series is identical to the ISO
A-series, but the JIS B-series is different from the ISO B-series. b[0-6]j should be used
for the JIS B-series.

† screen

  a special paper size with (W,H) = (225mm,180mm). For presentation with PC and
  video projector, “screen,centering” with ‘slide’ documentclass would be useful.

† paperwidth

  width of the paper. paperwidth=(length).

† paperheight

  height of the paper. paperheight=(length).

† papersize

  width and height of the paper. papersize={(width),(height)} or papersize=(length).

† landscape

  switches the paper orientation to landscape mode.

† portrait

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

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

5.2 Layout size

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

layout

  specifies the layout size by paper name. layout=(paper-name). All the paper names
defined in geometry are available. See Section 5.1 for details.

layoutwidth

  width of the layout. layoutwidth=(length).

layoutheight

  height of the layout. layoutheight=(length).

layoutsizer

  width and height of the layout. layoutsizer={(width),(height)} or
  layoutsize=(length).

layoutoffset

  specifies the horizontal offset from the left edge of the paper. layoutoffset=(length).

layoutoffset

  specifies the vertical offset from the top edge of the paper. layoutoffset=(length).

layoutoffset

  specifies both horizontal and vertical offsets. layoutoffset={(hoffset),(voffset)} or
  layoutsize=(length).

5.3 Body size

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

hscale

  ratio of width of total body to \paperwidth. hscale=(h-scale), e.g., hscale=0.8
  is equivalent to width=0.8\paperwidth. (0.7 by default)
Figure 4: The dimensions related to the layout size. Note that the layout size defaults to the same size as the paper, so you don’t have to specify layout-related options explicitly in most cases.

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=\left{(h-scale),(v-scale)\right)} or scale=scale. (0.7 by default)

width | totalwidth width of total body. width=\left{length\right) or totalwidth=\left{length\right). This dimension defaults to textwidth, but if includemp is set to true, width ≥ textwidth because width includes the width of the marginal notes. If textwidth and width are specified at the same time, textwidth takes priority over width.

height | totalheight height of total body, excluding header and footer by default. If includehead or includefoot is set, height includes the head or foot of the page as well as textheight. height=\left{length\right) or totalheight=\left{length\right). If both textheight and height are specified, height will be ignored.

total width and height of total body.

total=\left{\left{width\right),\left{height\right)\right) or total=\left{length\right).

textwidth specifies \textwidth, the width of body (the text area). textwidth=\left{length\right).

textheight specifies \textheight, the height of body (the text area). textheight=\left{length\right).

body specifies both \textwidth and \textheight of the body of page.

body=\left{\left{width\right),\left{height\right)\right) or text=\left{length\right).

lines enables users to specify \textheight by the number of lines. lines=\left{integer\right).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5.4 Margin size

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

<table>
<thead>
<tr>
<th>left</th>
<th>lmargin</th>
<th>inner</th>
</tr>
</thead>
<tbody>
<tr>
<td>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=\langle \text{length} \rangle. inner has no special meaning, just an alias of left and lmargin.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>right</th>
<th>rmargin</th>
<th>outer</th>
</tr>
</thead>
<tbody>
<tr>
<td>right or outer margin of total body. right=\langle \text{length} \rangle.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>top</th>
<th>tmargin</th>
</tr>
</thead>
<tbody>
<tr>
<td>top margin of the page. top=\langle \text{length} \rangle. Note this option has nothing to do with the native dimension \topmargin.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bottom</th>
<th>bmargin</th>
</tr>
</thead>
<tbody>
<tr>
<td>bottom margin of the page. bottom=\langle \text{length} \rangle.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>hmargin</th>
</tr>
</thead>
<tbody>
<tr>
<td>left and right margin. hmargin={(left , margin), \langle right , margin \rangle} or hmargin=\langle \text{length} \rangle.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>vmargin</th>
</tr>
</thead>
<tbody>
<tr>
<td>top and bottom margin. vmargin={(top , margin), \langle bottom , margin \rangle} or vmargin=\langle \text{length} \rangle.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>hmarginratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>horizontal margin ratio of left (inner) to right (outer). The value of \langle \text{ratio} \rangle 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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>vmarginratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>vertical margin ratio of top to bottom. The default ratio is 2:3.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>marginratio</th>
<th>ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>horizontal and vertical margin ratios. marginratio={(horizontal , ratio), \langle vertical , ratio \rangle} or marginratio=\langle \text{ratio} \rangle.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>hcentering</th>
</tr>
</thead>
<tbody>
<tr>
<td>sets auto-centering horizontally and is equivalent to hmarginratio=1:1. It is set to true by default for oneside. See also hmarginratio.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>vcentering</th>
</tr>
</thead>
<tbody>
<tr>
<td>sets auto-centering vertically and is equivalent to vmarginratio=1:1. The default is false. See also vmarginratio.</td>
</tr>
</tbody>
</table>

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 \oddsidemargin + \evensidemargin 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=\langle \text{length} \rangle. This is useful if pages are bound by a press binding (glued, stitched, stapled . . . ). See Figure 6.

hdivide, vdivide, divide  See description in Section 5.3.

5.5 Native dimensions

The options below overwrite \LaTeX native dimensions and switches for page layout (See the right-hand side in Figure 1).
Figure 6: The option \texttt{bindingoffset} adds the specified length to the inner margin. Note that \texttt{twoside} option swaps the horizontal margins and the marginal notes together with \texttt{bindingoffset} on even pages (see b)), but \texttt{asymmetric} option suppresses the swap of the margins and marginal notes (but \texttt{bindingoffset} is still swapped).

headheight | head modifies \texttt{\headheight}, height of header. \texttt{\headheight=⟨length⟩} or \texttt{\head=⟨length⟩}.

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

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

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

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

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

footnoteseq changes the dimension \texttt{\skip\footins}, separation between the bottom of text body and the top of footnote text.

marginparwidth | marginpar modifies \texttt{\marginparwidth}, width of the marginal notes. \texttt{\marginparwidth=⟨length⟩}.

marginparsep modifies \texttt{\marginparsep}, separation between body and marginal notes. \texttt{\marginparsep=⟨length⟩}.

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

columnsep modifies \texttt{\columnsep}, the separation between two columns in \texttt{twocolumn} mode.

hoffset modifies \texttt{\hoffset}. \texttt{\hoffset=⟨length⟩}.

voffset modifies \texttt{\voffset}. \texttt{\voffset=⟨length⟩}.

offset horizontal and vertical offset. \texttt{\offset={⟨hoffset⟩,⟨voffset⟩}} or \texttt{\offset=⟨length⟩}.

twocolumn sets \texttt{twocolumn} mode with \texttt{@twocolumntrue}. \texttt{twocolumn=false} denotes \texttt{onecolumn} mode with \texttt{@twocolumnfalse}. Instead of \texttt{twocolumn=false}, you can specify \texttt{onecolumn} (which defaults to \texttt{true})

onecolumn works as \texttt{twocolumn=false}. On the other hand, \texttt{onecolumn=false} is equivalent to \texttt{twocolumn}.

twoside sets both \texttt{@twosidetrue} and \texttt{@mparswitchtrue}. See Section 5.4.

textheight sets \texttt{\textheight} directly. See Section 5.3.

reversemp | reversemarginpar makes the marginal notes appear in the left (inner) margin with \texttt{@reversemargintrue}. The option doesn’t change \texttt{includemp} mode. It’s set \texttt{false} by default.
5.6 Drivers

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

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

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

\textit{† dvipdfm} works like dvips except for landscape correction. You can set this condition when using dvipdfm and xdvipdfmx to process the dvi output.

\textit{† pdftex} sets \texttt{\pdfpagewidth} and \texttt{\pdfpageheight} internally.

\textit{† luatex} sets \texttt{\pagewidth} and \texttt{\pageheight} internally.

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

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

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

5.7 Other options

The other useful options are described here.

\textit{† verbose} displays the parameter results on the terminal. verbose=false (default) still puts them into the log file.

\textit{† 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

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

with \texttt{\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.

\textit{† mag} sets magnification value (\texttt{\mag}) and automatically modifies \texttt{\hoffset} and \texttt{\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 (=√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 is order-independent and 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.

† showcrop prints crop marks at each corner of user-specified layout area.

6 Processing options

6.1 Order of loading

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

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

1. geometry.cfg if it exists.
2. Options specified with \documentclass[⟨options⟩]{...}.
3. Options specified with \usepackage[⟨options⟩]{geometry}
4. Options specified with \geometry{⟨options⟩}, which can be called multiple times. (reset option will cancel the specified options ever given in \usepackage{geometry} or \geometry.)

6.2 Order of options

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

\[\text{[left=2cm, right=3cm]}\]

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

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

\[\text{[verbose=true, verbose=false]}\]

results in \text{verbose=false}.

\[\text{[hmargin={3cm,2cm}, left=1cm]}\]

is the same as \text{hmargin={1cm,2cm}}, where the left (or inner) margin is overwritten by \text{left=1cm}.

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

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

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

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

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

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

\[
\text{priority: low} \quad \rightarrow \quad \text{high}
\]

\[
\begin{cases}
\text{hscale} \\
\text{vscale} \\
\text{scale}
\end{cases}
<
\begin{cases}
\text{width} \\
\text{height} \\
\text{total}
\end{cases}
<
\begin{cases}
\text{textwidth} \\
\text{textheight} \\
\text{text}
\end{cases}
<
\text{lines}.
\]

For example,

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

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

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

results in \texttt{[lines=30, textwidth=7in]}.

6.4 Defaults

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

The default vertical margin ratio is \(2/3\), namely,

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

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

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

Obviously the default horizontal margin ratio for oneside is ‘centering’.

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

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

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

Additional options overwrite the previous specified dimensions.

6.5 Auto-completion

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

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

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

\(S(0,0)\) Nothing is specified. The geometry package sets \texttt{body} with the default \texttt{scale} (= 0.7).

For example, \texttt{width} is set to be \(0.7 \times \texttt{layoutwidth}\). Note that by default \texttt{layoutwidth} and \texttt{layoutheight} will be equal to \texttt{\paperwidth} and \texttt{\paperheight} respectively. Thus \(S(0,0)\) goes to \(S(0,1)\). See \(S(0,1)\).
Figure 7: Specifications $S(0,0)$ to $S(2,1)$ and the completion rules (arrows). Column and row numbers denote the number of explicitly specified lengths for margin and body respectively. $S(m,b)$ denote a specification with a set of the numbers $(\text{margin}, \text{body}) = (m, b)$.

$S(0,1)$ Only \textit{body} is specified, such as \textit{width}=7in, \textit{lines}=20, \textit{body}={20cm,24cm}, \textit{scale}=0.9 and so forth. Then \texttt{geometry} sets margins with the margin ratio. If the margin ratio is not specified, the default is used. The default vertical margin ratio is defined as $\text{top} : \text{bottom} = 2 : 3$ default. (8)

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

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

For example, if \textit{height}=22cm is specified on A4 paper, \texttt{geometry} calculates \textit{top} margin as follows:

$$\text{top} = (\text{layoutheight} - \text{height}) \times 2/5$$
$$= (29.7 - 22) \times 2/5 = 3.08(\text{cm})$$ (10)

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

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

- \textbf{If the margin ratio is not specified}, \texttt{geometry} sets \textit{body} with the default \texttt{scale} (= 0.7). For example, if \textit{top}=2.4cm is specified, \texttt{geometry} sets \textit{height} = 0.7 $\times$ layoutheight (= 0.7 \texttt{paperheight} by default), then $S(1,0)$ goes to $S(1,1)$, in which \textit{bottom} is calculated with \textit{layoutheight} – (\textit{height} + \textit{top}) and results in 6.51cm on A4 paper if the layout size is equal to the paper size.

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

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

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

Note that the version 4 or earlier used to set the other margin with the margin ratio. In the version 5, therefore, with the same specification, the result will be different from the one in the version 4. For example, if only \textit{top}=2.4cm is specified, you got \textit{bottom}=2.4cm in the version 4 or earlier, but you will get \textit{bottom}=6.51cm in the version 5.
The `body` and two `margins` are all specified, such as `vdivide={1in,8in,1.5in}`, “left=3cm,width=13cm,right=4cm” and so forth. Since `geometry` basically gives priority to `margins` if dimensions are overspecified, `geometry` forgets and resets `body`. For example, if you specify

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

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

## 7 Changing layout mid-document

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

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

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

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

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

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

\begin{verbatim}
\usepackage[hmargin=3cm]{geometry}
\begin{document}
  L1
  \newgeometry{left=3cm,right=1cm,bottom=0.1cm}
  \savegeometry{L2}
    L2 (new, saved)
\end{document}
\end{verbatim}
8 Examples

1. 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

2. A twosided page layout with the inside offset for binding set to 1cm.
   - twoside, bindingoffset=1cm
   
   In this case, textwidth is shorter than that of the default twosided document by $0.7 \times 1\text{cm} (= 0.7\text{cm})$ because the default width of body is set with scale=0.7, which means $\text{width} = 0.7 \times \text{layoutwidth}$ (= $0.7\times\text{paperwidth}$ by default).

3. 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

4. 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, $\text{height} = \text{textheight} + \text{footskip}$ when includefoot=true in this case.
   - bmargin=2cm, height=10in, includefoot
   - bottom=2cm, textheight=10in, includefoot

5. A layout with textwidth and textheight 90% of the paper and with body centered. Each solution below results in the same page layout as long as layoutwidth and layoutheight are not modified from the default.
   - scale=0.9, centering
   - text={.9\paperwidth,.9\paperheight}, ratio=1:1
   - width=.9\paperwidth, vmargin=.05\paperheight, marginratio=1:1
   - hdivide=\*,.9\paperwidth,\*, vdivide=\*,.9\paperheight,\* (as for onesided documents)
   - margin=0.05\paperwidth,0.05\paperheight

   You can add heightrounded to avoid an “underfull vbox warning” like
   
   Underfull \vbox (badness 10000) has occurred while \output is active.
See Section 5.3 for the detailed description about `heightrounded`.

6. A layout with the width of marginal notes set to 3 cm and included in the width of total body. The following examples are the same.
   - `marginparwidth=3cm, includemp`
   - `marginpar=3cm, ignoremp=false`

7. A layout where body occupies the whole paper with A5 paper in landscape. The following examples are the same.
   - `a5paper, landscape, scale=1.0`
   - `landscape=TRUE, paper=a5paper, margin=0pt`

8. A screen size layout appropriate for presentation with PC and video projector.

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

9. A layout with fonts and spaces both enlarged from A4 to A3. In the case below, the resulting 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 type
   - `letterpaper, mag=2000, truedimen`

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

10. Changing the layout of the first page and leaving the others as default before loading `geometry`. Use `pass` option, `\newgeometry` and `\restoregeometry`.

    \documentclass{book}
    \usepackage[pass]{geometry}
    % 'pass' disregards the package layout,
    % so the original 'book' layout is memorized here.
    \begin{document}
    \newgeometry{margin=1cm} % changes the first page dimensions.
    \restoregeometry % restores the original 'book' layout.
    \end{document}

11. A complex page layout.

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

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

9 Known problems

- With `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 `mag \neq 1000`, no `truedimen` and `hyperref`, `hyperref` should be loaded before `geometry`. Otherwise the resulted PDF size will become wrong.

- With `crop` package and `mag \neq 1000`, `center` option of `crop` doesn’t work well.
10 Acknowledgments

11 Implementation

This package requires the following packages: keyval, ifpdf, ifvtex and ifxetex.

\RequirePackage{keyval}\
\RequirePackage{ifpdf}\
\RequirePackage{ifvtex}\
\RequirePackage{ifxetex}

Internal switches are declared here.

\newif\ifGm@verbose
\newif\ifGm@landscape
\newif\ifGm@swap@papersize
\newif\ifGm@includehead
\newif\ifGm@includefoot
\newif\ifGm@includemp
\newif\ifGm@hbody
\newif\ifGm@vbody
\newif\ifGm@heightrounded
\newif\ifGm@showframe
\newif\ifGm@pass
\newif\ifGm@resetpaper
\newif\ifGm@layout
\newif\ifGm@newgm
\Gm@cnth The counters for horizontal and vertical partitioning patterns.
\Gm@cntv
\c@Gm@tempcnt The counter is used to set number with calc.
\Gm@bindingoffset The binding offset for the inner margin.
\Gm@wd@mp \Gm@odd@mp \Gm@even@mp Correction lengths for \textwidth, \oddsidemargin and \evensidemargin in includemp mode.
\Gm@layoutwidth \Gm@layoutheight \Gm@layouthoffset \Gm@layoutvoffset The dimensions for the layout area.
\Gm@dimlist The token in which \LaTeX native dimensions can be stored.
\Gm@warning The macro to print warning messages.
\ifGm@preamble The macro executes the option given as an argument only if it’s specified in the preamble, as the options of \usepackage and/or the argument of \geometry. Otherwise, the macro would print the warning message and ignores the option setting.
The default values for the horizontal and vertical *marginalratio* are defined. `\Gm@Dhratio` denotes the default value of horizontal *marginalratio* for twoside page layout with left and right margins swapped on verso pages, which is set by `twoside`.

\[
\begin{align*}
\def\Gm@Dhratio{1:1}\% &= \text{left:right default for oneside} \\
\def\Gm@Dhratiotwo{2:3}\% &= \text{inner:outer default for twoside}.
\end{align*}
\]

The default values for the horizontal and vertical *scale* are defined with 0.7.

\[
\begin{align*}
\def\Gm@Dhscale{0.7}\% \\
\def\Gm@Dvscale{0.7}\%
\end{align*}
\]

The driver names.

\[
\begin{align*}
\def\Gm@dvips{dvips}\% \\
\def\Gm@dvipdfm{dvipdfm}\% \\
\def\Gm@pdftex{pdftex}\% \\
\def\Gm@luatex{luatex}\% \\
\def\Gm@xetex{xetex}\% \\
\def\Gm@vtex{vtex}\%
\end{align*}
\]

The macros for *true* and *false*.

\[
\begin{align*}
\def\Gm@true{true}\% \\
\def\Gm@false{false}\%
\end{align*}
\]

These macros keep original paper (media) size intact.

\[
\begin{align*}
\edef\Gm@orgpw{\the\paperwidth}\% \\
\edef\Gm@orgph{\the\paperheight}\%
\end{align*}
\]

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

\[
\begin{align*}
\def\Gm@savelength#1{\% \\
\g@addto@macro\Gm@restore{\expandafter\noexpand\expandafter\csname #1\endcsname=\the\csname #1\endcsname\relax}}\%
\end{align*}
\]

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

\[
\begin{align*}
\def\Gm@saveboolean#1{\% \\
\csname if#1\endcsname \g@addto@macro\Gm@restore{\expandafter\noexpand\csname #1true\endcsname} \else \g@addto@macro\Gm@restore{\expandafter\noexpand\csname #1false\endcsname} \fi}\%
\end{align*}
\]

The initialization for \Gm@restore.

\[
\begin{align*}
\def\Gm@restore{}\%
\end{align*}
\]

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

\[
\begin{align*}
\def\Gm@save{\% \\
\Gm@savelength{paperwidth}\% \\
\Gm@savelength{paperheight}\% \\
\Gm@savelength{textwidth}\% \\
\Gm@savelength{textheight}\% \\
\Gm@savelength{evensidemargin}\% \\
\Gm@savelength{oddsidemargin}\% \\
\Gm@savelength{topmargin}\% \\
\Gm@savelength{headheight}\% \\
\Gm@savelength{headsep}\% \\
\Gm@savelength{topskip}\% \\
\Gm@savelength{footskip}\% \\
\Gm@savelength{baselineskip}\% \\
\Gm@savelength{marginparwidth}\% \\
\Gm@savelength{marginparsep}\% \\
\Gm@savelength{columnsep}\% \\
\Gm@savelength{hoffset}\% \\
\Gm@savelength{voffset}\% \\
\Gm@savelength{Gm@layoutwidth}\% \\
\Gm@savelength{Gm@layoutheight}\%
\end{align*}
\]
The macro initializes the parameters for layout in `\newgeometry`.

\verbatim
\Gm@initnewgm
\def\Gm@initnewgm{% 
\Gm@passfalse
\Gm@swap@papersizefalse
\Gm@dimlist={} 
\Gm@hbodyfalse
\Gm@vbodyfalse
\Gm@heightroundedfalse
\Gm@includeheadfalse
\Gm@includefootfalse
\Gm@includempfalse
\let\Gm@width\@undefined
\let\Gm@height\@undefined
\let\Gm@textwidth\@undefined
\let\Gm@textheight\@undefined
\let\Gm@lines\@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
\Gm@layoutfalse
\Gm@layouthoffset\z@
\Gm@layoutvoffset\z@}

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

\verbatim
\Gm@initall
\def\Gm@initall{% 
\let\Gm@driver\@empty
\let\Gm@truedimen\@empty
\let\Gm@paper\@undefined
\Gm@resetpaperfalse
\Gm@landscapefalse
\Gm@verbosefalse
\Gm@showframefalse
\Gm@showcropfalse
\Gm@newgmfalse
\Gm@initnewgm}
\Gm@setdriver The macro sets the specified driver.

\verbatim
\Gm@setdriver
\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.

\verbatim
\Gm@unsetdriver
\def\Gm@unsetdriver#1{% 
\expandafter\ifx\csname Gm@#1\endcsname\Gm@driver\let\Gm@driver\@empty\fi}
\Gm@setbool The macros for boolean option processing.

\verbatim
\Gm@setbool \Gm@setboolrev
\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}}
\Gm@doif \Gm@doif execs 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 \textit{true}, and executes the fourth argument \#4 if false.

\begin{verbatim}
138 \def\Gm@doif\#1\#2\#3{\
139 \lowercase{\def\Gm@bool{\#2}}\
140 \ifx\Gm@bool\@empty\
141 \let\Gm@bool\Gm@true\
142 \fi\
143 \ifx\Gm@bool\Gm@true\
144 \else\
145 \ifx\Gm@bool\Gm@false\
146 \else\
147 \let\Gm@bool\relax\
148 \fi\
149 \fi}\
150 \def\Gm@doifelse\#1\#2\#3\#4{\
151 \Gm@doif\{\#1\}\{\#2\}{\ifx\Gm@bool\Gm@true \#3\else \#4\fi}}\
\end{verbatim}

\Gm@reverse The macro reverses a bool value.

\begin{verbatim}
157 \def\Gm@reverse\#1{\
158 \csname ifGm@#1\endcsname\
159 \csname Gm@#1false\endcsname\else\csname Gm@#1true\endcsname\fi}\
\end{verbatim}

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

\begin{verbatim}
160 \def\Gm@defbylen\#1\#2{\
161 \begingroup\setlength\@tempdima{\#2}\
162 \expandafter\xdef\csname Gm@#1\endcsname{\the\@tempdima}\endgroup}\
163 \def\Gm@defbycnt\#1\#2{\
164 \begingroup\setcounter{Gm@tempcnt}{\#2}\
165 \expandafter\xdef\csname Gm@#1\endcsname{\the\value{Gm@tempcnt}}\endgroup}\
\end{verbatim}

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

\begin{verbatim}
166 \def\Gm@set@ratio\#1\#2\#3\#4{\determine \#4 by ratio}\
167 \expandafter\setlength\expandafter\@tempdima\expandafter{\csname Gm@layout\#1\endcsname}\
168 \if\#1b\
169 \edef\@tempa{\the\@tempdima}\
170 \@tempdima=\@temptcb\
171 \@tempcntb=\@tempa\relax\
172 \fi\
173 \expandafter\setlength\expandafter\@tempdima\expandafter{\the\@tempdima}\
174 \ifnum\@tempcntb>\z@\
175 \multiply\@tempdima\@tempcnta\
176 \divide\@tempdima\@tempcntb\
177 \fi\
178 \edef\csname Gm@#4\endcsname{\the\@tempdima}}\
\end{verbatim}

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

\begin{verbatim}
181 \def\Gm@detiv\#1\#2\#3\#4{\determine \#4 by \texttt{calc}}\
182 \expandafter\setlength\expandafter\@tempdima\expandafter{\csname Gm@layout\#1\endcsname}\
183 \if\#1b\
184 \edef\@tempa{\the\@tempdima}\
185 \@tempdima=\@tempcntb\
186 \@tempcntb=\@tempa\relax\
187 \fi\
188 \expandafter\edef\csname Gm@#4\endcsname{\the\@tempdima}}\
\end{verbatim}
\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}}\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 Gm@layout#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\expandafter\Gm@sep@ratio\Gm@Dmratio\relax\else\expandafter\Gm@sep@ratio\Gm@mratio\relax\fi\ifnum\@tempcntb>\z@\else\Gm@warning{margin ratio a:b should be non-zero; default used}\expandafter\Gm@sep@ratio\Gm@Dmratio\relax\fi\
  \@tempdimb=\@tempdima\advance\@tempcntb\@tempcnta\divide\@tempdima\@tempcntb\multiply\@tempdima\@tempcnta\advance\@tempdimb-\@tempdima\expandafter\edef\csname Gm@#2\endcsname{\the\@tempdima}\
  \expandafter\edef\csname Gm@#3\endcsname{\the\@tempdimb}}\Gm@detall

\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{% determine \#2 and \#3.
  \@tempcnta=\z@\if#1h\let\Gm@mratio\Gm@hmarginratio\edef\Gm@Dmratio{\if@twoside\Gm@Dhratiotwo\else\Gm@Dratio\fi}\else\let\Gm@mratio\Gm@vmarginratio\edef\Gm@Dmratio{\Gm@Dratio}\fi\@tempcnta is treated as a three-digit binary value with top, middle and bottom denoted left(top), width(height) and right(bottom) margins user specified respectively.
  \if#1h\ifx\Gm@lmargin\@undefined\else\advance\@tempcnta4\relax\fi\else\ifx\Gm@tmargin\@undefined\else\advance\@tempcnta4\relax\fi\g@cntv\@tempcnta\else\ifx\Gm@bmargin\@undefined\else\advance\@tempcnta1\relax\fi\else\ifx\Gm@cmargin\@undefined\else\advance\@tempcnta1\relax\fi\Gm@c@cntv\@tempcnta\else\ifx\Gm@cmargin\@undefined\else\advance\@tempcnta1\relax\fi\else\ifx\Gm@cmargin\@undefined\else\advance\@tempcnta1\relax\fi\Gm@c@cntv\@tempcnta\fi
Case the value is 000 (=0) with nothing fixed (default):

\ifcase\@tempcnta
\if#1h
\Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}\
\else
\Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}\
\fi
\fi
\Gm@detiandiii{#2}{#3}{#4}\

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

\or\Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}\
\else\Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}\
\fi
\setlength\@tempdimc{\@nameuse{Gm@#4}}\
\Gm@detiiandiii{#2}{#3}{#4}\
\expandafter\let\csname Gm@#2\endcsname\@undefined\
\Gm@defbylen{#4}{\@tempdimc}\
\else\Gm@setbyratio[f]{#1}{#4}{#3}\
\Gm@detiv{#2}{#3}{#4}{#2}\
\fi

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

\or\Gm@detiandiii{#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@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}\
\else\Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}\
\fi
\setlength\@tempdimc{\@nameuse{Gm@#3}}\
\Gm@detiiandiii{#2}{#3}{#4}\
\expandafter\let\csname Gm@#2\endcsname\@undefined\
\Gm@defbylen{#3}{\@tempdimc}\
\else\Gm@setbyratio[b]{#1}{#3}{#4}\
\fi
\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.\
~~~"}\@spaces ‘#2’ (\@nameuse{Gm@#2}) is ignored}\
\Gm@detiv{#2}{#3}{#4}{#2}\
\else\fi\

\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\if\Gm@hbody\else\fi}
The macro parses \(h,v\)divide options.

\[\text{\texttt{\textbackslash Gm@parse@divide}}\]

\[\text{\texttt{\textbackslash Gm@branch}}\]

This macro is used to adjust offsets by \(\text{mag}\).

\[\text{\texttt{\textbackslash Gm@magtooffset}}\]

This macro stores \(\text{\LaTeX}\) native dimensions, which are stored and set afterwards.

\[\text{\texttt{\textbackslash Gm@setlength}}\]

This macro processes \(\text{Gm@dimlist}\).
The macro sets \texttt{paperwidth} and \texttt{paperheight} dimensions using \texttt{\Gm@setlength} macro.

```latex
\Gm@setsize The macro changes the paper size.
```

```latex
\Gm@setpaper@ifpre The macro sets \texttt{paperwidth} and \texttt{paperheight} dimensions using \texttt{\Gm@setlength} macro.
```

```latex
\begin{itemize}
  \item \texttt{a[0-6]paper}
  \item \texttt{b[0-6]paper}
  \item \texttt{ansi[a-e]paper}
  \item \texttt{letterpaper}
  \item \texttt{legalpaper}
  \item \texttt{executivepaper}
  \item \texttt{screen}
\end{itemize}
```

\texttt{paper}\ paper takes a paper name as its value.

```latex
\begin{itemize}
  \item \texttt{a0paper}
  \item \texttt{a1paper}
  \item \texttt{a2paper}
  \item \texttt{a3paper}
  \item \texttt{a4paper}
  \item \texttt{a5paper}
  \item \texttt{a6paper}
  \item \texttt{b0paper}
  \item \texttt{b1paper}
  \item \texttt{b2paper}
  \item \texttt{b3paper}
  \item \texttt{b4paper}
  \item \texttt{b5paper}
  \item \texttt{b6paper}
  \item \texttt{b0j}
  \item \texttt{b1j}
  \item \texttt{b2j}
  \item \texttt{b3j}
  \item \texttt{b4j}
  \item \texttt{b5j}
  \item \texttt{b6j}
  \item \texttt{ansiapaper}
  \item \texttt{ansibpaper}
  \item \texttt{ansicpaper}
  \item \texttt{ansidpaper}
  \item \texttt{ansiepaper}
  \item \texttt{letterpaper}
  \item \texttt{legalpaper}
  \item \texttt{executivepaper}
  \item \texttt{screen}
\end{itemize}
```

\texttt{papername} paper takes a paper name as its value.

---

1. \texttt{\def\Gm@td{\Gm@truedimen}}
2. \texttt{\def\Gm@len##1##2{\setlength{##1}{##2}}}  
3. \texttt{\the\Gm@dimlist}
4. \texttt{\Gm@setsize}

The following paper names are available.

1. \texttt{\def\Gm@setpaper@ifpre{\Gm@setpaper@ifpre}}
2. \texttt{\if\Gm@preamble{\Gm@setpaper@ifpre{\Gm@setpaper@ifpre}}

Various paper size are defined here.

1. \texttt{\Gm@setsize{a0paper}(841,1189){mm}}
2. \texttt{\Gm@setsize{a1paper}(594,841){mm}}
3. \texttt{\Gm@setsize{a2paper}(420,594){mm}}
4. \texttt{\Gm@setsize{a3paper}(297,420){mm}}
5. \texttt{\Gm@setsize{a4paper}(210,297){mm}}
6. \texttt{\Gm@setsize{a5paper}(148,210){mm}}
7. \texttt{\Gm@setsize{a6paper}(105,148){mm}}
8. \texttt{\Gm@setsize{b0paper}(1000,1414){mm}}
9. \texttt{\Gm@setsize{b1paper}(707,1000){mm}}
10. \texttt{\Gm@setsize{b2paper}(500,707){mm}}
11. \texttt{\Gm@setsize{b3paper}(353,500){mm}}
12. \texttt{\Gm@setsize{b4paper}(250,353){mm}}
13. \texttt{\Gm@setsize{b5paper}(176,250){mm}}
14. \texttt{\Gm@setsize{b6paper}(125,176){mm}}
15. \texttt{\Gm@setsize{c0paper}(917,1297){mm}}
16. \texttt{\Gm@setsize{c1paper}(648,917){mm}}
17. \texttt{\Gm@setsize{c2paper}(458,648){mm}}
18. \texttt{\Gm@setsize{c3paper}(324,458){mm}}
19. \texttt{\Gm@setsize{c4paper}(229,324){mm}}
20. \texttt{\Gm@setsize{c5paper}(162,229){mm}}
21. \texttt{\Gm@setsize{c6paper}(114,162){mm}}
22. \texttt{\Gm@setsize{b0j}(1030,1456){mm}}
23. \texttt{\Gm@setsize{b1j}(728,1030){mm}}
24. \texttt{\Gm@setsize{b2j}(515,728){mm}}
25. \texttt{\Gm@setsize{b3j}(364,515){mm}}
26. \texttt{\Gm@setsize{b4j}(257,364){mm}}
27. \texttt{\Gm@setsize{b5j}(182,257){mm}}
28. \texttt{\Gm@setsize{b6j}(128,182){mm}}
29. \texttt{\Gm@setsize{ansiapaper}(8.5,11){in}}
30. \texttt{\Gm@setsize{ansibpaper}(11,17){in}}
31. \texttt{\Gm@setsize{ansiicpaper}(17,22){in}}
32. \texttt{\Gm@setsize{ansidpaper}(22,34){in}}
33. \texttt{\Gm@setsize{ansiapaper}(34,44){in}}
34. \texttt{\Gm@setsize{letterpaper}(8.5,11){in}}
35. \texttt{\Gm@setsize{legalpaper}(7.25,10.5){in}}
36. \texttt{\Gm@setsize{executivepaper}(225,180){mm}}

- \texttt{[0-6]paper}
- \texttt{[0-6]paper}
- \texttt{ansi[a-e]paper}
- \texttt{letterpaper}
- \texttt{legalpaper}
- \texttt{executivepaper}
- \texttt{screen}
Direct specification for paper size is also possible.

```
\define@key{Gm}{paperwidth}{\ifGm@preamble{paperwidth}{% 
\def\Gm@paper{custom}\Gm@setlength\paperwidth{#1}}}%
\define@key{Gm}{paperheight}{\ifGm@preamble{paperheight}{% 
\def\Gm@paper{custom}\Gm@setlength\paperheight{#1}}}%
\define@key{Gm}{papersize}{\ifGm@preamble{papersize}{% 
\def\Gm@paper{custom}\Gm@branch{#1}{paperwidth}{paperheight}}}%
```

Paper orientation setting.

```
\define@key{Gm}{landscape}[true]{\ifGm@preamble{landscape}{% 
\Gm@doifelse{landscape}{#1}{% 
{\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}% 
{\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}}}%
\define@key{Gm}{portrait}[true]{\ifGm@preamble{portrait}{% 
\Gm@doifelse{portrait}{#1}{% 
{\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}% 
{\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}}}%
```

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

```
\define@key{Gm}{hscale}{\ifGm@preamble{hscale}{% 
\Gm@hscalefalse(landscape)\#1}}%
\define@key{Gm}{vscale}{\ifGm@preamble{vscale}{% 
\Gm@vscalefalse(landscape)\#1}}%
\define@key{Gm}{scale}{\ifGm@preamble{scale}{% 
\Gm@branch{#1}{hscale}{vscale}}}%
```

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

```
\define@key{Gm}{totalwidth}{\ifGm@preamble{totalwidth}{% 
\Gm@totalwidthfalse(landscape)\#1}}%
\define@key{Gm}{totalheight}{\ifGm@preamble{totalheight}{% 
\Gm@totalheightfalse(landscape)\#1}}%
```
These options directly set the dimensions \textwidth and \textheight. body is an alias of text.

\let\KV@Gm@totalwidth\KV@Gm@width
\let\KV@Gm@totalheight\KV@Gm@height

The option sets \textwidth with the number of lines.

\let\KV@Gm@body\KV@Gm@text

These options exclude head, foot and marginpars when determining body.

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

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

These options round \textheight to n-times of \baselineskip plus \topskip.

The options take the corresponding dimensions as part of body.

These options directly sets the dimensions \textwidth and \textheight.

\let\KV@Gm@body\KV@Gm@text

The option sets \textheight with the number of lines.

\let\KV@Gm@body\KV@Gm@text

These options set \textwidth and \textheight. body is an alias of text.

The options are useful to specify partitioning in each direction of the paper.
These options are shorthands for setting margins.

Options specifying the margin ratios.

Useful shorthands to place body centered.

The direct settings of head and/or foot dimensions.

They are only shorthands to set head and/or foot to be Opt. 

The option directly sets a native dimension \footnotesep.

They directly set native dimensions \marginparwidth and \marginparsep.

The macro is a shorthand for \marginparwidth=Opt and \marginparsep=Opt.

The option sets a native dimension \columnsep.
The former two options set native dimensions `\hoffset` and `\voffset`. `offset` can set both of them with the same value.

543 \define@key{Gm}{hoffset}{\Gm@setlength\hoffset{#1}}%  
544 \define@key{Gm}{voffset}{\Gm@setlength\voffset{#1}}%  
545 \define@key{Gm}{offset}{\Gm@branch{#1}{hoffset}{voffset}}%  

546 \define@key{Gm}{layouthoffset}{\Gm@setlength\Gm@layouthoffset{#1}}%  
547 \define@key{Gm}{layoutvoffset}{\Gm@setlength\Gm@layoutvoffset{#1}}%  
548 \define@key{Gm}{layoutoffset}{\Gm@branch{#1}{layouthoffset}{layoutvoffset}}%  

549 \define@key{Gm}{twocolumn}{true}{%  
550 \Gm@doif{twocolumn}{#1}{\csname @twocolumn\Gm@bool\endcsname}}%  

551 \define@key{Gm}{onecolumn}{true}{%  
552 \Gm@doifelse{onecolumn}{#1}{\@twocolumnfalse}{\@twocolumntrue}}%  

553 \define@key{Gm}{reversemp}{true}{%  
554 \Gm@doif{reversemp}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%  
555 \define@key{Gm}{reversemarginpar}{true}{%  
556 \Gm@doif{reversemarginpar}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%  

557 \define@key{Gm}{driver}{\ifGm@preamble{driver}{%  
558 \edef\@@tempa{#1}\edef\@@auto{auto}\edef\@@none{none}%  
559 \ifx\@@tempa\@empty\let\Gm@driver\relax\else%  
560 \ifx\@@tempa\@@none\let\Gm@driver\relax\else%  
561 \ifx\@@tempa\@@auto\let\Gm@driver\@empty\else%  
562 \setkeys{Gm}{#1}\fi\fi\fi\let\@@auto\relax\let\@@none\relax}}%  

563 \define@key{Gm}{dvips}{true}{%  
564 \Gm@doif{dvips}{#1}{\Gm@setdriver{dvips}}{%  
565 \Gm@unsetdriver{dvips}}}}%  
566 \define@key{Gm}{dvipdfm}{true}{%  
567 \Gm@doif{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{%  
568 \Gm@unsetdriver{dvipdfm}}}}%  
569 \define@key{Gm}{dvipdfmx}{true}{%  
570 \Gm@doif{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{%  
571 \Gm@unsetdriver{dvipdfm}}}}%  
572 \define@key{Gm}{xdvipdfm}{true}{%  
573 \Gm@doif{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{%  
574 \Gm@unsetdriver{dvipdfm}}}}%  
575 \define@key{Gm}{xdvipdfm}{true}{%  
576 \Gm@doif{xdvipdfm}{#1}{\Gm@setdriver{xdvipdfm}}{%  
577 \Gm@unsetdriver{xdvipdfm}}}}%  
578 \define@key{Gm}{pdftex}{true}{%  
579 \Gm@doif{pdftex}{#1}{\Gm@setdriver{pdftex}}{%  
580 \Gm@unsetdriver{pdftex}}}}%  
581 \define@key{Gm}{luatex}{true}{%  
582 \Gm@doif{luatex}{#1}{\Gm@setdriver{luatex}}{%  
583 \Gm@unsetdriver{luatex}}}}%  
584 \define@key{Gm}{xetex}{true}{%  
585 \Gm@doif{xetex}{#1}{\Gm@setdriver{xetex}}{%  
586 \Gm@unsetdriver{xetex}}}}%  
587 \define@key{Gm}{vte}{true}{%  
588 \Gm@doif{vte}{#1}{\Gm@setdriver{vte}}{%  
589 \Gm@unsetdriver{vte}}}}%  

580 \define@key{Gm}{reset}{true}{%  
581 \Gm@doif{reset}{#1}{\Gm@restore@org\Gm@initall}}%  
582 \ProcessOptionsKV[c]{Gm}{\Gm@setdefaultpaper{}}}}%  

The verbose mode.

579 \define@key{Gm}{verbose}{true}{%  
580 \Gm@doif{verbose}{#1}{\Gm@setbool{verbose}{#1}}}}%  

The geometry package supports dvips, dvipdfm, pdflatex, luatex and vte. dvipdfm works like dvips.

563 \define@key{Gm}{dvips}{true}{%  
564 \Gm@doif{dvips}{#1}{\Gm@setdriver{dvips}}{%  
565 \Gm@unsetdriver{dvips}}}}%  
566 \define@key{Gm}{dvipdfm}{true}{%  
567 \Gm@doif{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{%  
568 \Gm@unsetdriver{dvipdfm}}}}%  
569 \define@key{Gm}{xdvipdfm}{true}{%  
570 \Gm@doif{xdvipdfm}{#1}{\Gm@setdriver{xdvipdfm}}{%  
571 \Gm@unsetdriver{xdvipdfm}}}}%  
572 \define@key{Gm}{pdftex}{true}{%  
573 \Gm@doif{pdftex}{#1}{\Gm@setdriver{pdftex}}{%  
574 \Gm@unsetdriver{pdftex}}}}%  
575 \define@key{Gm}{luatex}{true}{%  
576 \Gm@doif{luatex}{#1}{\Gm@setdriver{luatex}}{%  
577 \Gm@unsetdriver{luatex}}}}%  
578 \define@key{Gm}{xetex}{true}{%  
579 \Gm@doif{xetex}{#1}{\Gm@setdriver{xetex}}{%  
580 \Gm@unsetdriver{xetex}}}}%  
581 \define@key{Gm}{vte}{true}{%  
582 \Gm@doif{vte}{#1}{\Gm@setdriver{vte}}{%  
583 \Gm@unsetdriver{vte}}}}%  

580 \define@key{Gm}{reset}{true}{%  
581 \Gm@doif{reset}{#1}{\Gm@restore@org\Gm@initall}}%  
582 \ProcessOptionsKV[c]{Gm}{\Gm@setdefaultpaper{}}}}%  

The option cancels all the options specified before reset, except pass. mag (≠ 1000) with truedimen cannot be also reset.
If \texttt{resetpaper} is set to \texttt{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.

\begin{verbatim}
\define@key{Gm}{resetpaper}{true}{\ifGm@preamble{resetpaper}{% \Gm@setbool{resetpaper}{#1}}}%
\end{verbatim}

\texttt{mag} is expanded immediately when it is specified. So \texttt{reset} can’t reset \texttt{mag} when it is set with \texttt{true}.dimen.

\begin{verbatim}
\define@key{Gm}{mag}{\ifGm@preamble{mag}{\mag=#1}}%
\end{verbatim}

If \texttt{true}dimen is set to \texttt{true}, all of the internal explicit dimensions is changed to \texttt{true} dimensions, e.g., 1in is changed to 1truein.

\begin{verbatim}
\define@key{Gm}{truedimen}{true}{\ifGm@preamble{truedimen}{% \Gm@doifelse{truedimen}{#1}{\let\Gm@truedimen\Gm@true}{}{\let\Gm@truedimen\@empty}}}%
\end{verbatim}

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

\begin{verbatim}
\define@key{Gm}{pass}{true}{\ifGm@preamble{pass}{\Gm@setbool{pass}{#1}}}%
\end{verbatim}

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

\begin{verbatim}
\define@key{Gm}{showframe}{true}{\ifGm@preamble{showframe}{% \Gm@setbool{showframe}{#1}}}%
\end{verbatim}

The showcrop option prints crop marks at each corner of the layout area.

\begin{verbatim}
\define@key{Gm}{showcrop}{true}{\ifGm@preamble{showcrop}{% \Gm@setbool{showcrop}{#1}}}%
\end{verbatim}

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

\begin{verbatim}
\Gm@setdefaultpaper The macro checks whether or not the marginpars overrun the page.
\end{verbatim}

\begin{verbatim}
\Gm@checkmp The macro checks whether or not the marginpars overrun the page.
\end{verbatim}
The macro sets marginpar correction when \texttt{includemp} is set, which is used in \texttt{\Gm@process}. The variables \texttt{\Gm@wd@mp}, \texttt{\Gm@odd@mp} and \texttt{\Gm@even@mp} are set here. Note that \texttt{\Gm@even@mp} should be used only for twoside layout.

\begin{verbatim}
\def\Gm@adjustmp{\%
  \ifGm@includemp\%
    \@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
  \fi
}\%
\end{verbatim}

\subsection*{\texttt{\Gm@adjustbody}}

If the horizontal dimension of \texttt{body} is specified by user, \texttt{\Gm@width} is set properly here.

\begin{verbatim}
\def\Gm@adjustbody{
  \ifGm@hbody\%
    \ifx\Gm@width\@undefined\%
      \ifx\Gm@hscale\@undefined\%
        \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
      \else\%
        \Gm@defbylen{width}{\Gm@hscale\Gm@layoutwidth}%
      \fi
    \else\%
      \Gm@defbylen{width}{\Gm@width}%
    \fi
  \fi
}\%
\end{verbatim}
If the vertical dimension of \textit{body} is specified by user, \texttt{Gm@height} is set properly here.

\begin{verbatim}
\def\Gm@process{% 
  \if\Gm@pass 
    \Gm@restore@org 
  \else 
    \Gm@@process 
  \fi 
}
\end{verbatim}

The main macro processing the specified dimensions is defined.

\begin{verbatim}
\def\Gm@process{% 
  \if\Gm@pass 
    \Gm@restore@org 
  \else 
    \Gm@@process 
  \fi 
}
\end{verbatim}
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\else\evensidemargin{oddsidemargin}\fi

The bindingoffset correction for \oddsidemargin.

\ifGm@includehead\addtolength{oddsidemargin}{\Gm@bindingoffset}\addtolength{oddsidemargin}{-1\Gm@truedimen in}\fi

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@}\advance{\@tempdima}{\@tempdimb}\
\addtolength{\@tempdimb}{\baselineskip}\
\addtolength{\@tempdimb}{\topskip}\
\@tempdima{\@tempdimb}\
\textheight{\@tempdima}\fi

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

\advance{oddsidemargin}{\Gm@layouthoffset}\
\advance{evensidemargin}{\Gm@layouthoffset}
\Gm@detectdriver  The macro checks the typeset environment and changes the driver option if necessary. To make the engine detection more robust, the macro is rewritten with packages \texttt{ifpdf}, \texttt{ifvtex} and \texttt{ifxetex}.

\Gm@detectdriver() If the driver option is not specified explicitly, then driver auto-detection works.

\if\Gm@driver\@empty
  \typeout{*geometry* driver: auto-detecting}%
  \ifpdf
    \ifpdfextension\@undefined
      \Gm@setdriver{pdftex}%
    \else
      \Gm@setdriver{luatex}%
    \fi
  \else
    \Gm@setdriver{dvips}%
  \fi
\ifvtex
  \Gm@setdriver{vtex}%
\fi
\ifxetex
  \Gm@setdriver{xetex}%%
\fi
When the driver option is set by the user, check if it is valid or not.
\else
  \if\Gm@driver\Gm@xetex\%
    \ifxetex\else
      \Gm@warning{Wrong driver setting: 'xetex'; trying 'pdftex' driver}%
      \Gm@setdriver{pdftex}
    \fi
  \fi
  \if\Gm@driver\Gm@vtex
    \ifvtex\else
      \Gm@warning{Wrong driver setting: 'vtex'; trying 'dvips' driver}%
      \Gm@setdriver{dvips}%
    \fi
  \fi
  \if\Gm@driver\relax
    \typeout{*geometry* detected driver: <none>}%
  \else
    \typeout{*geometry* detected driver: \Gm@driver}%
  \fi
\fi
\Gm@showparams  Prints the resulted parameters and dimensions to STDOUT if \texttt{verbose} is true. \Gm@width and \Gm@height are expanded to get the real size.
\Gm@showparams#1{\ifGm@verbose\expandafter\typeout\else\expandafter\wlog\fi{\Gm@logcontent{#1}}}%
\Gm@showdim#1{* \string#1 = \the#1^^J}
\Gm@showbool#1{\@nameuse{ifGm@#1}#1\space\fi}
\Gm@logcontent  The content of geometry parameters and native dimensions for the page layout.
\Gm@logcontent#1{*geometry* verbose mode - [ #1 ] result:^^J%\ifGm@pass * pass: disregarded the geometry package!^^J%\else
* driver: \if\Gm@driver<none>\else\Gm@driver\fi\^^J%
* paper: \if\Gm@paper\undefined\default\else\Gm@paper\fi\^^J%
* layout: \if\Gm@layout\custom\else\same\size\as\paper\fi\^^J%
* \if\Gm@layout
* layout(width,height): (\the\Gm@layoutwidth,\the\Gm@layoutheight)\^^J%
* layoutoffset:(h,v)=(\the\Gm@layouthoffset,\the\Gm@layoutvoffset)\^^J%
* \if\Gm@layout
* bindingoffset: \the\Gm@bindingoffset\^^J
* modes: %
\Gm@showbool{landscape}%
\Gm@showbool{includehead}%
\Gm@showbool{includefoot}%
\Gm@showbool{includemp}%
\if@twoside twoside\space\fi%
\if@mparswitch\else\if@twoside asymmetric\space\fi\fi%
\Gm@showbool{heightrounded}%
\ifx\Gm@truedimen\@empty\else truedimen\space\fi%
\Gm@showbool{showframe}%
\Gm@showbool{showcrop}%
* h-part:(L,W,R)=(\Gm@lmargin, \Gm@width, \Gm@rmargin)^^J%
* v-part:(T,H,B)=(\Gm@tmargin, \Gm@height, \Gm@bmargin)^^J%
\fi
* \Gm@showdim{\paperwidth}%
* \Gm@showdim{\paperheight}%
* \Gm@showdim{\textwidth}%
* \Gm@showdim{\textheight}%
* \Gm@showdim{\oddsidemargin}%
* \Gm@showdim{\evensidemargin}%
* \Gm@showdim{\topmargin}%
* \Gm@showdim{\headheight}%
* \Gm@showdim{\headsep}%
* \Gm@showdim{\topskip}%
* \Gm@showdim{\footskip}%
* \Gm@showdim{\marginparsep}%
* \Gm@showdim{\columnsep}%
* \string\skip\string\footins=\the\skip\footins^^J%
* \Gm@showdim{\hoffset}%
* \Gm@showdim{\voffset}%
* \Gm@showdim{\mag}%
* \string\@twocolumn%\if\twocolumn\true\else\false\fi^^J%
* \string\@twoside%\if\twoside\true\else\false\fi^^J%
* \string\@mparswitch%\if\mparswitch\true\else\false\fi^^J%
* \string\@reversemargin%\if\reversemargin\true\else\false\fi^^J%
* (1in=72.27pt=25.4mm, 1cm=28.453pt)^^J}

Macros for the page frames and cropmarks.

\def\Gm@cropmark(#1,#2,#3,#4){%
\begin{picture}(0,0)
\setlength{\unitlength}{1trueem}%
\linethickness{0.25pt}%
\put(#3,0){\line(#1,0){17}}%
\put(0,#4){\line(0,#2){17}}%
\end{picture}}%
\ PROVIDECOMMAND*\vb@xt@{\vbox to}{}
\def\Gm@vrule{\hrule\width 0.2pt\height \textheight\depth \z@}%
\def\Gm@hrule{\hrule\height 0.2pt\depth \z@\width \textwidth}%
\def\Gm@hruled{\hrule\height \z@\depth 0.2pt\width \textwidth}%
\newcommand{\Gm@vrules@mpi}{%
\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.

```latex
\def\ProcessOptionsKV{\@ifnextchar[\@ProcessOptionsKV{\@ProcessOptionsKV[]}}
\def\@ProcessOptionsKV[#1]#2{\@if\unknownkey\@empty{\@optionlist{\@classname.@\@currext}}\@for\CurrentOption:=\@classoptionslist\do{\@if\unknownkey\CurrentOption\@tempa{\CurrentOption,}}}}
```

\ProcessOptionsKV
\Gm@processconf \ExecuteOptions is replaced with \Gm@setkey to make it possible to deal with '(key)=(value)' as its argument.

\AtBeginDocument \Gm@s wavelength{paperwidth}
\Gm@s wavelength{paperheight}
\edef\Gm@restore@org{\Gm@restore}
\ifGm@resetpaper
\edef\Gm@pw{\Gm@orgpw}
\edef\Gm@ph{\Gm@orgph}
\else
\edef\Gm@pw{\the\paperwidth}
\edef\Gm@ph{\the\paperheight}
\fi
\ifGm@pass\else
\ifnum\mag=\@m
\Gm@magtooffset
\divide\paperwidth\@m
\multiply\paperwidth\the\mag
\divide\paperheight\@m
\multiply\paperheight\the\mag
\fi
\fi
\AtBeginDocument
\Gm@save
\edef\Gm@restore@org{\Gm@restore}
\Gm@initall
\Gm@processconf
\ProcessOptionsKV[c]{Gm}
\Gm@setdefaultpaper
\ProcessOptionsKV[p]{Gm}
\Gm@process
\Gm@s wavelength{paperwidth}
\Gm@s wavelength{paperheight}
\edef\Gm@restore@org{\Gm@restore}
\ifGm@resetpaper
\edef\Gm@pw{\Gm@orgpw}
\edef\Gm@ph{\Gm@orgph}
\else
\edef\Gm@pw{\the\paperwidth}
\edef\Gm@ph{\the\paperheight}
\fi
\AtBeginDocument
The original page layout before loading geometry is saved here. \Gm@restore@org is defined here for reset option.

Processing config file.

The optional arguments to \documentclass are processed here.

Paper dimensions given by class default are stored.

The optional arguments to \usepackage are processed here.

Actual settings and calculation for layout dimensions are processed.

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

The original paper size is used if resetpaper.

If pass is not set, the paper size is multiplied according to the specified mag.
Checking the driver options.

If \texttt{xetex} and \texttt{pdfpagewidth} is defined, \texttt{pdfpagewidth} and \texttt{pdfpageheight} would be set.

1000  \Gm@detectdriver

1001  \ifx\Gm@driver\Gm@xetex
1002       \@ifundefined{pdfpagewidth}{}{%
1003            \setlength\pdfpagewidth{\Gm@pw}%
1004            \setlength\pdfpageheight{\Gm@ph}%
1005       \ifnum\mag\@m
1006            \ifx\Gm@truedimen\Gm@true
1007                \setlength\paperwidth{\Gm@pw}%
1008                \setlength\paperheight{\Gm@ph}%
1009            \fi
1010        \fi
1011       \fi
1012  \fi

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

1013  \ifx\Gm@driver\Gm@pdftex
1014       \@ifundefined{pdfpagewidth}{}{%
1015            \setlength\pdfpagewidth{\Gm@pw}%
1016            \setlength\pdfpageheight{\Gm@ph}%
1017            \@tempdima=\mag sp
1018            \@ifundefined{pdfhorigin}{}{%
1019                \divide\pdfhorigin@tempdima
1020                \multiply\pdfhorigin\@m
1021                \divide\pdfvorigin@tempdima
1022                \multiply\pdfvorigin\@m}%
1023            \ifx\Gm@truedimen\Gm@true
1024                \setlength\paperwidth{\Gm@pw}%
1025                \setlength\paperheight{\Gm@ph}%
1026            \fi
1027       \fi
1028  \fi

If \texttt{luatex} is set to \texttt{true}, pdf-commands are set properly. To avoid \texttt{luatex} magnification problem, \texttt{horigin} and \texttt{vorigin} are adjusted for \texttt{mag}.

1029  \ifx\Gm@driver\Gm@luatex
1030       \setlength\pagewidth{\Gm@pw}%
1031       \setlength\pageheight{\Gm@ph}%
1032       \ifnum\mag\@m
1033           \@tempdima=\mag sp
1034           \edef\Gm@horigin{\pdfvariable horigin}%
1035           \edef\Gm@vorigin{\pdfvariable vorigin}%
1036           \divide\Gm@horigin@tempdima
1037           \multiply\Gm@horigin\@m
1038           \divide\Gm@vorigin@tempdima
1039           \multiply\Gm@vorigin\@m}%
1040       \ifx\Gm@truedimen\Gm@true
1041           \setlength\paperwidth{\Gm@pw}%
1042           \setlength\paperheight{\Gm@ph}%
1043       \fi
1044  \fi
1045  \fi

With \texttt{VTeX} environment, \texttt{VTeX} variables are set here.

1046  \ifx\Gm@driver\Gm@vtex
1047       \@ifundefined{mediawidth}{}{%
1048           \mediawidth=\paperwidth
1049           \mediaheight=\paperheight}%
1050  \ifvtexdv
1051       \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1052  \fi
1053  \fi

39
If `dvips` or `dvipdfm` is specified, paper size is embedded in dvi file with `\special`. For `dvips`, a landscape correction is added because a landscape document converted by `dvips` is upside-down in PostScript viewers.

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

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

Page frames are shipped out when `showframe=true`, cropmarks for `showcrop=true` on each page. The `atbegshi` package is used for overloading `\shipout`.

```latex
\if\Gm@showframe
\@tempswatrue
\else\if\Gm@showcrop
\@tempswatrue
\fi\fi
\if@tempswa
\RequirePackage{atbegshi}%
\AtBeginShipout{\setbox\AtBeginShipoutBox=\vbox{%
\baselineskip\z@skip\lineskip\z@skip\lineskiplimit\z@\Gm@pageframes\box\AtBeginShipoutBox}}%
\fi
```

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

```latex
\Gm@save
\edef\Gm@restore@pkg{\Gm@restore}%
```

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

```latex
\if\Gm@verbose\if\Gm@pass\else\Gm@checkmp\fi\fi
\Gm@showparams{preamble}%
```

The following lines free the memories no longer needed.

```latex
\let\Gm@pw\relax
\let\Gm@ph\relax
```

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

```latex
\newcommand{\geometry}[1]{{%
\Gm@clean
\setkeys{Gm}{#1}%
\Gm@process}%
\Gm@onlypreamble\geometry
```

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

```latex
\DeclareRobustCommand{\Gm@changelayout}{% 
\setlength{\@colht}{\textheight}
\setlength{\@colroom}{\textheight}%
\setlength{\vsize}{\textheight}%
\setlength{\columnwidth}{\textwidth}%
\if@twocolumn%
\advance\columnwidth-\columnsep
40
```
The macro \newgeometry, which changes the layout, can be used only in the document. It would reset the options specified in the preamble except for paper size options and \mag.

\newcommand{\newgeometry}{[1]{%\clearpage \Gm@restore@org \Gm@initnewgm \Gm@newgmtrue \setkeys{Gm}{#1} \Gm@newgmfalse \Gm@process \ifnum\mag=\@m\else\Gm@magtooffset\fi \Gm@changelayout \Gm@showparams{newgeometry}}}%

\restoregeometry The macro restores the resulting layout specified in the preamble, namely the first-page layout right after \begin{document}.
\newcommand{\restoregeometry}{%\clearpage \Gm@restore@pkg \Gm@changelayout}

\savegeometry The macro saves the layout with the name specified with the argument. The saved layout can be loaded with \loadgeometry{⟨name⟩}.
\newcommand*{\savegeometry}{[1]{%\Gm@save \expandafter\edef\csname Gm@restore@@#1\endcsname{\Gm@restore}}%

\loadgeometry The macro loads the layout saved with \savegeometry{⟨name⟩}. If the name is not found, the macro would warn it and do nothing for the layout.
\newcommand*{\loadgeometry}{[1]{%\clearpage \@ifundefined{Gm@restore@@#1}{%\PackageError{geometry}{%\string\loadgeometry : name '#1' undefined}{%The name '⟨name⟩' should be predefined with \string\savegeometry} \{%\@nameuse{Gm@restore@@#1} \Gm@changelayout\}}%}

12 Config file
In the configuration file geometry.cfg, one can use \ExecuteOptions to set the site or user default settings.
\newcommand*{\ExecuteOptions}{%\clearpage \@ifundefined{Gm@restore@@#1}{%\PackageError{geometry}{%\string\loadgeometry : name '#1' undefined}{%The name '⟨name⟩' should be predefined with \string\savegeometry} \{%\@nameuse{Gm@restore@@#1} \Gm@changelayout\}}%}

13 Sample file
Here is a sample document for the geometry package.