

L'extension geometry

Hideo Umeki
latexgeometry@gmail.com

2010/09/12 v5.6

Résumé

Cette extension met à disposition une interface flexible et simple pour traiter les dimensions des pages. Vous pouvez ainsi changer la mise en page avec des paramètres intuitifs. Par exemple, si vous voulez fixer une marge de 2 cm pour chaque bord de la feuille, vous pouvez juste taper `\usepackage[margin=2cm]{geometry}`. L'aspect de la page peut être changé en cours de document avec la commande `\newgeometry`.

1 Préface à la version 5

- **Changement de la mise en page en cours de document.**
Les nouvelles commandes `\newgeometry{...}` et `\restoregeometry` permettent à l'utilisateur de changer les dimensions de la page en cours de document. `\newgeometry` est assez similaire à `\geometry`, à ceci près que `\newgeometry` désactive toutes les options indiquées dans le préambule mais conserve les options liées à la taille du papier : `landscape`, `portrait` et les options de format (telles que `papersize`, `paper=a4paper` et ainsi de suite).
- **Un nouvel ensemble d'options pour spécifier la zone de mise en page.**
Les options spécifiant la zone dans laquelle sont calculées les dimensions de la page sont ajoutées : `layout`, `layoutsizex`, `layoutwidth`, `layoutheight` et ainsi de suite. Ces options aident à imprimer la mise en forme pour une autre taille de papier. Par exemple, avec `a4paper` et `layout=a5paper`, l'extension `geometry` utilise la mise en forme A5 pour calculer les marges avec un format de papier A4.
- **Une nouvelle option de pilote xetex.**
Une nouvelle option de pilote, `xetex`, est ajoutée. La routine d'auto-détection des pilotes a été revue pour éviter une erreur avec les commandes non définies. Notez que « `geometry.cfg` » (dans `TEX Live`), qui désactive la routine d'auto-détection et sélectionne `pdftex`, n'est plus nécessaire et ne crée plus de problème quand bien même il existe toujours. Sélectionner `xetex` est fortement recommandé avec `XYLATEX`.
- **Nouveaux formats de papier prédéfinis pour les formats JIS B et les formats ISO C**
Les formats de papier prédéfinis `b0j` à `b6j` pour les formats JIS (Japanese Industrial Standards) B et `c0paper` à `c6paper` pour les formats ISO C (v5.4~) sont ajoutés.
- **Changement des valeurs par défaut pour les marges sous-définies.**
Dans les versions précédentes, si une marge seulement était spécifiée, `bottom=1cm` par exemple, alors `geometry` fixait l'autre marge avec le ratio de marge (1:1 par défaut pour les dimensions verticales) et obtenait `top=1cm` dans ce cas. La version 5 fixe la taille du corps du texte avec la valeur par défaut `scale` (= 0.7) et détermine la marge non spécifiée. (Voir Section 6.5)
- **Les options `showframe` et `showcrop` fonctionnent sur chaque page.**
Avec l'option `showframe`, le cadre de la page est affiché sur chaque page. En complément, une nouvelle option `showcrop` affiche les traits de coupe à chaque coin de la zone de mise en page sur chaque page. Notez que les marques seront invisibles sans spécification d'une taille de mise en page plus petite que la taille du papier. La version 5.4 introduit un nouveau processus de débordement `\shipout` utilisant l'extension `atbegshi` impliquant le chargement de cette extension quand les options `showframe` ou `showcrop` sont sélectionnées.
- **Le chargement de `geometry.cfg` précède le traitement des options de classe.**
La version antérieure chargeait `geometry.cfg` après avoir traité les options de classe du document. Maintenant que le fichier de configuration est chargé avant, vous pouvez changer le comportement indiqué dans `geometry.cfg` en ajoutant des options dans `\documentclass` aussi bien que dans `\usepackage` et `\geometry`.
- **Options supprimées : `compat2` et `twosideshift`.** La version 5 n'est plus compatible avec ces options `compat2` et `twosideshift` pour des raisons de simplification.

2 Introduction

Fixer des dimensions pour une mise en page sous L^AT_EX n'est pas immédiat. Vous devez ajuster plusieurs dimensions inhérentes à L^AT_EX pour placer une zone de texte où vous le souhaitez. Si vous voulez centrer la zone de texte dans un papier que vous utilisez, par exemple, vous devez spécifier les dimensions fondamentales suivantes :

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

Sans l'extension `calc`, l'exemple ci-dessus demanderait des réglages plus fastidieux. L'extension `geometry` présente une façon simple de fixer des paramètres de mise en page. Dans ce cas, ce que vous avez à faire est juste

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

En plus du centrage, le réglage des marges par rapport à chaque bord de la feuille est également problématique. Mais `geometry` simplifie aussi ce point. Si vous souhaitez fixer chaque marge à 1,8 cm, vous pouvez saisir

```
\usepackage[margin=3.8cm]{geometry}
```

De fait, l'extension `geometry` dispose d'un mécanisme d'auto-complétion dans lequel les dimensions manquantes sont automatiquement déterminées. L'extension `geometry` sera également utile lorsque vous avez à traiter la mise en page en suivant des instructions strictes. Par exemple,

La largeur maximale de la zone de texte est de 16,5 cm de large par 22,25 cm de haut. La marge de tête sur chaque page doit être de 3 cm depuis le bord haut de la feuille. La marge gauche doit être de 2,3 cm depuis le bord gauche. Le pied de page avec le numéro de la page doit être en bas de la zone de texte.

Dans ce cas, avec `geometry` vous saisissez

```
\usepackage[total={16.5cm,22.25cm},
top=3cm, left=2.3cm, includefoot]{geometry}.
```

Définir une zone de texte sur papier dans un système de préparation de document présente certaines analogies avec le placement d'une fenêtre dans un système d'environnement de bureau. Le nom « `geometry` » provient de l'option `-geometry` utilisée pour spécifier la taille et la localisation d'une fenêtre dans l'environnement de bureau X Window.

3 Anatomie d'une page

La Figure 1 montre les dimensions associées à la mise en page telle que définies par l'extension `geometry`. Le gabarit contient un *corps intégral* (zone imprimable) et des *marges*. Le *corps intégral* se compose d'un *corps* (zone de texte) et d'éléments optionnels : un *en-tête*, un *pied de page* et une note marginale (`marginpar`). Il y a quatre marges : *gauche*, *droite*, *haute* et *basse*. Pour les documents en recto-verso, les marges horizontales devraient être appelées *intérieure* et *extérieure*.

<i>papier</i>	:	<i>corps intégral</i> et <i>marges</i>
<i>corps intégral</i>	:	<i>corps</i> (zone de texte) (<i>en-tête</i> , <i>pied de page</i> et <i>note marginale</i> optionnels)
<i>marges</i>	:	<i>gauche</i> (<i>intérieure</i>), <i>droite</i> (<i>extérieure</i>), <i>haute</i> et <i>basse</i>

Chaque marge est mesurée à partir du bord du papier associé. Par exemple, la marge gauche (marge intérieure) correspond à distance horizontale entre le bord gauche (intérieur) du papier et celui du corps total. C'est pourquoi les marges gauche et haute définies dans `geometry` diffèrent des dimensions usuelles `\leftmargin` et `\topmargin`. La taille du corps (zone de texte) peut être modifiée avec `\textwidth` et `\textheight`.

Les dimensions du papier, du corps intégral et des marges ont les relations suivantes.

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

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

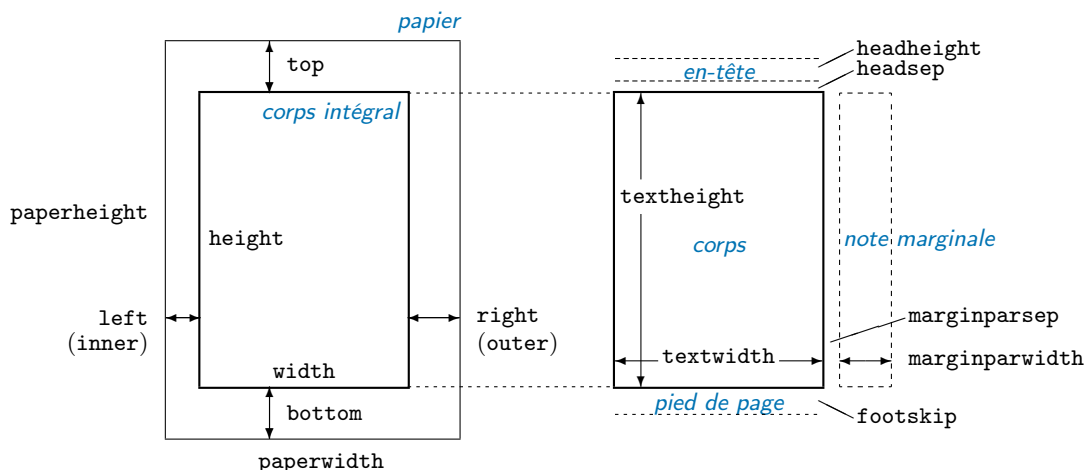


FIGURE 1 – Noms des dimensions utilisées dans l’extension `geometry`. `width` = `textwidth` et `height` = `textheight` par défaut. `left`, `right`, `top` et `bottom` sont des marges. Si les marges des pages au verso sont interverties par l’option `twoside`, les marges spécifiées par les options `left` et `right` sont utilisées pour les marges intérieures et extérieures respectivement. `inner` et `outer` sont des alias pour `left` et `right` respectivement.

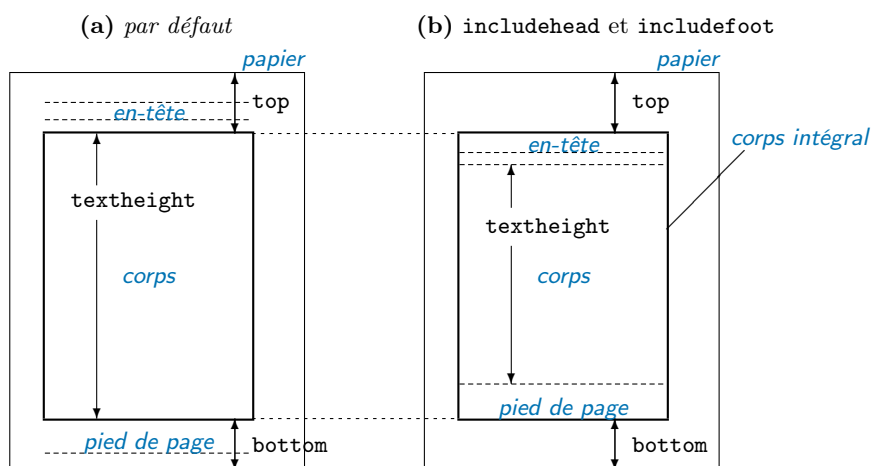


FIGURE 2 – `includehead` et `includefoot` incluent respectivement l’en-tête et le pied de page dans le `corps intégral`. (a) `height` = `textheight` (par défaut). (b) `height` = `textheight` + `headheight` + `headsep` + `footskip` si `includehead` et `includefoot` valent `true`. Si les marges haute et basse sont spécifiées, `includehead` et `includefoot` conduisent à un `textheight` plus court.

La largeur `width` et la hauteur `height` du corps intégral sont définies par :

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

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

Dans l’Équation (3), `width` égale `textwidth` par défaut, tandis que `marginparsep` et `marginparwidth` sont inclus dans `width` si l’option `includemp` est fixée à `true`. Dans l’Équation (4), `height` égale `textheight` par défaut. Si `includehead` est fixée à `true`, `headheight` et `headsep` sont considérées comme des parties de `height`. De la même manière, `includefoot` place `footskip` dans `height`. La Figure 2 montre comment ces options fonctionnent verticalement.

Ainsi, le gabarit se décompose en trois parties (longueurs) dans chaque direction : un corps et deux marges. Si deux d’entre elles sont explicitement spécifiées, l’autre longueur est alors évidente et n’a pas besoin d’être spécifiée. La Figure 3 montre un modèle simple de dimensions d’une page. Quand une longueur L est donnée et est décomposée entre le corps b , les marges a et c , il est évident que

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

Une spécification avec deux des trois longueurs (a, b et c) fixées explicitement est résolvable. Si deux longueurs ou plus sont laissées non spécifiées ou « sous-spécifiées », l’Équation (5) ne peut être résolue

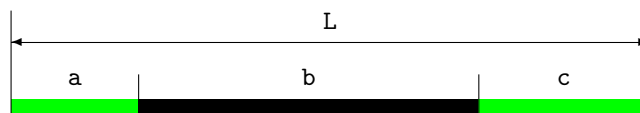


FIGURE 3 – Un modèle simple des dimensions de la page.

sans une autre relation entre elles. Si elles sont toutes spécifiées autrement dit trop spécifiées ou « sur-spécifiées », alors nous devons vérifier si elles satisfont ou pas l'Équation (5).

L'extension `geometry` dispose d'un mécanisme d'auto-complétion qui évite les problèmes liés à la spécification des dimensions de la mise en page. Par exemple, vous pouvez retenir

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

sur papier A4. Dans ce cas, nous n'avez pas à définir la marge droite. Les détails de cette auto-complétion sont décrits dans la Section 6.5.

4 Interface utilisateur

4.1 Commandes

L'extension `geometry` propose les commandes suivantes :

- `\geometry{<options>}`
- `\newgeometry{<options>}` et `\restoregeometry`
- `\savegeometry{<nom>}` et `\loadgeometry{<nom>}`

`\geometry{<options>}` change la mise en page selon les options spécifiées en argument. Cette commande, plus que toute autre, devrait être placée uniquement dans le préambule (avant `\begin{document}`).

L'extension `geometry` peut être utilisée comme une partie d'une classe ou d'une autre extension que vous utilisez dans votre document. La commande `\geometry` peut écraser certains réglages dans l'en-tête. L'utilisation de plusieurs `\geometry` est autorisée et conduit à un traitement de l'ensemble des options concaténées. Si `geometry` n'est pas encore chargé, vous pouvez seulement utiliser `\usepackage[<options>]{geometry}` au lieu de `\geometry`.

`\newgeometry{<options>}` change la mise en page en cours de document. `\newgeometry` est similaire à `\geometry` à ceci près que `\newgeometry` annule toutes les options spécifiées par `\usepackage` et `\geometry` en préambule et ne tient pas compte des options liées à la taille du papier. `\restoregeometry` restaure la mise en page spécifiée dans le préambule. Cette commande n'a pas d'argument. Voir la Section 7 pour plus d'informations.

`\savegeometry{<nom>}` sauvegarde sous le nom `<nom>` les dimensions de la page à l'endroit où la commande est placée. `\loadgeometry{<nom>}` charge les dimensions de la page sauvegardées sous le nom `<nom>`. Voir la Section 7 pour plus d'informations.

4.2 Argument optionnel

L'extension `geometry` adopte l'interface keyval « `<clé>=<valeur>` » pour l'argument optionnel de `\usepackage`, `\geometry` et `\newgeometry`.

L'argument inclut une liste d'options keyval séparées par des virgules et a les règles de base suivantes :

- des lignes multiples sont autorisées mais pas les lignes vides ;
- les espaces entre les mots sont ignorés ;
- les options sont traitées indépendamment de leur ordre de saisie. (Il y a ici quelques exceptions.

Voir la Section 6.2 pour plus d'informations.)

Par exemple,

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

est équivalent à

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

Certaines options sont autorisées à présenter une sous-liste, par exemple `{3cm,0.8in}`. Notez que l'ordre des valeurs dans la sous-liste est significatif. Le réglage ci-dessus est équivalent aux suivants :

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

ou

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

Ainsi, l'utilisation de plusieurs `\geometry` cumule juste les options.

`geometry` supporte l'extension *calc*¹. Par exemple,

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

4.3 Types d'option

Les options de `geometry` sont réparties en quatre types :

1. Type booléen

prend une valeur booléenne (vrai `true` ou faux `false`). En l'absence de valeur, `true` est sélectionné par défaut.

$\langle key \rangle = \text{true} \mid \text{false}$.

$\langle key \rangle$ sans valeur est équivalent à $\langle key \rangle = \text{true}$.

Exemples : `verbose=true`, `includehead`, `twoside=false`.

Le format (ou nom) du papier est ici l'exception. Le format du papier désiré devrait être mis dans valeur. Quelle que soit la valeur donnée, elle est ignorée. Par exemple, `a4paper=XXX` est équivalent à `a4paper`.

2. Type à valeur unique

prend une valeur obligatoire.

$\langle clé \rangle = \langle valeur \rangle$.

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

3. Type à valeur double

prend une paire de valeurs obligatoires séparées par une virgule et regroupées entre accolades. Les deux valeurs peuvent être réduites à une valeur unique si elles sont identiques.

$\langle clé \rangle = \{ \langle valeur1 \rangle, \langle valeur2 \rangle \}$.

$\langle clé \rangle = \langle valeur \rangle$ est équivalent à $\langle clé \rangle = \{ \langle valeur \rangle, \langle valeur \rangle \}$.

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

4. Type à triple valeur

prend trois valeurs obligatoires séparées par une virgule et regroupées entre accolades.

$\langle clé \rangle = \{ \langle valeur1 \rangle, \langle valeur2 \rangle, \langle valeur3 \rangle \}$

Chaque valeur doit être une dimension ou être nulle. Quand vous indiquez une valeur vide ou « * », cela correspond à un nul et laisse la valeur souhaitée aux bons soins du mécanisme d'auto-complétion. Vous devez spécifier au moins une dimension, typiquement deux dimensions. Vous pouvez indiquer des nuls pour toutes les valeurs mais cela n'a pas de sens. *Exemples :*

`hdivide={2cm,*,1cm}`, `vdivide={3cm,19cm, }`, `divide={1in,*,1in}`.

5 Détail des options

Cette section décrit toutes les options disponibles dans `geometry`. Les options avec une dague[†] ne sont pas disponibles comme argument de `\newgeometry` (voir la Section 7).

5.1 Taille du papier

Les options ci-dessous définissent la taille du papier/médium et son orientation.

[†] `paper` | `papername`

spécifie la taille du papier par son format. `paper=⟨format⟩`. Par simplification, vous pouvez spécifier cette taille sans `paper=`. Par exemple, `a4paper` est équivalent à `paper=a4paper`.

1. CTAN : `macros/latex/required/tools`

[†] <code>a0paper</code> , <code>a1paper</code> , <code>a2paper</code> , <code>a3paper</code> , <code>a4paper</code> , <code>a5paper</code> , <code>a6paper</code> , <code>b0paper</code> , <code>b1paper</code> , <code>b2paper</code> , <code>b3paper</code> , <code>b4paper</code> , <code>b5paper</code> , <code>b6paper</code> , <code>c0paper</code> , <code>c1paper</code> , <code>c2paper</code> , <code>c3paper</code> , <code>c4paper</code> , <code>c5paper</code> , <code>c6paper</code> , <code>b0j</code> , <code>b1j</code> , <code>b2j</code> , <code>b3j</code> , <code>b4j</code> , <code>b5j</code> , <code>b6j</code> , <code>ansipaper</code> , <code>ansibpaper</code> , <code>ansicpaper</code> , <code>ansidpaper</code> , <code>ansiepaper</code> , <code>letterpaper</code> , <code>executivepaper</code> , <code>legalpaper</code>	spécifie le format du papier. La valeur assignée à cette option est toujours ignorée. Par exemple, les cas suivants ont le même effet : <code>a5paper</code> , <code>a5paper=true</code> , <code>a5paper=false</code> et ainsi de suite. <code>a[0-6]paper</code> , <code>b[0-6]paper</code> et <code>c[0-6]paper</code> sont les formats ISO A, B et C respectivement. Les formats JIS (Japanese Industrial Standards) A sont identiques aux formats ISO A mais les formats JIS B sont différents des formats ISO B. <code>b[0-6]j</code> devrait être utilisé pour les formats JIS B.
[†] <code>screen</code>	un format spécial avec (L,H) = (225mm,180mm). Pour les présentations sur ordinateur ou pour projecteur, « <code>screen,centering</code> » avec la classe de document « <code>slide</code> » devrait être utile.
[†] <code>paperwidth</code>	largeur du papier. <code>paperwidth=<longueur></code> .
[†] <code>paperheight</code>	hauteur du papier. <code>paperheight=<longueur></code> .
[†] <code>papersize</code>	largeur et hauteur du papier. <code>papersize={<largeur>,<hauteur>}</code> ou <code>papersize=<longueur></code> .
[†] <code>landscape</code>	bascule l'orientation du papier en mode paysage.
[†] <code>portrait</code>	bascule l'orientation du papier en mode portrait. Ceci est équivalent à <code>landscape=false</code> .

Les options de format de papier (telle `a4paper`) et d'orientation (`portrait` et `landscape`) peuvent être déclarée dans les options de la classe du document. Par exemple, si vous utilisez

```
\documentclass[a4paper,landscape]{article}
```

alors `a4paper` et `landscape` seront traités dans `geometry` également. C'est également le cas pour `twoside` et `twocolumn` (voir également la Section 5.5).

5.2 Taille du gabarit

Vous pouvez spécifier le gabarit (ou zone de mise en page) avec les options décrites dans cette section indépendamment de la taille du papier. Ces options aident à imprimer un gabarit spécifique sur un papier de taille différente. Par exemple, avec `a4paper` et `layout=a5paper`, l'extension utilise le gabarit « A5 » pour calculer les marges sur un papier « A4 ». La taille du gabarit est par défaut celle du papier. Les options pour la taille du gabarit sont disponibles dans `\newgeometry`, de telle manière à ce que vous puissiez changer la taille du gabarit en cours de document. La taille du papier ne peut cependant être modifiée. La Figure 4 montre la différence entre `layout` et `paper`.

<code>layout</code>	spécifie la taille du gabarit par nom de format. <code>layout=<format></code> . Tous les formats de papier définis dans <code>geometry</code> sont disponibles. Voir la Section 5.1 pour plus d'informations.
<code>layoutwidth</code>	largeur du gabarit. <code>layoutwidth=<longueur></code> .
<code>layoutheight</code>	hauteur du gabarit. <code>layoutheight=<longueur></code> .
<code>layoutsizes</code>	largeur et hauteur du gabarit. <code>layoutsizes={<largeur>,<hauteur>}</code> ou <code>layoutsizes=<longueur></code> .
<code>layouthoffset</code>	spécifie le décalage horizontal par rapport au bord gauche du papier. <code>layouthoffset=<longueur></code> .
<code>layoutvoffset</code>	spécifie le décalage vertical par rapport au bord haut du papier. <code>layoutvoffset=<longueur></code> .
<code>layoutoffset</code>	spécifie le décalage horizontal et le décalage vertical. <code>layoutoffset={<hoffset>,<voffset>}</code> ou <code>layoutsizes=<longueur></code> .

5.3 Taille du corps

Les options spécifiant la taille du *corps intégral* sont décrites dans cette section.

<code>hscale</code>	rapport de largeur entre le <i>corps intégral</i> et <code>\paperwidth</code> noté <code>hscale=<rapport-h></code> . Par exemple, <code>hscale=0.8</code> est équivalent à <code>width=0.8\paperwidth</code> . (0.7 par défaut)
<code>vscale</code>	rapport de hauteur entre le <i>corps intégral</i> et <code>\paperheight</code> noté <code>vscale=<rapport-v></code> . Par exemple, <code>vscale=0.9</code> est équivalent à <code>height=0.9\paperheight</code> . (0.7 par défaut)

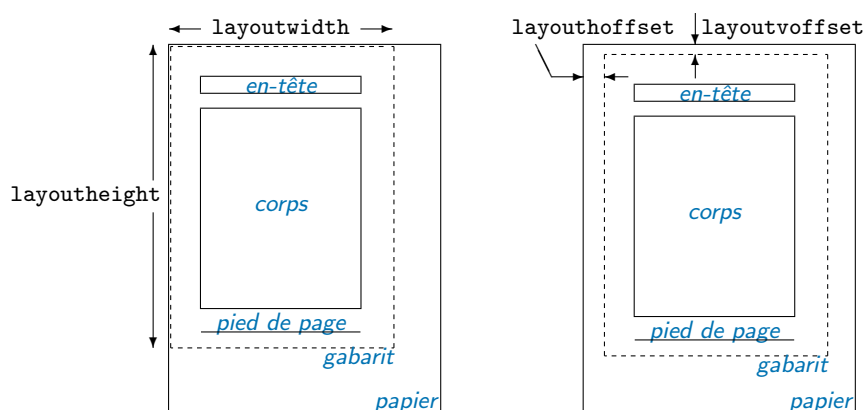


FIGURE 4 – Les dimensions liées à la taille du gabarit. Notez que la taille du gabarit est par défaut celle du papier, ce qui fait que vous n’avez pas à spécifier explicitement les options liées au gabarit la plupart du temps.

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

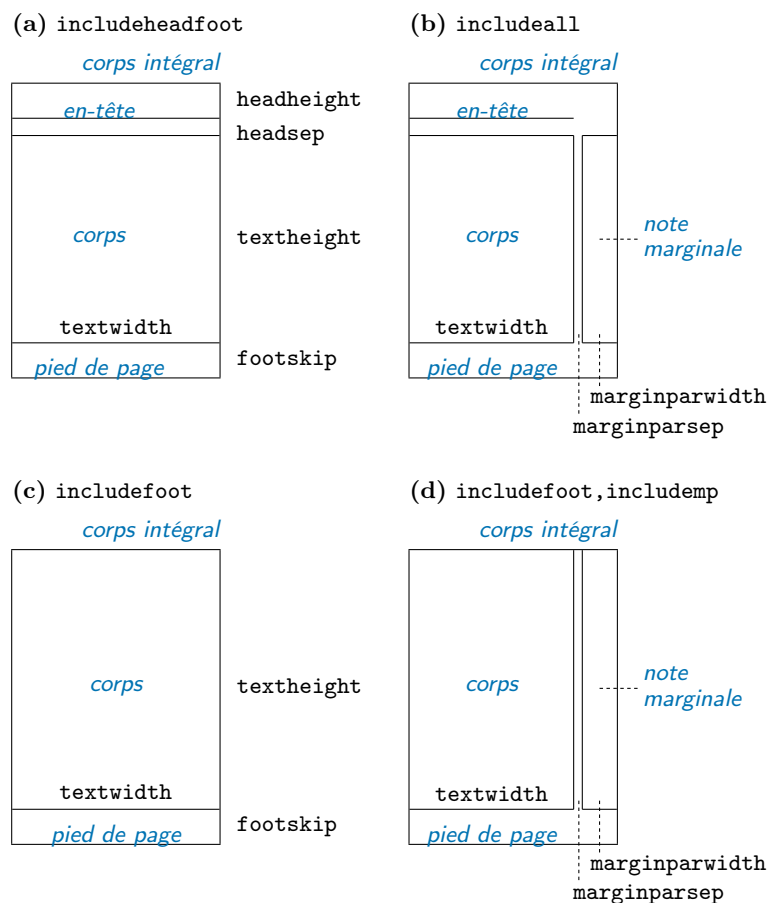


FIGURE 5 – Sample layouts for *corps intégral* with different switches. (a) `includeheadfoot`, (b) `includeall`, (c) `includefoot` and (d) `includefoot,includemp`. If `reversemp` is set to `true`, the location of the marginal notes are swapped on every page. Option `twoside` swaps both margins and marginal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

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

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

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

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

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

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

5.4 Margin size

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

<code>left</code> <code>lmargin</code> <code>inner</code>	left margin (for oneside) or inner margin (for twoside) of <i>corps intégral</i> . In other words, the distance between the left (inner) edge of the paper and that of <i>corps intégral</i> . <code>left=⟨length⟩</code> . <code>inner</code> has no special meaning, just an alias of <code>left</code> and <code>lmargin</code> .
<code>right</code> <code>rmargin</code> <code>outer</code>	right or outer margin of <i>corps intégral</i> . <code>right=⟨length⟩</code> .
<code>top</code> <code>tmargin</code>	top margin of the page. <code>top=⟨length⟩</code> . Note this option has nothing to do with the native dimension <code>\topmargin</code> .
<code>bottom</code> <code>bmargin</code>	bottom margin of the page. <code>bottom=⟨length⟩</code> .
<code>hmargin</code>	left and right margin. <code>hmargin={⟨left margin⟩,⟨right margin⟩}</code> or <code>hmargin=⟨length⟩</code> .
<code>vmargin</code>	top and bottom margin. <code>vmargin={⟨top margin⟩,⟨bottom margin⟩}</code> or <code>vmargin=⟨length⟩</code> .
<code>margin</code>	<code>margin={A,B}</code> is equivalent to <code>hmargin={A,B}</code> and <code>vmargin={A,B}</code> . <code>margin=A</code> is automatically expanded to <code>hmargin=A</code> and <code>vmargin=A</code> .
<code>hmarginratio</code>	horizontal margin ratio of <code>left</code> (inner) to <code>right</code> (outer). The value of <code>⟨ratio⟩</code> should be specified with colon-separated two values. Each value should be a positive integer less than 100 to prevent arithmetic overflow, e.g., 2:3 instead of 1:1.5. The default ratio is 1:1 for oneside, 2:3 for twoside.
<code>vmarginratio</code>	vertical margin ratio of <code>top</code> to <code>bottom</code> . The default ratio is 2:3.
<code>marginratio</code> <code>ratio</code>	horizontal and vertical margin ratios. <code>marginratio={⟨horizontal ratio⟩,⟨vertical ratio⟩}</code> or <code>marginratio=⟨ratio⟩</code> .
<code>hcentering</code>	sets auto-centering horizontally and is equivalent to <code>hmarginratio=1:1</code> . It is set to <code>true</code> by default for oneside. See also <code>hmarginratio</code> .
<code>vcentering</code>	sets auto-centering vertically and is equivalent to <code>vmarginratio=1:1</code> . The default is <code>false</code> . See also <code>vmarginratio</code> .
<code>centering</code>	sets auto-centering and is equivalent to <code>marginratio=1:1</code> . See also <code>marginratio</code> . The default is <code>false</code> . See also <code>marginratio</code> .
<code>twoside</code>	switches on twoside mode with left and right margins swapped on verso pages. The option sets <code>\@twoside</code> and <code>\@mparswitch</code> switches. See also <code>asymmetric</code> .
<code>asymmetric</code>	implements a twosided layout in which margins are not swapped on alternate pages (by setting <code>\oddsidemargin</code> to <code>\evensidemargin + bindingoffset</code>) and in which the marginal notes stay always on the same side. This option can be used as an alternative to the <code>twoside</code> option. See also <code>twoside</code> .
<code>bindingoffset</code>	removes a specified space from the lefthand-side of the page for oneside or the inner-side for twoside. <code>bindingoffset=⟨length⟩</code> . This is useful if pages are bound by a press binding (glued, stitched, stapled ...). See Figure 6.
<code>hdivide</code>	See description in Section 5.3.
<code>vdivide</code>	See description in Section 5.3.
<code>divide</code>	See description in Section 5.3.

5.5 Native dimensions

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

<code>headheight</code> <code>head</code>	modifies <code>\headheight</code> , height of header. <code>headheight=⟨length⟩</code> or <code>head=⟨length⟩</code> .
<code>headsep</code>	modifies <code>\headsep</code> , separation between header and text (body). <code>headsep=⟨length⟩</code> .
<code>footskip</code> <code>foot</code>	modifies <code>\footskip</code> , distance separation between baseline of last line of text and baseline of footer. <code>footskip=⟨length⟩</code> or <code>foot=⟨length⟩</code> .
<code>nohead</code>	eliminates spaces for the head of the page, which is equivalent to both <code>\headheight=0pt</code> and <code>\headsep=0pt</code> .
<code>nofoot</code>	eliminates spaces for the foot of the page, which is equivalent to <code>\footskip=0pt</code> .
<code>noheadfoot</code>	equivalent to <code>nohead</code> and <code>nofoot</code> , which means that <code>\headheight</code> , <code>\headsep</code> and <code>\footskip</code> are all set to <code>0pt</code> .

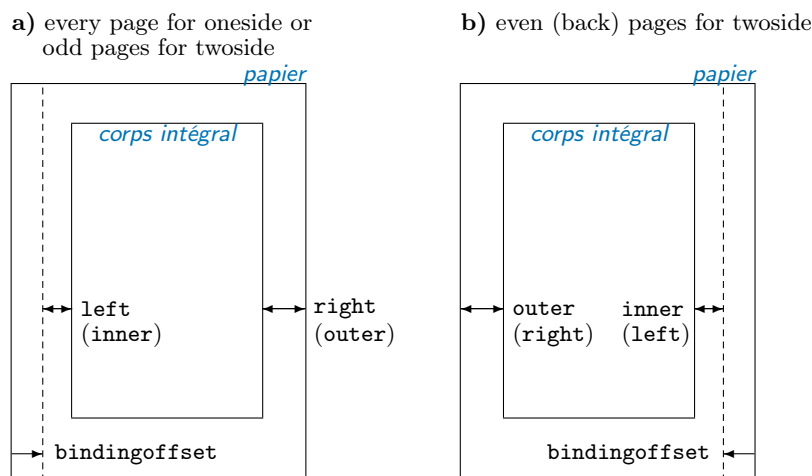


FIGURE 6 – The option `bindingoffset` adds the specified length to the inner margin. Note that `twoside` option swaps the horizontal margins and the marginal notes together with `bindingoffset` on even pages (see **b**)), but `asymmetric` option suppresses the swap of the margins and marginal notes (but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

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

5.6 Drivers

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

[†] <code>driver</code>	specifies the driver with <code>driver=<driver name></code> . <code>dvips</code> , <code>dvipdfm</code> , <code>pdftex</code> , <code>vtex</code> , <code>xetex</code> , <code>auto</code> and <code>none</code> are available as a driver name. The names except for <code>auto</code> and <code>none</code> can be specified directly with the name without <code>driver=</code> . <code>driver=auto</code> makes the auto-detection work whatever the previous setting is. <code>driver=none</code> disables the auto-detection and sets no driver, which may be useful when you want to let other package work out the driver setting. For example, if you want to use <code>crop</code> package with
----------------------------------	---

	geometry, you should call <code>\usepackage[driver=none]{geometry}</code> before the <code>crop</code> package.
[†] <code>dvips</code>	writes the paper size in dvi output with the <code>\special</code> macro. If you use <i>dvips</i> as a DVI-to-PS driver, for example, to print a document with <code>\geometry{a3paper,landscape}</code> on A3 paper in landscape orientation, you don't need options <code>"-t a3 -t landscape"</code> to <i>dvips</i> .
[†] <code>dvipdfm</code>	works like <code>dvips</code> except for landscape correction. You can set this option when using <code>dvipdfmx</code> and <code>xdvipdfmx</code> to process the dvi output.
[†] <code>pdftex</code>	sets <code>\pdfpagewidth</code> and <code>\pdfpageheight</code> internally.
[†] <code>xetex</code>	is the same as <code>pdftex</code> except for ignoring <code>\pdf{h,v}origin</code> undefined in X _Ǝ LaTeX. This option is introduced in the version 5. Note that 'geometry.cfg' in T _E X Live, which disables the auto-detection routine and sets <code>pdftex</code> , is no longer necessary, but has no problem even though it's left undeleted. Instead of <code>xetex</code> , you can specify <code>dvipdfm</code> with X _Ǝ LaTeX if you want to use specials of <code>dvipdfm</code> X _Ǝ LaTeX supports.
[†] <code>vtex</code>	sets dimensions <code>\mediawidth</code> and <code>\mediaheight</code> for V _T E _X . When this driver is selected (explicitly or automatically), <code>geometry</code> will auto-detect which output mode (DVI, PDF or PS) is selected in V _T E _X , and do proper settings for it.

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

5.7 Other options

The other useful options are described here.

[†] <code>verbose</code>	displays the parameter results on the terminal. <code>verbose=false</code> (default) still puts them into the log file.
[†] <code>reset</code>	sets back the layout dimensions and switches to the settings before <code>geometry</code> is loaded. Options given in <code>geometry.cfg</code> are also cleared. Note that this cannot reset <code>pass</code> and <code>mag</code> with <code>truedimen</code> . <code>reset=false</code> has no effect and cannot cancel the previous <code>reset(=true)</code> if any. For example, when you go <pre> \documentclass[landscape]{article} \usepackage[twoside,reset,left=2cm]{geometry} </pre> with <code>\ExecuteOptions{scale=0.9}</code> in <code>geometry.cfg</code> , then as a result, <code>landscape</code> and <code>left=2cm</code> remain effective, and <code>scale=0.9</code> and <code>twoside</code> are ineffective.
[†] <code>mag</code>	sets magnification value (<code>\mag</code>) and automatically modifies <code>\hoffset</code> and <code>\voffset</code> according to the magnification. <code>mag=<value></code> . Note that <code><value></code> should be an integer value with 1000 as a normal size. For example, <code>mag=1414</code> with <code>a4paper</code> provides an enlarged print fitting in <code>a3paper</code> , which is 1.414 ($=\sqrt{2}$) times larger than <code>a4paper</code> . Font enlargement needs extra disk space. Note that setting <code>mag</code> should precede any other settings with 'true' dimensions, such as <code>1.5truein</code>, <code>2truecm</code> and so on. See also <code>truedimen</code> option.
[†] <code>truedimen</code>	changes all internal explicit dimension values into <i>true</i> dimensions, e.g., <code>1in</code> is changed to <code>1truein</code> . Typically this option will be used together with <code>mag</code> option. Note that this is ineffective against externally specified dimensions. For example, when you set <code>"mag=1440, margin=10pt, truedimen"</code> , margins are not 'true' but magnified. If you want to set exact margins, you should set like <code>"mag=1440, margin=10truept, truedimen"</code> instead.
[†] <code>pass</code>	disables all of the <code>geometry</code> options and calculations except <code>verbose</code> and <code>showframe</code> . It is order-independent and can be used for checking out the page layout of the documentclass, other packages and manual settings without <code>geometry</code> .
[†] <code>showframe</code>	shows visible frames for the text area and page, and the lines for the head and foot on the first page.
[†] <code>showcrop</code>	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 T_EX can find it, `geometry` loads it first. For example, in `geometry.cfg` you may write `\ExecuteOptions{a4paper}`, which specifies A4 paper as the default paper.

Basically you can use all the options defined in `geometry` with `\ExecuteOptions{}`.

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

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

6.2 Order of options

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

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

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

`[verbose=true, verbose=false]` results in `verbose=false`.

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

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

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

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

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

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

6.3 Priority

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

priority : low \longrightarrow high

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

For example,

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

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

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

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

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,

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

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

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

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

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

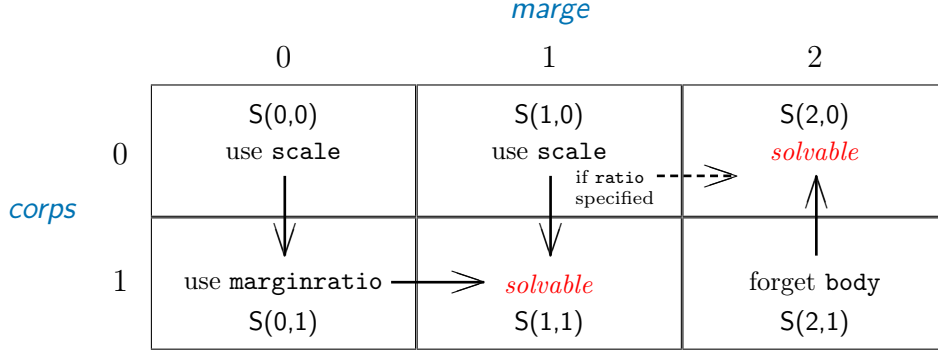


FIGURE 7 – Specifications $S(0,0)$ to $S(2,1)$ and the completion rules (arrows). Column and row numbers denote the number of explicitly specified lengths for margin and body respectively. $S(m,b)$ denote a specification with a set of the numbers $(\textit{marge}, \textit{corps}) = (m, b)$.

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

For *twosided* document with *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 $(\textit{marge}, \textit{corps}) = (m, b)$.

For example, the specification *width*=14cm, *left*=3cm is categorized into $S(1,1)$, which is an adequate specification. If you add *right*=4cm, it would be in $S(2,1)$ and overspecified. If only *width*=14cm 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 *corps* with the default *scale* (= 0.7). For example, *width* is set to be $0.7 \times \textit{layoutwidth}$. Note that by default *layoutwidth* and *layoutheight* will be equal to *\paperwidth* and *\paperheight* respectively. Thus $S(0,0)$ goes to $S(0,1)$. See $S(0,1)$.

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

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

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

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

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

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

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

- S(1,0)** Only one margin is specified, such as `bottom=2cm`, `left=1in`, `top=3cm`, and so forth.
- **If the margin ratio is *not* specified**, geometry sets *corps* with the default *scale* ($= 0.7$). For example, if `top=2.4cm` is specified, geometry sets $\text{height} = 0.7 \times \text{layoutheight}$ ($= 0.7 \backslash \text{paperheight}$ by default), then `S(1,0)` goes to `S(1,1)`, in which `bottom` is calculated with $\text{layoutheight} - (\text{height} + \text{top})$ and results in 6.51cm on A4 paper if the layout size is equal to the paper size.
 - **If the margin ratio is specified**, such as `hmarginratio={1:2}`, `vratio={3:4}` and so forth, geometry sets the other margin with the specified margin ratio. For example, if a set of options “`top=2.4cm, vratio={3:4}`” is specified, geometry sets `bottom` to be 3.2cm calculating
$$\text{bottom} = \text{top} / 3 \times 4 = 3.2\text{cm}$$
 Thus `S(1,0)` goes to `S(2,0)`.

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 `top=2.4cm` is specified, you got `bottom=2.4cm` in the version 4 or earlier, but you will get `bottom=6.51cm` in the version 5.

- S(2,1)** The *corps* and two *marges* 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 *marges* if dimensions are overspecified, geometry forgets and resets *corps*. For example, if you specify

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

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

7 Changing layout mid-document

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

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

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

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

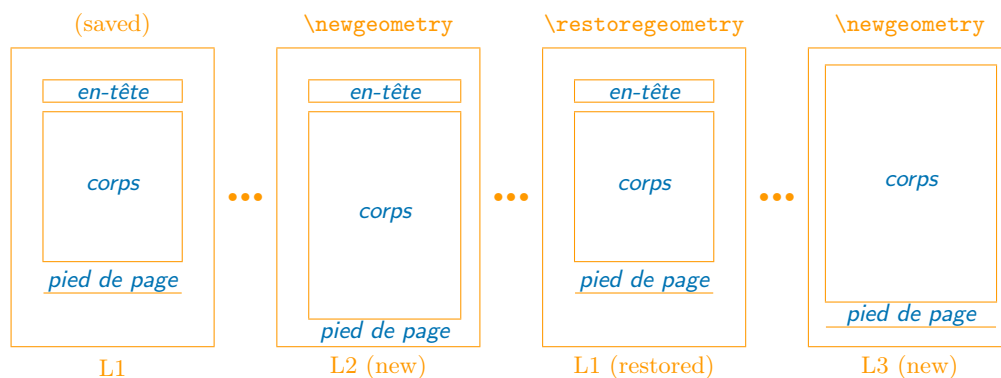
```
\usepackage[hmargin=3cm]{geometry}
\begin{document}
  Layout L1

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

  \restoregeometry
  Layout L1 (restored)

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

\end{document}
```



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

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

8 Examples

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 *corps* is set with `scale=0.7`, which means `width = 0.7 \times layoutwidth` ($= 0.7 \backslash \text{paperwidth}$ 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 *corps intégral*. 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 *corps intégral* 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 *en-tête* and *pied de page* are excluded from `height` of *corps intégral*. 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`
- 5. A layout with `textwidth` and `textheight` 90% of the paper and with *corps* 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={*,0.9\paperwidth,*}, vdivide={*,0.9\paperheight,*}` (as for onesided documents)
 - `margin={0.05\paperwidth,0.05\paperheight}`
 You can add `heightrounded` to avoid an “underfull vbox warning” like


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

 See Section 5.3 for the detailed description about `heightrounded`.
- 6. A layout with the width of marginal notes set to 3cm and included in the width of *corps intégral*. The following examples are the same.
 - `marginparwidth=3cm, includemp`
 - `marginpar=3cm, ignoremp=false`
- 7. A layout where *corps* 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.
Page 1
\restoregeometry % restores the original 'book' layout.
Page 2 and more
\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

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

11 Implementation

```

1 (*package)

This package requires the following packages : keyval, ifpdf, ifvtex and ifxetex.
2 \RequirePackage{keyval}%
3 \RequirePackage{ifpdf}%
4 \RequirePackage{ifvtex}%
5 \RequirePackage{ifxetex}%

    Internal switches are declared here.
6 \newif\ifGm@verbose
7 \newif\ifGm@landscape
8 \newif\ifGm@swap@papersize
9 \newif\ifGm@includehead
10 \newif\ifGm@includefoot
11 \newif\ifGm@includemp
12 \newif\ifGm@hbody
13 \newif\ifGm@vbody
14 \newif\ifGm@heightrounded
15 \newif\ifGm@showframe
16 \newif\ifGm@showcrop
17 \newif\ifGm@pass
18 \newif\ifGm@resetpaper
19 \newif\ifGm@layout
20 \newif\ifGm@newgm

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

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

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

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

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

\Gm@dimlist The token in which LATEX native dimensions can be stored.
32 \newtoks\Gm@dimlist

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

\ifGm@preamble The macro executes the option given as an argument only if it's specified in the preamble, as the
options of \usepackage and/or the argument of \geometry. Otherwise, the macro would print the
warning message and ignores the option setting.
34 \def\ifGm@preamble#1{%
35   \ifGm@newgm
36     \Gm@warning{‘#1’: not available in ‘\string\newgeometry’; skipped}%
37   \else
38     \expandafter\@firstofone
39   \fi}%

```

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

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

```

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

```

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

`\Gm@Dvscale`

```

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

```

`\Gm@dvi` The driver names.

`\Gm@dvi`

```

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

```

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

`\Gm@false`

```

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

```

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

`\Gm@orgph`

```

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

```

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

```

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

```

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

```

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

```

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

```

63 \def\Gm@restore{}%

```

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

```

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

```

```

85 \Gm@savelength{Gm@layoutvoffset}%
86 \Gm@saveboolean{@twocolumn}%
87 \Gm@saveboolean{@twoside}%
88 \Gm@saveboolean{@mparswitch}%
89 \Gm@saveboolean{@reversemargin}}%

\Gm@initnewgm The macro initializes the parameters for layout in \newgeometry.
90 \def\Gm@initnewgm{%
91 \Gm@passfalse
92 \Gm@swap@papersizefalse
93 \Gm@dimlist={}
94 \Gm@hbodyfalse
95 \Gm@vbodyfalse
96 \Gm@heightroundedfalse
97 \Gm@includeheadfalse
98 \Gm@includefootfalse
99 \Gm@includempfalse
100 \let\Gm@width\@undefined
101 \let\Gm@height\@undefined
102 \let\Gm@textwidth\@undefined
103 \let\Gm@textheight\@undefined
104 \let\Gm@lines\@undefined
105 \let\Gm@hscale\@undefined
106 \let\Gm@vscale\@undefined
107 \let\Gm@hmarginratio\@undefined
108 \let\Gm@vmarginratio\@undefined
109 \let\Gm@lmargin\@undefined
110 \let\Gm@rmargin\@undefined
111 \let\Gm@tmargin\@undefined
112 \let\Gm@bmargin\@undefined
113 \Gm@layoutfalse
114 \Gm@layouthoffset\z@
115 \Gm@layoutvoffset\z@
116 \Gm@bindingoffset\z@}%

\Gm@initall This initialization is called as soon as the package is load It's also called as soon as reset option is
specified.
117 \def\Gm@initall{%
118 \let\Gm@driver\@empty
119 \let\Gm@truedimen\@empty
120 \let\Gm@paper\@undefined
121 \Gm@resetpaperfalse
122 \Gm@landscapefalse
123 \Gm@verbosefalse
124 \Gm@showframefalse
125 \Gm@showcropfalse
126 \Gm@newgmfalse
127 \Gm@initnewgm}%

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

\Gm@unsetdriver The macro unsets the specified driver if it has been set.
130 \def\Gm@unsetdriver#1{%
131 \expandafter\ifx\csname Gm@#1\endcsname\Gm@driver\let\Gm@driver\@empty\fi}%

\Gm@setbool The macros for boolean option processing.
\Gm@setboolrev 132 \def\Gm@setbool{ \@dblarg\Gm@@setbool}%
133 \def\Gm@setboolrev{ \@dblarg\Gm@@setboolrev}%
134 \def\Gm@@setbool[#1]#2#3{\Gm@doif{#1}{#3}{\csname Gm@#2\Gm@bool\endcsname}}%
135 \def\Gm@@setboolrev[#1]#2#3{\Gm@doifelse{#1}{#3}%
136 {\csname Gm@#2\Gm@false\endcsname}{\csname Gm@#2\Gm@true\endcsname}}%

```

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

```

137 \def\Gm@doif#1#2#3{%
138   \lowercase{\def\Gm@bool{#2}}%
139   \ifx\Gm@bool\@empty
140     \let\Gm@bool\Gm@true
141   \fi
142   \ifx\Gm@bool\Gm@true
143   \else
144     \ifx\Gm@bool\Gm@false
145     \else
146       \let\Gm@bool\relax
147     \fi
148   \fi
149   \ifx\Gm@bool\relax
150     \Gm@warning{‘#1’ should be set to ‘true’ or ‘false’}%
151   \else
152     #3
153   \fi}%
154 \def\Gm@doifelse#1#2#3#4{%
155   \Gm@doif{#1}{#2}{\ifx\Gm@bool\Gm@true #3\else #4\fi}}%

```

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

```

156 \def\Gm@reverse#1{%
157   \csname ifGm@#1\endcsname
158   \csname Gm@#1false\endcsname\else\csname Gm@#1true\endcsname\fi}%

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

180 \def\Gm@detiv#1#2#3#4{% determine #4.
181   \expandafter\setlength\expandafter\@tempdima\expandafter
182     {\csname Gm@layout#1\endcsname}%

```

```

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

```

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

```

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

```

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

```

220 \def\Gm@detall#1#2#3#4{%
221   \@tempcnta\z@
222   \if#1h
223     \let\Gm@mratio\Gm@hmarginratio
224     \edef\Gm@Dmratio{\if@twoside\Gm@Dhratiotwo\else\Gm@Dhratio\fi}%
225   \else
226     \let\Gm@mratio\Gm@vmarginratio
227     \edef\Gm@Dmratio{\Gm@Dvratio}%
228   \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.

```

229   \if#1h
230     \ifx\Gm@lmargin\undefined\else\advance\@tempcnta4\relax\fi
231     \ifGm@hbody\advance\@tempcnta2\relax\fi
232     \ifx\Gm@rmargin\undefined\else\advance\@tempcnta1\relax\fi
233     \Gm@cntb\@tempcnta
234   \else
235     \ifx\Gm@tmargin\undefined\else\advance\@tempcnta4\relax\fi
236     \ifGm@vbody\advance\@tempcnta2\relax\fi
237     \ifx\Gm@bmargin\undefined\else\advance\@tempcnta1\relax\fi
238     \Gm@cntv\@tempcnta
239   \fi

```


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

```

240 \ifcase\@tempcnta
241 \if#1h
242 \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
243 \else
244 \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
245 \fi
246 \Gm@detiandiii{#2}{#3}{#4}%

```

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

```

247 \or
248 \ifx\Gm@mratio\@undefined
249 \if#1h
250 \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
251 \else
252 \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
253 \fi
254 \setlength\@tempdimc{\@nameuse{Gm@#4}}%
255 \Gm@detiandiii{#2}{#3}{#4}%
256 \expandafter\let\csname Gm@#2\endcsname\@undefined
257 \Gm@defbylen{#4}{\@tempdimc}%
258 \else
259 \Gm@setbyratio[f]{#1}{#4}{#3}%
260 \fi
261 \Gm@detiv{#2}{#3}{#4}{#2}%

```

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

```

262 \or\Gm@detiandiii{#2}{#3}{#4}%

```

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

```

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

```

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

```

264 \or
265 \ifx\Gm@mratio\@undefined
266 \if#1h
267 \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
268 \else
269 \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
270 \fi
271 \setlength\@tempdimc{\@nameuse{Gm@#3}}%
272 \Gm@detiandiii{#2}{#4}{#3}%
273 \expandafter\let\csname Gm@#2\endcsname\@undefined
274 \Gm@defbylen{#3}{\@tempdimc}%
275 \else
276 \Gm@setbyratio[b]{#1}{#3}{#4}%
277 \fi
278 \Gm@detiv{#2}{#3}{#4}{#2}%

```

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

```

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

```

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

```

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

```

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

```

281 \or\Gm@warning{Over-specification in ‘#1’-direction.%
282 ~~~~\spaces ‘#2’ (\@nameuse{Gm@#2}) is ignored}%
283 \Gm@detiv{#2}{#3}{#4}{#2}%
284 \else\fi}%

```

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

```

285 \def\Gm@clean{%
286 \ifnum\Gm@cnth<4\let\Gm@lmargin\@undefined\fi
287 \ifodd\Gm@cnth\else\let\Gm@rmargin\@undefined\fi
288 \ifnum\Gm@cntv<4\let\Gm@tmargin\@undefined\fi
289 \ifodd\Gm@cntv\else\let\Gm@bmargin\@undefined\fi
290 \ifGm@hbody\else

```

```

291 \let\Gm@hscale\@undefined
292 \let\Gm@width\@undefined
293 \let\Gm@textwidth\@undefined
294 \fi
295 \ifGm@vbody\else
296 \let\Gm@vscale\@undefined
297 \let\Gm@height\@undefined
298 \let\Gm@textheight\@undefined
299 \fi
300 }%

```

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

```

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

```

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

```

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

```

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

```

332 \def\Gm@magtooffset{%
333 \@tempdima=\mag\Gm@truedimen sp%
334 \@tempdimb=1\Gm@truedimen in%
335 \divide\@tempdimb\@tempdima
336 \multiply\@tempdimb\@m
337 \addtolength{\hoffset}{1\Gm@truedimen in}%
338 \addtolength{\voffset}{1\Gm@truedimen in}%
339 \addtolength{\hoffset}{-\the\@tempdimb}%
340 \addtolength{\voffset}{-\the\@tempdimb}}%

```

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

```

341 \def\Gm@setlength#1#2{%
342 \let\Gm@len=\relax\let\Gm@td=\relax
343 \edef\addtolist{\noexpand\Gm@dimlist=%
344 {\the\Gm@dimlist \Gm@len{#1}{#2}}}\addtolist}%

```

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

```

345 \def\Gm@expandlengths{%

```

```

346 \def\Gm@td{\Gm@truedimen}%
347 \def\Gm@len#1#2{\setlength{#1}{#2}}%
348 \the\Gm@dimlist}%

```

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

```

349 \def\Gm@setsize#1(#2,#3)#4{%
350 \let\Gm@td\relax
351 \expandafter\Gm@setlength\csname #1width\endcsname{#2\Gm@td #4}%
352 \expandafter\Gm@setlength\csname #1height\endcsname{#3\Gm@td #4}%
353 \ifGm@landscape\Gm@swap@papersizetrue\else\Gm@swap@papersizefalse\fi}%

```

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

```

354 \def\Gm@setpaper@ifpre#1{%
355 \ifGm@preamble{#1}\def\Gm@paper{#1}\@nameuse{Gm@#1}{paper}}}%

```

Various paper size are defined here.

```

356 \namedef{Gm@a0paper}#1{\Gm@setsize{#1}(841,1189){mm}}% ISO A0
357 \namedef{Gm@a1paper}#1{\Gm@setsize{#1}(594,841){mm}}% ISO A1
358 \namedef{Gm@a2paper}#1{\Gm@setsize{#1}(420,594){mm}}% ISO A2
359 \namedef{Gm@a3paper}#1{\Gm@setsize{#1}(297,420){mm}}% ISO A3
360 \namedef{Gm@a4paper}#1{\Gm@setsize{#1}(210,297){mm}}% ISO A4
361 \namedef{Gm@a5paper}#1{\Gm@setsize{#1}(148,210){mm}}% ISO A5
362 \namedef{Gm@a6paper}#1{\Gm@setsize{#1}(105,148){mm}}% ISO A6
363 \namedef{Gm@b0paper}#1{\Gm@setsize{#1}(1000,1414){mm}}% ISO B0
364 \namedef{Gm@b1paper}#1{\Gm@setsize{#1}(707,1000){mm}}% ISO B1
365 \namedef{Gm@b2paper}#1{\Gm@setsize{#1}(500,707){mm}}% ISO B2
366 \namedef{Gm@b3paper}#1{\Gm@setsize{#1}(353,500){mm}}% ISO B3
367 \namedef{Gm@b4paper}#1{\Gm@setsize{#1}(250,353){mm}}% ISO B4
368 \namedef{Gm@b5paper}#1{\Gm@setsize{#1}(176,250){mm}}% ISO B5
369 \namedef{Gm@b6paper}#1{\Gm@setsize{#1}(125,176){mm}}% ISO B6
370 \namedef{Gm@c0paper}#1{\Gm@setsize{#1}(917,1297){mm}}% ISO C0
371 \namedef{Gm@c1paper}#1{\Gm@setsize{#1}(648,917){mm}}% ISO C1
372 \namedef{Gm@c2paper}#1{\Gm@setsize{#1}(458,648){mm}}% ISO C2
373 \namedef{Gm@c3paper}#1{\Gm@setsize{#1}(324,458){mm}}% ISO C3
374 \namedef{Gm@c4paper}#1{\Gm@setsize{#1}(229,324){mm}}% ISO C4
375 \namedef{Gm@c5paper}#1{\Gm@setsize{#1}(162,229){mm}}% ISO C5
376 \namedef{Gm@c6paper}#1{\Gm@setsize{#1}(114,162){mm}}% ISO C6
377 \namedef{Gm@b0j}#1{\Gm@setsize{#1}(1030,1456){mm}}% JIS B0
378 \namedef{Gm@b1j}#1{\Gm@setsize{#1}(728,1030){mm}}% JIS B1
379 \namedef{Gm@b2j}#1{\Gm@setsize{#1}(515,728){mm}}% JIS B2
380 \namedef{Gm@b3j}#1{\Gm@setsize{#1}(364,515){mm}}% JIS B3
381 \namedef{Gm@b4j}#1{\Gm@setsize{#1}(257,364){mm}}% JIS B4
382 \namedef{Gm@b5j}#1{\Gm@setsize{#1}(182,257){mm}}% JIS B5
383 \namedef{Gm@b6j}#1{\Gm@setsize{#1}(128,182){mm}}% JIS B6
384 \namedef{Gm@ansipaper}#1{\Gm@setsize{#1}(8.5,11){in}}%
385 \namedef{Gm@ansibpaper}#1{\Gm@setsize{#1}(11,17){in}}%
386 \namedef{Gm@ansicpaper}#1{\Gm@setsize{#1}(17,22){in}}%
387 \namedef{Gm@ansidpaper}#1{\Gm@setsize{#1}(22,34){in}}%
388 \namedef{Gm@ansiepaper}#1{\Gm@setsize{#1}(34,44){in}}%
389 \namedef{Gm@letterpaper}#1{\Gm@setsize{#1}(8.5,11){in}}%
390 \namedef{Gm@legalpaper}#1{\Gm@setsize{#1}(8.5,14){in}}%
391 \namedef{Gm@executivepaper}#1{\Gm@setsize{#1}(7.25,10.5){in}}%
392 \namedef{Gm@screen}#1{\Gm@setsize{#1}(225,180){mm}}%

```

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

```

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

```

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

```

‘b[0-6]paper’ 395 \define@key{Gm}{a0paper}[true]{\Gm@setpaper@ifpre{a0paper}}%
‘b[0-6]j’ 396 \define@key{Gm}{a1paper}[true]{\Gm@setpaper@ifpre{a1paper}}%
‘ansi[a-e]paper’ 397 \define@key{Gm}{a2paper}[true]{\Gm@setpaper@ifpre{a2paper}}%
‘letterpaper’ 398 \define@key{Gm}{a3paper}[true]{\Gm@setpaper@ifpre{a3paper}}%
‘legalpaper’ 399 \define@key{Gm}{a4paper}[true]{\Gm@setpaper@ifpre{a4paper}}%
‘executivepaper’ 400 \define@key{Gm}{a5paper}[true]{\Gm@setpaper@ifpre{a5paper}}%
‘screen’

```

```

401 \define@key{Gm}{a6paper}[true]{\Gm@setpaper@ifpre{a6paper}}%
402 \define@key{Gm}{b0paper}[true]{\Gm@setpaper@ifpre{b0paper}}%
403 \define@key{Gm}{b1paper}[true]{\Gm@setpaper@ifpre{b1paper}}%
404 \define@key{Gm}{b2paper}[true]{\Gm@setpaper@ifpre{b2paper}}%
405 \define@key{Gm}{b3paper}[true]{\Gm@setpaper@ifpre{b3paper}}%
406 \define@key{Gm}{b4paper}[true]{\Gm@setpaper@ifpre{b4paper}}%
407 \define@key{Gm}{b5paper}[true]{\Gm@setpaper@ifpre{b5paper}}%
408 \define@key{Gm}{b6paper}[true]{\Gm@setpaper@ifpre{b6paper}}%
409 \define@key{Gm}{c0paper}[true]{\Gm@setpaper@ifpre{c0paper}}%
410 \define@key{Gm}{c1paper}[true]{\Gm@setpaper@ifpre{c1paper}}%
411 \define@key{Gm}{c2paper}[true]{\Gm@setpaper@ifpre{c2paper}}%
412 \define@key{Gm}{c3paper}[true]{\Gm@setpaper@ifpre{c3paper}}%
413 \define@key{Gm}{c4paper}[true]{\Gm@setpaper@ifpre{c4paper}}%
414 \define@key{Gm}{c5paper}[true]{\Gm@setpaper@ifpre{c5paper}}%
415 \define@key{Gm}{c6paper}[true]{\Gm@setpaper@ifpre{c6paper}}%
416 \define@key{Gm}{b0j}[true]{\Gm@setpaper@ifpre{b0j}}%
417 \define@key{Gm}{b1j}[true]{\Gm@setpaper@ifpre{b1j}}%
418 \define@key{Gm}{b2j}[true]{\Gm@setpaper@ifpre{b2j}}%
419 \define@key{Gm}{b3j}[true]{\Gm@setpaper@ifpre{b3j}}%
420 \define@key{Gm}{b4j}[true]{\Gm@setpaper@ifpre{b4j}}%
421 \define@key{Gm}{b5j}[true]{\Gm@setpaper@ifpre{b5j}}%
422 \define@key{Gm}{b6j}[true]{\Gm@setpaper@ifpre{b6j}}%
423 \define@key{Gm}{ansiapaper}[true]{\Gm@setpaper@ifpre{ansiapaper}}%
424 \define@key{Gm}{ansibpaper}[true]{\Gm@setpaper@ifpre{ansibpaper}}%
425 \define@key{Gm}{ansicpaper}[true]{\Gm@setpaper@ifpre{ansicpaper}}%
426 \define@key{Gm}{ansidpaper}[true]{\Gm@setpaper@ifpre{ansidpaper}}%
427 \define@key{Gm}{ansiepaper}[true]{\Gm@setpaper@ifpre{ansiepaper}}%
428 \define@key{Gm}{letterpaper}[true]{\Gm@setpaper@ifpre{letterpaper}}%
429 \define@key{Gm}{legalpaper}[true]{\Gm@setpaper@ifpre{legalpaper}}%
430 \define@key{Gm}{executivepaper}[true]{\Gm@setpaper@ifpre{executivepaper}}%
431 \define@key{Gm}{screen}[true]{\Gm@setpaper@ifpre{screen}}%

```

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

```

‘paperheight’ 432 \define@key{Gm}{paperwidth}{\ifGm@preamble{paperwidth}{%
‘papersize’ 433 \def\Gm@paper{custom}\Gm@setlength\paperwidth{#1}}}%
434 \define@key{Gm}{paperheight}{\ifGm@preamble{paperheight}{%
435 \def\Gm@paper{custom}\Gm@setlength\paperheight{#1}}}%
436 \define@key{Gm}{papersize}{\ifGm@preamble{papersize}{%
437 \def\Gm@paper{custom}\Gm@branch{#1}{paperwidth}{paperheight}}}%

```

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

```

‘layoutwidth’ 438 \define@key{Gm}{layout}{\Gm@layouttrue\@nameuse{Gm#1}{Gm@layout}}%
‘layoutheight’ 439 \let\KV@Gm@layoutname\KV@Gm@layout
‘layoutsizes’ 440 \define@key{Gm}{layoutwidth}{\Gm@layouttrue\Gm@setlength\Gm@layoutwidth{#1}}%
441 \define@key{Gm}{layoutheight}{\Gm@layouttrue\Gm@setlength\Gm@layoutheight{#1}}%
442 \define@key{Gm}{layoutsizes}{\Gm@branch{#1}{layoutwidth}{layoutheight}}%

```

‘landscape’ Paper orientation setting.

```

‘portrait’ 443 \define@key{Gm}{landscape}[true]{\ifGm@preamble{landscape}{%
444 \Gm@doifelse{landscape}{#1}%
445 {\ifGm@landscape\else\Gm@landscapetrue\Gm@reverse{swap@papersize}\fi}%
446 {\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}}%
447 \define@key{Gm}{portrait}[true]{\ifGm@preamble{portrait}{%
448 \Gm@doifelse{portrait}{#1}%
449 {\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}%
450 {\ifGm@landscape\else\Gm@landscapetrue\Gm@reverse{swap@papersize}\fi}}%

```

‘hscale’ These options can determine the length(s) of *corps intégral* giving *scale(s)* against the paper size.

```

‘vscale’ 451 \define@key{Gm}{hscale}{\Gm@hbodytrue\edef\Gm@hscale{#1}}%
‘scale’ 452 \define@key{Gm}{vscale}{\Gm@vbodytrue\edef\Gm@vscale{#1}}%
453 \define@key{Gm}{scale}{\Gm@branch{#1}{hscale}{vscale}}%

```

‘width’ These options give concrete dimension(s) of *corps intégral*. totalwidth and totalheight are aliases of width and height respectively.

```

‘total’
‘totalwidth’
‘totalheight’

```

```

454 \define@key{Gm}{width}{\Gm@hbodytrue\Gm@defbylen{width}{#1}}%
455 \define@key{Gm}{height}{\Gm@vbodytrue\Gm@defbylen{height}{#1}}%
456 \define@key{Gm}{total}{\Gm@branch{#1}{width}{height}}%
457 \let\KV@Gm@totalwidth\KV@Gm@width
458 \let\KV@Gm@totalheight\KV@Gm@height

‘textwidth’ These options directly sets the dimensions \textwidth and \textheight. body is an alias of text.
‘textheight’ 459 \define@key{Gm}{textwidth}{\Gm@hbodytrue\Gm@defbylen{textwidth}{#1}}%
‘text’ 460 \define@key{Gm}{textheight}{\Gm@vbodytrue\Gm@defbylen{textheight}{#1}}%
‘body’ 461 \define@key{Gm}{text}{\Gm@branch{#1}{textwidth}{textheight}}%
462 \let\KV@Gm@body\KV@Gm@text

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

‘includehead’ The options take the corresponding dimensions as part of corps.
‘includefoot’ 464 \define@key{Gm}{includehead}[true]{\Gm@setbool{includehead}{#1}}%
‘includeheadfoot’ 465 \define@key{Gm}{includefoot}[true]{\Gm@setbool{includefoot}{#1}}%
‘includemp’ 466 \define@key{Gm}{includeheadfoot}[true]{\Gm@doifelse{includeheadfoot}{#1}}%
‘includeall’ 467 {\Gm@includeheadtrue\Gm@includefoottrue}%
468 {\Gm@includeheadfalse\Gm@includefootfalse}%
469 \define@key{Gm}{includemp}[true]{\Gm@setbool{includemp}{#1}}%
470 \define@key{Gm}{includeall}[true]{\Gm@doifelse{includeall}{#1}}%
471 {\Gm@includeheadtrue\Gm@includefoottrue\Gm@includemptrue}%
472 {\Gm@includeheadfalse\Gm@includefootfalse\Gm@includempfalse}%

‘ignorehead’ These options exclude en-tête, pied de page and notes marginales when determining corps.
‘ignorefoot’ 473 \define@key{Gm}{ignorehead}[true]{%
‘ignoreheadfoot’ 474 \Gm@setboolrev[ignorehead]{includehead}{#1}}%
‘ignoremp’ 475 \define@key{Gm}{ignorefoot}[true]{%
‘ignoreall’ 476 \Gm@setboolrev[ignorefoot]{includefoot}{#1}}%
477 \define@key{Gm}{ignoreheadfoot}[true]{\Gm@doifelse{ignoreheadfoot}{#1}}%
478 {\Gm@includeheadfalse\Gm@includefootfalse}%
479 {\Gm@includeheadtrue\Gm@includefoottrue}%
480 \define@key{Gm}{ignoremp}[true]{%
481 \Gm@setboolrev[ignoremp]{includemp}{#1}}%
482 \define@key{Gm}{ignoreall}[true]{\Gm@doifelse{ignoreall}{#1}}%
483 {\Gm@includeheadfalse\Gm@includefootfalse\Gm@includempfalse}%
484 {\Gm@includeheadtrue\Gm@includefoottrue\Gm@includemptrue}%

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

‘hdivide’ The options are useful to specify partitioning in each direction of the paper.
‘vdivide’ 486 \define@key{Gm}{hdivide}{\Gm@parse@divide{#1}{lmargin}{width}{rmargin}}%
‘divide’ 487 \define@key{Gm}{vdivide}{\Gm@parse@divide{#1}{tmargin}{height}{bmargin}}%
488 \define@key{Gm}{divide}{\Gm@parse@divide{#1}{lmargin}{width}{rmargin}}%
489 \Gm@parse@divide{#1}{tmargin}{height}{bmargin}}%

‘lmargin’ These options set marges. left, inner, innermargin are aliases of lmargin. right, outer,
‘rmargin’ outermargin are aliases of rmargin. top and bottom are aliases of tmargin and bmargin respectively.
‘tmargin’ 490 \define@key{Gm}{lmargin}{\Gm@defbylen{lmargin}{#1}}%
‘bmargin’ 491 \define@key{Gm}{rmargin}{\Gm@defbylen{rmargin}{#1}}%
‘left’ 492 \let\KV@Gm@left\KV@Gm@lmargin
‘inner’ 493 \let\KV@Gm@inner\KV@Gm@lmargin
‘innermargin’ 494 \let\KV@Gm@innermargin\KV@Gm@lmargin
‘right’ 495 \let\KV@Gm@right\KV@Gm@rmargin
‘outer’ 496 \let\KV@Gm@outer\KV@Gm@rmargin
‘outermargin’ 497 \let\KV@Gm@outermargin\KV@Gm@rmargin
‘top’ 498 \define@key{Gm}{tmargin}{\Gm@defbylen{tmargin}{#1}}%
‘bottom’ 499 \define@key{Gm}{bmargin}{\Gm@defbylen{bmargin}{#1}}%
500 \let\KV@Gm@top\KV@Gm@tmargin
501 \let\KV@Gm@bottom\KV@Gm@bmargin

```

```

‘hmargin’   These options are shorthands for setting marges.
‘vmargin’   502 \define@key{Gm}{hmargin}{\Gm@branch{#1}{lmargin}{rmargin}}%
‘margin’    503 \define@key{Gm}{vmargin}{\Gm@branch{#1}{tmargin}{bmargin}}%
            504 \define@key{Gm}{margin}{\Gm@branch{#1}{lmargin}{tmargin}}%
            505 \Gm@branch{#1}{rmargin}{bmargin}}%

‘hmarginratio’ Options specifying the margin ratios.
‘vmarginratio’ 506 \define@key{Gm}{hmarginratio}{\edef\Gm@hmarginratio{#1}}%
‘marginratio’ 507 \define@key{Gm}{vmarginratio}{\edef\Gm@vmarginratio{#1}}%
‘hratio’      508 \define@key{Gm}{marginratio}{\Gm@branch{#1}{hmarginratio}{vmarginratio}}%
‘vratio’      509 \let\KV@Gm@hratio\KV@Gm@hmarginratio
‘ratio’       510 \let\KV@Gm@vratio\KV@Gm@vmarginratio
            511 \let\KV@Gm@ratio\KV@Gm@marginratio

‘hcentering’ Useful shorthands to place corps centered.
‘vcentering’ 512 \define@key{Gm}{hcentering}[true]{\Gm@doifelse{hcentering}{#1}}%
‘centering’  513 {\def\Gm@hmarginratio{1:1}}{}}%
            514 \define@key{Gm}{vcentering}[true]{\Gm@doifelse{vcentering}{#1}}%
            515 {\def\Gm@vmarginratio{1:1}}{}}%
            516 \define@key{Gm}{centering}[true]{\Gm@doifelse{centering}{#1}}%
            517 {\def\Gm@hmarginratio{1:1}\def\Gm@vmarginratio{1:1}}{}}%

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

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

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

‘headheight’ The direct settings of en-tête and/or pied de page dimensions.
‘headsep’    523 \define@key{Gm}{headheight}{\Gm@setlength\headheight{#1}}%
‘footskip’   524 \define@key{Gm}{headsep}{\Gm@setlength\headsep{#1}}%
‘head’       525 \define@key{Gm}{footskip}{\Gm@setlength\footskip{#1}}%
‘foot’       526 \let\KV@Gm@head\KV@Gm@headheight
            527 \let\KV@Gm@foot\KV@Gm@footskip

‘nohead’     They are only shorthands to set en-tête and/or pied de page to be Opt.
‘nofoot’     528 \define@key{Gm}{nohead}[true]{\Gm@doifelse{nohead}{#1}}%
‘noheadfoot’ 529 {\Gm@setlength\headheight\z@\Gm@setlength\headsep\z@}{}}%
            530 \define@key{Gm}{nofoot}[true]{\Gm@doifelse{nofoot}{#1}}%
            531 {\Gm@setlength\footskip\z@}{}}%
            532 \define@key{Gm}{noheadfoot}[true]{\Gm@doifelse{noheadfoot}{#1}}%
            533 {\Gm@setlength\headheight\z@\Gm@setlength\headsep\z@\Gm@setlength\footskip\z@}{}}%
            534 {\z@\Gm@setlength\footskip\z@}{}}%

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

‘marginparwidth’ They directly set native dimensions \marginparwidth and \marginparsep.
‘marginpar’   536 \define@key{Gm}{marginparwidth}{\Gm@setlength\marginparwidth{#1}}%
‘marginparsep’ 537 \let\KV@Gm@marginpar\KV@Gm@marginparwidth
            538 \define@key{Gm}{marginparsep}{\Gm@setlength\marginparsep{#1}}%

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

‘columnsep’ The option sets a native dimension \columnsep.
            541 \define@key{Gm}{columnsep}{\Gm@setlength\columnsep{#1}}%

```



```

‘hoffset’ The former two options set native dimensions \hoffset and \voffset. offset can set both of them
‘voffset’ with the same value.
‘offset’ 542 \define@key{Gm}{hoffset}{\Gm@setlength\hoffset{#1}}%
543 \define@key{Gm}{voffset}{\Gm@setlength\voffset{#1}}%
544 \define@key{Gm}{offset}{\Gm@branch{#1}{hoffset}{voffset}}%

‘layouthoffset’
‘layoutvoffset’ 545 \define@key{Gm}{layouthoffset}{\Gm@setlength\Gm@layouthoffset{#1}}%
‘layoutvoffset’ 546 \define@key{Gm}{layoutvoffset}{\Gm@setlength\Gm@layoutvoffset{#1}}%
547 \define@key{Gm}{layoutoffset}{\Gm@branch{#1}{layouthoffset}{layoutvoffset}}%

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

‘onecolumn’ This option has the reverse effect of twocolumn option.
550 \define@key{Gm}{onecolumn}[true]{%
551 \Gm@doifelse{onecolumn}{#1}{\@twocolumnfalse}{\@twocolumntrue}}%

‘reversemp’ The both options set \reversemargin.
‘reversemarginpar’ 552 \define@key{Gm}{reversemp}[true]{%
553 \Gm@doif{reversemp}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%
554 \define@key{Gm}{reversemarginpar}[true]{%
555 \Gm@doif{reversemarginpar}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%

‘driver’
556 \define@key{Gm}{driver}{\ifGm@preamble{driver}{%
557 \edef\@tempa{#1}\edef\@auto{auto}\edef\@none{none}%
558 \ifx\@tempa\empty\let\Gm@driver\relax\else
559 \ifx\@tempa\@none\let\Gm@driver\relax\else
560 \ifx\@tempa\@auto\let\Gm@driver\empty\else
561 \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’ 562 \define@key{Gm}{dvips}[true]{\ifGm@preamble{dvips}{%
‘pdftex’ 563 \Gm@doifelse{dvips}{#1}{\Gm@setdriver{dvips}}{\Gm@unsetdriver{dvips}}}%
‘xetex’ 564 \define@key{Gm}{dvipdfm}[true]{\ifGm@preamble{dvipdfm}{%
‘vtex’ 565 \Gm@doifelse{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{\Gm@unsetdriver{dvipdfm}}}%
566 \define@key{Gm}{pdftex}[true]{\ifGm@preamble{pdftex}{%
567 \Gm@doifelse{pdftex}{#1}{\Gm@setdriver{pdftex}}{\Gm@unsetdriver{pdftex}}}%
568 \define@key{Gm}{xetex}[true]{\ifGm@preamble{xetex}{%
569 \Gm@doifelse{xetex}{#1}{\Gm@setdriver{xetex}}{\Gm@unsetdriver{xetex}}}%
570 \define@key{Gm}{vtex}[true]{\ifGm@preamble{vtex}{%
571 \Gm@doifelse{vtex}{#1}{\Gm@setdriver{vtex}}{\Gm@unsetdriver{vtex}}}%

‘verbose’ The verbose mode.
572 \define@key{Gm}{verbose}[true]{\ifGm@preamble{verbose}{\Gm@setbool{verbose}{#1}}}%

‘reset’ The option cancels all the options specified before reset, except pass. mag ( $\neq$  1000) with truedimen
cannot be also reset.
573 \define@key{Gm}{reset}[true]{\ifGm@preamble{reset}{%
574 \Gm@doifelse{reset}{#1}{\Gm@restore@org\Gm@initall
575 \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.
576 \define@key{Gm}{resetpaper}[true]{\ifGm@preamble{resetpaper}{%
577 \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.
578 \define@key{Gm}{mag}{\ifGm@preamble{mag}{\mag=#1}}%

```


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

```
579 \define@key{Gm}{truedimen}[true]{\ifGm@preamble{truedimen}{%
580   \Gm@doifelse{truedimen}{#1}{\let\Gm@truedimen\Gm@true}%
581   {\let\Gm@truedimen\@empty}}}%
```

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

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

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

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

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

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

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

```
585 \def\Gm@setdefaultpaper{%
586   \ifx\Gm@paper\@undefined
587     \Gm@setsize{paper}{\strip@pt\paperwidth,\strip@pt\paperheight}{pt}%
588     \Gm@setsize{Gm@layout}{\strip@pt\paperwidth,\strip@pt\paperheight}{pt}%
589     \Gm@swap@papersizefalse
590   \fi}%
```

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

```
591 \def\Gm@adjustpaper{%
592   \ifdim\paperwidth>\p@else
593     \PackageError{geometry}{%
594       \string\paperwidth\space(\the\paperwidth) too short}{%
595       Set a paper type (e.g., ‘a4paper’).}%
596   \fi
597   \ifdim\paperheight>\p@else
598     \PackageError{geometry}{%
599       \string\paperheight\space(\the\paperheight) too short}{%
600       Set a paper type (e.g., ‘a4paper’).}%
601   \fi
602   \ifGm@swap@papersize
603     \setlength\@tempdima{\paperwidth}%
604     \setlength\paperwidth{\paperheight}%
605     \setlength\paperheight{\@tempdima}%
606   \fi
607   \ifGm@layout\else
608     \setlength\Gm@layoutwidth{\paperwidth}%
609     \setlength\Gm@layoutheight{\paperheight}%
610   \fi}%
```

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

```
611 \def\Gm@checkmp{%
612   \ifGm@includemp\else
613     \@tempcnta\z@\@tempcntb\@ne
614     \if@twocolumn
615       \@tempcnta\@ne
616     \else
617       \if@reversemargin
618         \@tempcnta\@ne\@tempcntb\z@
619       \fi
620     \fi
621     \@tempdima\marginparwidth
622     \advance\@tempdima\marginparsep
623     \ifnum\@tempcnta=\@ne
624       \@tempdimc\@tempdima
```

```

625     \setlength\@tempdimb{\Gm@lmargin}%
626     \advance\@tempdimc-\@tempdimb
627     \ifdim\@tempdimc>\z@
628         \Gm@warning{The marginal notes overrun the paper edge.^^J
629         \spaces Add \the\@tempdimc\space and more to the left margin}%
630     \fi
631 \fi
632 \ifnum\@tempcntb=\@ne
633     \@tempdimc\@tempdima
634     \setlength\@tempdimb{\Gm@rmargin}%
635     \advance\@tempdimc-\@tempdimb
636     \ifdim\@tempdimc>\z@
637         \Gm@warning{The marginal notes overrun the paper.^^J
638         \spaces Add \the\@tempdimc\space and more to the right margin}%
639     \fi
640 \fi
641 \fi}%

```

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

```

642 \def\Gm@adjustmp{%
643     \ifGm@includemp
644         \@tempdimb\marginparwidth
645         \advance\@tempdimb\marginparsep
646         \Gm@wd@mp\@tempdimb
647         \Gm@odd@mp\z@
648         \Gm@even@mp\z@
649         \if@twocolumn
650             \Gm@wd@mp2\@tempdimb
651             \Gm@odd@mp\@tempdimb
652             \Gm@even@mp\@tempdimb
653         \else
654             \if@reversemargin
655                 \Gm@odd@mp\@tempdimb
656             \if@mparswitch\else
657                 \Gm@even@mp\@tempdimb
658             \fi
659         \else
660             \if@mparswitch
661                 \Gm@even@mp\@tempdimb
662             \fi
663         \fi
664     \fi
665 \fi}%

```

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

```

666 \def\Gm@adjustbody{
667     \ifGm@hbody
668         \ifx\Gm@width\undefined
669             \ifx\Gm@hscale\undefined
670                 \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
671             \else
672                 \Gm@defbylen{width}{\Gm@hscale\Gm@layoutwidth}%
673             \fi
674         \fi
675         \ifx\Gm@textwidth\undefined\else
676             \setlength\@tempdima{\Gm@textwidth}%
677             \ifGm@includemp
678                 \advance\@tempdima\Gm@wd@mp
679             \fi
680             \edef\Gm@width{\the\@tempdima}%
681         \fi
682     \fi

```

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

```

683 \ifGm@vbody
684   \ifx\Gm@height\@undefined
685     \ifx\Gm@vscale\@undefined
686       \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
687     \else
688       \Gm@defbylen{height}{\Gm@vscale\Gm@layoutheight}%
689     \fi
690   \fi
691   \ifx\Gm@lines\@undefined\else
\topskip has to be adjusted so that the formula “ $\text{textheight} = (\text{lines} - 1) \times \text{baselineskip} + \text{topskip}$ ”
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.
692     \ifdim\topskip<\ht\strutbox
693       \setlength\@tempdima{\topskip}%
694       \setlength\topskip{\ht\strutbox}%
695       \Gm@warning{\noexpand\topskip was changed from \the\@tempdima\space
696         to \the\topskip}%
697     \fi
698     \setlength\@tempdima{\baselineskip}%
699     \multiply\@tempdima\Gm@lines
700     \addtolength\@tempdima{\topskip}%
701     \addtolength\@tempdima{-\baselineskip}%
702     \edef\Gm@textheight{\the\@tempdima}%
703   \fi
704   \ifx\Gm@textheight\@undefined\else
705     \setlength\@tempdima{\Gm@textheight}%
706     \ifGm@includehead
707       \addtolength\@tempdima{\headheight}%
708       \addtolength\@tempdima{\headsep}%
709     \fi
710     \ifGm@includefoot
711       \addtolength\@tempdima{\footskip}%
712     \fi
713     \edef\Gm@height{\the\@tempdima}%
714   \fi
715 \fi}%

```

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

```

716 \def\Gm@process{%
  If pass is set, the original dimensions and switches are restored and process is ended here.
717   \ifGm@pass
718     \Gm@restore@org
719   \else
720     \Gm@@process
721   \fi}%

```

The main processing macro.

```

722 \def\Gm@@process{%
723   \Gm@expandlengths
724   \Gm@adjustpaper
725   \addtolength\Gm@layoutwidth{-\Gm@bindingoffset}%
726   \Gm@adjustmp
727   \Gm@adjustbody
728   \Gm@detall{h}{width}{lmargin}{rmargin}%
729   \Gm@detall{v}{height}{tmargin}{bmargin}%

```

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

```

730   \setlength\textwidth{\Gm@width}%
731   \setlength\textheight{\Gm@height}%
732   \setlength\topmargin{\Gm@tmargin}%
733   \setlength\oddsidemargin{\Gm@lmargin}%
734   \addtolength\oddsidemargin{-1\Gm@truedimen in}%

```

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

```
735 \ifGm@includemp
736   \advance\textwidth-\Gm@wd@mp
737   \advance\oddsidemargin\Gm@odd@mp
738 \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`.

```
739 \if@mparswitch
740   \setlength\evensidemargin{\Gm@rmargin}%
741   \addtolength\evensidemargin{-1\Gm@truedimen in}%
742   \ifGm@includemp
743     \advance\evensidemargin\Gm@even@mp
744   \fi
745 \else
746   \evensidemargin\oddsidemargin
747 \fi
```

The `\bindingoffset` correction for `\oddsidemargin`.

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

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

```
750 \ifGm@includehead
751   \addtolength\textheight{-\headheight}%
752   \addtolength\textheight{-\headsep}%
753 \else
754   \addtolength\topmargin{-\headheight}%
755   \addtolength\topmargin{-\headsep}%
756 \fi
```

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

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

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

```
760 \ifGm@heightrounded
761   \setlength\@tempdima{\textheight}%
762   \addtolength\@tempdima{-\topskip}%
763   \@tempcnta\@tempdima
764   \@tempcntb\baselineskip
765   \divide\@tempcnta\@tempcntb
766   \setlength\@tempdimb{\baselineskip}%
767   \multiply\@tempdimb\@tempcnta
768   \advance\@tempdima-\@tempdimb
769   \multiply\@tempdima\tw@
770   \ifdim\@tempdima>\baselineskip
771     \addtolength\@tempdimb{\baselineskip}%
772   \fi
773   \addtolength\@tempdimb{\topskip}%
774   \textheight\@tempdimb
775 \fi
```

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

```
776 \advance\oddsidemargin\Gm@layouthoffset%
777 \advance\evensidemargin\Gm@layouthoffset%
778 \advance\topmargin\Gm@layoutvoffset%
779 \addtolength\Gm@layoutwidth{\Gm@bindingoffset}%
780 }% end of \Gm@@process
```

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

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

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

```

782 \ifx\Gm@driver\@empty
783   \typeout{*geometry* driver: auto-detecting}%
   \ifpdf is defined in ifpdf package in ‘oberdiek’ bundle.
784   \ifpdf
785     \Gm@setdriver{pdftex}%
786   \else
787     \Gm@setdriver{dvips}%
788   \fi

```

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

```

789 \ifvtex
790   \Gm@setdriver{vtex}%
791 \fi

```

\ifxetex is defined in ifxetex package written by Will Robertson.

```

792 \ifxetex
793   \Gm@setdriver{xetex}
794 \fi

```

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

```

795 \else
796   \ifx\Gm@driver\Gm@xetex %%
797     \ifxetex\else
798       \Gm@warning{Wrong driver setting: ‘xetex’; trying ‘pdftex’ driver}%
799       \Gm@setdriver{pdftex}
800     \fi
801   \fi
802   \ifx\Gm@driver\Gm@vtex
803     \ifvtex\else
804       \Gm@warning{Wrong driver setting: ‘vtex’; trying ‘dvips’ driver}%
805       \Gm@setdriver{dvips}%
806     \fi
807   \fi
808 \fi
809 \ifx\Gm@driver\relax
810   \typeout{*geometry* detected driver: <none>}%
811 \else
812   \typeout{*geometry* detected driver: \Gm@driver}%
813 \fi}%

```

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

```

814 \def\Gm@showparams#1{%
815   \ifGm@verbose\expandafter\typeout\else\expandafter\wlog\fi
816   {\Gm@logcontent{#1}}}%
817 \def\Gm@showdim#1{* \string#1=\the#1^^J}%
818 \def\Gm@showbool#1{\@nameuse{ifGm#1}#1\space\fi}%

```

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

```

819 \def\Gm@logcontent#1{%
820   *geometry* verbose mode - [ #1 ] result:^^J%
821   \ifGm@pass * pass: disregarded the geometry package!^^J%
822   \else
823     * driver: \if\Gm@driver<none>\else\Gm@driver\fi^^J%
824     * paper: \ifx\Gm@paper\@undefined<default>\else\Gm@paper\fi^^J%
825     * layout: \ifGm@layout<custom>\else<same size as paper>\fi^^J%
826     \ifGm@layout
827       * layout(width,height): (\the\Gm@layoutwidth,\the\Gm@layoutheight)^^J%
828     \fi
829     * layoutoffset:(h,v)=(\the\Gm@layoutoffset,\the\Gm@layoutvoffset)^^J%
830     \@ifundefined{Gm@lines}{* lines: \Gm@lines^^J}%
831     \@ifundefined{Gm@hmarginratio}{* hratio: \Gm@hmarginratio^^J}%
832     \@ifundefined{Gm@vmarginratio}{* vratio: \Gm@vmarginratio^^J}%
833     \ifdim\Gm@bindingoffset=\z@\else

```

```

834 * bindingoffset: \the\Gm@bindingoffset^^J\fi
835 * modes: %
836 \Gm@showbool{landscape}%
837 \Gm@showbool{includehead}%
838 \Gm@showbool{includefoot}%
839 \Gm@showbool{includemp}%
840 \if@twoside twoside\space\fi%
841 \if@mparswitch\else\if@twoside asymmetric\space\fi\fi%
842 \Gm@showbool{heightrounded}%
843 \ifx\Gm@truedimen\@empty\else truedimen\space\fi%
844 \Gm@showbool{showframe}%
845 \Gm@showbool{showcrop}%
846 ^^J%
847 * h-part:(L,W,R)=(\Gm@lmargin, \Gm@width, \Gm@rmargin)^^J%
848 * v-part:(T,H,B)=(\Gm@tmargin, \Gm@height, \Gm@bmargin)^^J%
849 \fi
850 \Gm@showdim{\paperwidth}%
851 \Gm@showdim{\paperheight}%
852 \Gm@showdim{\textwidth}%
853 \Gm@showdim{\textheight}%
854 \Gm@showdim{\oddsidemargin}%
855 \Gm@showdim{\evensidemargin}%
856 \Gm@showdim{\topmargin}%
857 \Gm@showdim{\headheight}%
858 \Gm@showdim{\headsep}%
859 \Gm@showdim{\topskip}%
860 \Gm@showdim{\footskip}%
861 \Gm@showdim{\marginparwidth}%
862 \Gm@showdim{\marginparsep}%
863 \Gm@showdim{\columnsep}%
864 * \string\skip\string\footins=\the\skip\footins^^J%
865 \Gm@showdim{\hoffset}%
866 \Gm@showdim{\voffset}%
867 \Gm@showdim{\mag}%
868 * \string\@twocolumn\if@twocolumn true\else false\fi^^J%
869 * \string\@twoside\if@twoside true\else false\fi^^J%
870 * \string\@mparswitch\if@mparswitch true\else false\fi^^J%
871 * \string\@reversemargin\if@reversemargin true\else false\fi^^J%
872 * (1in=72.27pt=25.4mm, 1cm=28.453pt)^^J}%

```

Macros for the page frames and cropmarks.

```

873 \def\Gm@cropmark(#1,#2,#3,#4){%
874 \begin{picture}(0,0)
875 \setlength\unitlength{1truemm}%
876 \linethickness{0.25pt}%
877 \put(#3,0){\line(#1,0){17}}%
878 \put(0,#4){\line(0,#2){17}}%
879 \end{picture}}%
880 \providecommand*{\vb@xt@{\vbox to}}%
881 \def\Gm@vrule{\vrule width 0.2pt height\textheight depth\z@}%
882 \def\Gm@hrule{\hrule height 0.2pt depth\z@ width\textwidth}%
883 \def\Gm@hruled{\hrule height\z@ depth0.2pt width\textwidth}%
884 \newcommand*{\Gm@vrules@mpi}{%
885 \hb@xt@{\@tempdima}{\llap{\Gm@vrule}\ignorespaces
886 \hskip \textwidth\Gm@vrule\hskip \marginparsep
887 \llap{\Gm@vrule}\hfil\Gm@vrule}}%
888 \newcommand*{\Gm@vrules@mpii}{%
889 \hb@xt@{\@tempdima}{\hskip-\marginparwidth\hskip-\marginparsep
890 \llap{\Gm@vrule}\ignorespaces
891 \hskip \marginparwidth\rlap{\Gm@vrule}\hskip \marginparsep
892 \llap{\Gm@vrule}\hskip\textwidth\rlap{\Gm@vrule}\hss}}%
893 \newcommand*{\Gm@pageframes}{%
894 \vb@xt@\z@{%
895 \ifGm@showcrop
896 \vb@xt@\z@{\vskip-1\Gm@truedimen in\vskip\Gm@layoutvoffset%

```

```

897 \hb@xt@\z@\hskip-1\Gm@truedimen in\hskip\Gm@layouthoffset%
898 \vb@xt@\Gm@layoutheight{%
899 \let\protect\relax
900 \hb@xt@\Gm@layoutwidth{\Gm@cropmark(-1,1,-3,3)\hfil\Gm@cropmark(1,1,3,3)}%
901 \vfil
902 \hb@xt@\Gm@layoutwidth{\Gm@cropmark(-1,-1,-3,-3)\hfil\Gm@cropmark(1,-1,3,-3)}%
903 \hss}%
904 \vss}%
905 \fi%
906 \ifGm@showframe
907 \iftwoside
908 \ifodd\count\z@
909 \let\@themargin\oddsidemargin
910 \else
911 \let\@themargin\evensidemargin
912 \fi
913 \fi
914 \moveright\@themargin%
915 \vb@xt@\z@{%
916 \vskip\topmargin\vb@xt@\z@{\vss\Gm@hrule}%
917 \vskip\headheight\vb@xt@\z@{\vss\Gm@hruled}%
918 \vskip\headsep\vb@xt@\z@{\vss\Gm@hrule}%
919 \@tempdima\textwidth
920 \advance\@tempdima by \marginparsep
921 \advance\@tempdima by \marginparwidth
922 \if@mparswitch
923 \ifodd\count\z@
924 \Gm@vrules@mpi
925 \else
926 \Gm@vrules@mpii
927 \fi
928 \else
929 \Gm@vrules@mpi
930 \fi
931 \vb@xt@\z@{\vss\Gm@hrule}%
932 \vskip\footskip\vb@xt@\z@{\vss\Gm@hruled}%
933 \vss}%
934 \fi%
935 }%

```

\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.

```

936 \def\ProcessOptionsKV{\ifnextchar[%]
937 {\@ProcessOptionsKV}{\@ProcessOptionsKV[]}}%
938 \def\@ProcessOptionsKV[#1]#2{%
939 \let\@tempa\@empty
940 \@tempcnta\z@
941 \if#1p\@tempcnta\@one\else\if#1c\@tempcnta\tw\fi\fi
942 \ifodd\@tempcnta
943 \edef\@tempa{\@optionlist{\@currname.\@current}}%
944 \else
945 \@for\CurrentOption:=\@classoptionslist\do{%
946 \@ifundefined{KV@#2\CurrentOption}%
947 {\edef\@tempa{\@tempa,\CurrentOption,}}}%
948 \ifnum\@tempcnta=\z@
949 \edef\@tempa{\@tempa,\@optionlist{\@currname.\@current}}%
950 \fi
951 \fi
952 \edef\@tempa{\noexpand\setkeys{#2}{\@tempa}}%
953 \@tempa
954 \AtEndOfPackage{\let\@unprocessedoptions\relax}}%
955 \def\Gm@setkeys{\setkeys{Gm}}%

```

\Gm@processconf \ExecuteOptions is replaced with \Gm@setkey to make it possible to deal with ‘key=value’ as its argument.


```

956 \def\Gm@processconfig{%
957   \let\Gm@origExecuteOptions\ExecuteOptions
958   \let\ExecuteOptions\Gm@setkeys
959   \InputIfFileExists{geometry.cfg}{-}{-}
960   \let\ExecuteOptions\Gm@origExecuteOptions}%

```

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

```

961 \Gm@save
962 \edef\Gm@restore@org{\Gm@restore}%
963 \Gm@initall

```

Processing config file.

```
964 \Gm@processconfig
```

The optional arguments to \documentclass are processed here.

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

Paper dimensions given by class default are stored.

```
966 \Gm@setdefaultpaper
```

The optional arguments to \usepackage are processed here.

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

Actual settings and calculation for layout dimensions are processed.

```
968 \Gm@process
```

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

```

969 \AtBeginDocument{%
970   \Gm@savelength{paperwidth}%
971   \Gm@savelength{paperheight}%
972   \edef\Gm@restore@org{\Gm@restore}%

```

The original paper size is used if resetpaper.

```

973   \ifGm@resetpaper
974     \edef\Gm@pw{\Gm@orgpw}%
975     \edef\Gm@ph{\Gm@orgph}%
976   \else
977     \edef\Gm@pw{\the\paperwidth}%
978     \edef\Gm@ph{\the\paperheight}%
979   \fi

```

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

```

980   \ifGm@pass\else
981     \ifnum\mag=\@m\else
982       \Gm@magtooffset
983       \divide\paperwidth\@m
984       \multiply\paperwidth\the\mag
985       \divide\paperheight\@m
986       \multiply\paperheight\the\mag
987     \fi
988   \fi

```

Checking the driver options.

```
989 \Gm@detectdriver
```

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

```

990 \ifx\Gm@driver\Gm@xetex
991   \ifundefined{pdfpagewidth}{-}{-}%
992   \setlength\pdfpagewidth{\Gm@pw}%
993   \setlength\pdfpageheight{\Gm@ph}%
994   \ifnum\mag=\@m\else
995     \ifx\Gm@truedimen\Gm@true
996       \setlength\paperwidth{\Gm@pw}%
997       \setlength\paperheight{\Gm@ph}%

```

```

998     \fi
999     \fi
1000  \fi

```

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

```

1001  \ifx\Gm@driver\Gm@pdftex
1002    \ifundefined{pdfpagewidth}{\%
1003      \setlength{pdfpagewidth}{\Gm@pw}%
1004      \setlength{pdfpageheight}{\Gm@ph}}%
1005    \ifnum\mag=\@m\else
1006      \@tempdima=\mag sp%
1007      \ifundefined{pdfhorigin}{\%
1008        \divide\pdfhorigin\@tempdima
1009        \multiply\pdfhorigin\@m
1010        \divide\pdfvorigin\@tempdima
1011        \multiply\pdfvorigin\@m}%
1012      \ifx\Gm@truedimen\Gm@true
1013        \setlength{paperwidth}{\Gm@pw}%
1014        \setlength{paperheight}{\Gm@ph}%
1015      \fi
1016    \fi
1017  \fi

```

With V_T_EX environment, V_T_EX variables are set here.

```

1018  \ifx\Gm@driver\Gm@vTEX
1019    \ifundefined{mediawidth}{\%
1020      \mediawidth=\paperwidth
1021      \mediaheight=\paperheight}%
1022    \ifvTEXdvi
1023      \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1024    \fi
1025  \fi

```

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

```

1026  \ifx\Gm@driver\Gm@dvips
1027    \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1028    \ifx\Gm@driver\Gm@dvips\ifGm@landscape
1029      \AtBeginDvi{\special{! /landplus90 true store}}%
1030    \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.

```

1031  \else\ifx\Gm@driver\Gm@dvipdfm
1032    \ifcase\ifx\AtBeginShipoutFirst\relax\@ne\else
1033      \ifx\AtBeginShipoutFirst\@undefined\@ne\else\z@\fi\fi
1034      \AtBeginShipoutFirst{\special{papersize=\the\paperwidth,\the\paperheight}}%
1035    \or
1036      \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1037    \fi
1038  \fi\fi

```

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

```

1039  \@tempswafalse
1040  \ifGm@showframe
1041    \@tempswatruetrue
1042  \else\ifGm@showcrop
1043    \@tempswatruetrue
1044  \fi\fi
1045  \if@tempswa
1046    \RequirePackage{atbegshi}%
1047    \AtBeginShipout{\setbox\AtBeginShipoutBox=\vbox{
1048      \baselineskip\z@skip\lineskip\z@skip\lineskiplimit\z@
1049      \Gm@pageframes\box\AtBeginShipoutBox}}%

```

```

1050 \fi
    The layout dimensions for \restoregeometry are saved at the end of the \AtBeginDocument.
1051 \Gm@save
1052 \edef\Gm@restore@pkg{\Gm@restore}%
    The package checks whether or not the marginpars overrun the page, if verbose and unless pass.
1053 \ifGm@verbose\ifGm@pass\else\Gm@checkmp\fi\fi
    \Gm@showparams puts the resulting parameters and dimensions into the log file. With verbose, they
    are shown on the terminal as well.
1054 \Gm@showparams{preamble}%
    The following lines free the memories no longer needed.
1055 \let\Gm@pw\relax
1056 \let\Gm@ph\relax
1057 }% end of \AtBeginDocument

```

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

```

1058 \newcommand{\geometry}[1]{%
1059 \Gm@clean
1060 \setkeys{Gm}{#1}%
1061 \Gm@process}%
1062 \@onlypreamble\geometry

```

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

```

1063 \DeclareRobustCommand\Gm@changelayout{%
1064 \setlength{\@colht}{\textheight}
1065 \setlength{\@colroom}{\textheight}%
1066 \setlength{\vsize}{\textheight}
1067 \setlength{\columnwidth}{\textwidth}%
1068 \if@twocolumn%
1069 \advance\columnwidth-\columnsep
1070 \divide\columnwidth\tw@%
1071 \@firstcolumntrue%
1072 \fi%
1073 \setlength{\hsize}{\columnwidth}%
1074 \setlength{\linewidth}{\hsize}}%

```

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

```

1075 \newcommand{\newgeometry}[1]{%
1076 \clearpage
1077 \Gm@restore@org
1078 \Gm@initnewgm
1079 \Gm@newgmtrue
1080 \setkeys{Gm}{#1}%
1081 \Gm@newgmfalse
1082 \Gm@process
1083 \ifnum\mag=\@m\else\Gm@magtooffset\fi
1084 \Gm@changelayout
1085 \Gm@showparams{newgeometry}}%

```

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

```

1086 \newcommand{\restoregeometry}{%
1087 \clearpage
1088 \Gm@restore@pkg
1089 \Gm@changelayout}%

```

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

```

1090 \newcommand*{\savegeometry}[1]{%
1091 \Gm@save
1092 \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.

```

1093 \newcommand*{\loadgeometry}[1]{%
1094   \clearpage
1095   \@ifundefined{Gm@restore@@#1}{%
1096     \PackageError{geometry}{%
1097       \string\loadgeometry : name ‘#1’ undefined}{%
1098       The name ‘#1’ should be predefined with \string\savegeometry}%
1099   }{\@nameuse{Gm@restore@@#1}%
1100   \Gm@changelayout}}%
1101 \end{package}

```

12 Config file

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

```

1102 <*config>
1103 %<<SAVE_INTACT
1104
1105 % Uncomment and edit the line below to set default options.
1106 %\ExecuteOptions{a4paper}
1107
1108 %SAVE_INTACT
1109 </config>

```

13 Sample file

Here is a sample document for the geometry package.

```

1110 <*samples>
1111 %<<SAVE_INTACT
1112 \documentclass[12pt]{article}% uses letterpaper by default
1113 \documentclass[12pt,a4paper]{article}% for A4 paper
1114 %-----
1115 % Edit and uncomment one of the settings below
1116 %-----
1117 % \usepackage{geometry}
1118 % \usepackage[centering]{geometry}
1119 % \usepackage[width=10cm,vscale=.7]{geometry}
1120 % \usepackage[margin=1cm, papersize={12cm,19cm}, resetpaper]{geometry}
1121 % \usepackage[margin=1cm,includeheadfoot]{geometry}
1122 % \usepackage[margin=1cm,includeheadfoot,includemp]{geometry}
1123 % \usepackage[margin=1cm,bindingoffset=1cm,twoside]{geometry}
1124 % \usepackage[hmarginratio=2:1, vmargin=2cm]{geometry}
1125 % \usepackage[hscale=0.5,twoside]{geometry}
1126 % \usepackage[hscale=0.5,asymmetric]{geometry}
1127 % \usepackage[hscale=0.5,heightrounded]{geometry}
1128 % \usepackage[left=1cm,right=4cm,top=2cm,includefoot]{geometry}
1129 % \usepackage[lines=20,left=2cm,right=6cm,top=2cm,twoside]{geometry}
1130 % \usepackage[width=15cm, marginparwidth=3cm, includemp]{geometry}
1131 % \usepackage[hdivide={1cm,,2cm}, vdivide={3cm,8in,,}, nohead]{geometry}
1132 % \usepackage[headsep=20pt, head=40pt,foot=20pt,includeheadfoot]{geometry}
1133 % \usepackage[text={6in,8in}, top=2cm, left=2cm]{geometry}
1134 % \usepackage[centering,includemp,twoside,landscape]{geometry}
1135 % \usepackage[mag=1414,margin=2cm]{geometry}
1136 % \usepackage[mag=1414,margin=2truecm,truedimen]{geometry}
1137 % \usepackage[a5paper, landscape, twocolumn, twoside,
1138 %   left=2cm, hmarginratio=2:1, includemp, marginparwidth=43pt,
1139 %   bottom=1cm, foot=.7cm, includefoot, textheight=11cm, heightrounded,
1140 %   columnsep=1cm,verbose]{geometry}
1141 %-----
1142 % No need to change below
1143 %-----

```

```

1144 \geometry{verbose,showframe}% the options appended.
1145 \usepackage{lipsum}% for dummy text of 150 paragraphs
1146 \newcommand\mynote{\marginpar[\raggedright
1147 A sample margin note in the left side.]}%
1148 {\raggedright A sample margin note.}}%
1149 \newcommand\myfootnote{\footnote{This is a sample footnote text.}}
1150 \begin{document}
1151 \lipsum[1-2]\mynote\lipsum[3-4]\mynote
1152 \lipsum[5-11]\mynote\lipsum[12]\myfootnote
1153 \lipsum[13-22]\mynote\lipsum[23-32]
1154 \end{document}
1155 %SAVE_INTACT
1156 </samples>
1157 %\tradfin

```