

Page Layout with Crop Marks

Zdeněk Wagner

<http://icebearsoft.euweb.cz>

Package date: 2008/12/26

Abstract

This package was developed as a typographers toolbox offering the most important features for everyday work. First it allows setting the paper size as well as the page layout. The next important feature is the ability of printing crop marks both with \TeX + dvips or (x)dvipdfm(x) and with pdf \TeX . Finally it is possible to reflect pages both horizontally and vertically.

Contents

1	Introduction	2
2	Usage	3
3	Option values	3
4	Paper size and orientation	3
4.1	Orientation	3
4.2	Paper size	3
4.2.1	Generic paper size	4
4.2.2	Standard paper sizes	4
5	Page layout options	5
5.1	Margins	5
5.1.1	Option MARGINS, standard 0 mm	5
5.1.2	Option TOPMARGIN, default 1 in	5
5.1.3	Option BOTMARGIN, standard TOPMARGIN	5
5.1.4	Option LEFTMARGIN, default -1 in	5
5.1.5	Option RIGHTMARGIN, default -1 in	5
5.2	Text dimensions	5
5.2.1	Option TEXTWIDTH, default -1 in	5
5.2.2	Option TEXTHEIGHT, default -1 in	5
5.2.3	Option HEADHEIGHT, default 0 mm	6
5.2.4	Option HEADSEP, default 0 mm	6
5.2.5	Option FOOTSKIP, default 0 mm	6
5.2.6	Option TOPSKIP, default <code>\topskip</code>	6
5.2.7	Option STRICTHEIGHT, default <i>false</i> , standard <i>true</i>	6
5.2.8	Option ADJUSTFOOTSKIP, default <i>true</i> , standard <i>true</i>	6
5.2.9	Option ADJUSTHEADSEP, default <i>true</i> , standard <i>true</i>	6
6	Crop marks	7
6.1	Basic crop marks options	7
6.1.1	Option ONLYCROPMARKS, default <i>false</i> , standard <i>true</i>	7
6.1.2	Option CROPMARKS, default <i>false</i> , standard <i>true</i>	7
6.1.3	Option NOCROPMARKS, default <i>true</i> , standard <i>true</i>	8

6.1.4	Option CROPLENGTH, default 5 mm	8
6.1.5	Option CROPGAP, default 5 mm	8
6.1.6	Option SPINE	8
6.1.7	Option XSPINE	8
6.1.8	Option FLAP	8
6.1.9	Options TRIM, XTRIM, YTRIM	8
6.1.10	Option CROPFRAME, default <i>false</i> , standard <i>true</i>	8
6.1.11	Option NOCROPFRAME, default <i>true</i> , standard <i>true</i>	8
6.1.12	Option CROPSTYLE	8
6.1.13	Option CROPTITLE	9
6.1.14	Option CROPSEPARATOR	9
6.1.15	Option PAGENUMBERFIRST, default <i>false</i> , standard <i>true</i>	9
6.1.16	Option PAGENUMBERLAST, default <i>true</i> , standard <i>true</i>	9
6.1.17	Option USEPAGENUMBERS, default <i>true</i> , standard <i>true</i>	9
6.1.18	Option NOPAGENUMBERS, default <i>false</i> , standard <i>true</i>	9
6.1.19	Option NOBLEEDCLIP, default <i>false</i> , standard <i>true</i>	9
6.2	Color support for crop marks	9
7	Calculation of missing dimensions	10
7.1	Calculation of the paper size	10
7.1.1	Paper height	11
7.1.2	Paper width	11
7.2	Calculation of page layout dimensions	11
7.2.1	Vertical dimensions	11
7.2.2	Horizontal dimensions	12
8	Page reflection	12
9	Useful macros	12
9.1	Userspace macros	12
9.2	Crop mark macros	13
10	Customizing crop styles	16
11	License	16

1 Introduction

Users will probably ask: “Why another page layout package? Is not GEOMETRY all what we need?” The answer to this question is not easy. The GEOMETRY package provides a lot of features, maybe even too many features. However, some features are missing. Packages for crop marks can also be found but correct positioning of the crop marks requires cooperation with the page layout package. It is therefore natural to integrate both these functions into a single package.

The package is useful also for preparation of book covers. It places the marks showing the position of a spine and optionally flaps. If you want to see whether the spine and the front and back cover contents are properly aligned, you can display the frames. Moreover, it is often necessary to create a proof of the cover while the book is not yet finished and the thickness of the spine is not known and can be only approximated. You can thus leave the paper width empty and the package will calculate it from the page width, the flap width and the spine thickness. Whenever the final value of the spine thickness is known, you change it in the list of options and the whole layout will adapt automatically.

The package is a result of a long-term development inspired by author’s everyday needs. It is a collection of macros that were manually copied from document to

document, later placed into private packages and finally it matured into a package prepared for general use.

Being a result of development it may happen that you may be in the same situation in which I was myself. The book was finished and the last task was to add the crop marks. Removing layout definition from a document and replacing it with a different package may cause errors and one more proof-reading is time consuming and expensive. This package therefore allows the page layout logic to be switched off and just add the crop marks provided the paper dimensions are correctly supplied. The details will be explained later when describing the package options.

2 Usage

The package is loaded simply by:

```
\usepackage[<options>]{zwpagelayout}
```

Remember that the options cannot be given later, all of them must be present either here or as options to the `\documentclass` command. The package will set some dimensions based upon `\normalsize`. The package that modifies font sizes must therefore be loaded before ZWPAGELAYOUT.

3 Option values

Some options accept a value. If the option is not specified, it may have a **default** value. For ease of use an option may be given without a value. It will then have a **standard** value. The default and standard values will be given in the option description.

4 Paper size and orientation

The first task of the package is to define paper size and orientation. Remember that the package is intended for use in real life where we cannot be limited to standard papers sizes. However, the standard paper sizes are used quite often and we must be able to use them too. The options were tested with pdf \TeX , DVIPS as well as (X)DVIPDFM(X).

4.1 Orientation

The package usually accepts the dimensions as they are, i.e. width and height of the page. The orientation cannot be changed. If you wish to have the possibility to change the orientation, you have to use this option. It is used automatically with all standard paper sizes (see section 4.2.2). You will rarely specify it yourself.

These options sets the paper orientation, `PORTRAIT` is the default. They can be used only if `ALLOWWIDTHHEIGHTSWITCHING` was specified, otherwise they are ignored.

`ALLOWWIDTH-
HEIGHTSWITCHING`

`PORTRAIT
LANDSCAPE`

4.2 Paper size

The package option defines the trimmed paper size. If the crop marks are added, the real paper size will be added. Dimensions `\paperwidth` and `\paperheight` will thus

contain the values that will subsequently be sent to the driver. If you for any reason wish to know the trimmed dimensions, you can calculate them as follows:

$$\begin{aligned}\text{trimmed width} &= \text{\paperwidth} - 2\text{\hoffset} \\ \text{trimmed height} &= \text{\paperheight} - 2\text{\voffset}\end{aligned}$$

4.2.1 Generic paper size

Option PAPER SIZE is used to define arbitrary paper size. You define it as a pair of two dimensions, the width and the height. The numbers are separated by a comma, therefore the specification must be enclosed within braces, e. g.

```
\usepackage[papersize={200mm,100mm}]{zwpagelayout}
```

The above written command sets the paper width to 200 mm and the paper height to 100 mm, i. e. the paper orientation will be landscape. Now consider the following:

```
\usepackage[papersize={200mm,100mm},
  AllowWidthHeightSwitching]{zwpagelayout}
```

Since the width/height switching is allowed and the default orientation is portrait (option PORTRAIT is the default), the width will be 100 mm and the height will be 200 mm.

You can achieve the same result as the first command by the lengthy code:

```
\usepackage[papersize={200mm,100mm},
  AllowWidthHeightSwitching,Landscape]{zwpagelayout}
```

You see that the command may become confusing. You should better not use ALLOWWIDTHHEIGHTSWITCHING together with PAPER SIZE.

If the PAPER SIZE option is not used, it defaults to the A4 size, i. e. to paper dimensions 210 mm × 297 mm.

If the book cover is being typeset, we want to calculate at least the width. The detailed explanation will be given in section 7. Here we put just as an example:

```
\usepackage[papersize={,200mm},spine=15mm,textwidth=14cm,
  margins]{zwpagelayout}
```

It is even possible to calculate both dimensions using e. g.

```
\usepackage[papersize,spine=15mm,textwidth=14cm,textheight=20cm,
  margins]{zwpagelayout}
```

4.2.2 Standard paper sizes

Options A0...A10, B0...B10, C0...C10 are used to select paper size according to the A, B or C series where the dimensions are rounded to integers in millimeters. For instance, the A6 size is 105 mm × 148 mm. The package also supports paper sizes EXECUTIVE (7.25 in × 10.5 in), LEGAL (8.5 in × 14 in), and LETTER (8.5 in × 11 in). If one of these options is used, ALLOWWIDTHHEIGHTSWITCHING and PORTRAIT are assumed. The page orientation may be switched by option LANDSCAPE (see section 4.1).

A0...
...C10
EXECUTIVE
LEGAL
LETTER

5 Page layout options

These options define the layout of the odd page. The layout of the even page is assumed to be a mirror image. The values of `\oddsidemargin` and `\evensidemargin` are calculated from the values of the options.

5.1 Margins

This set of options is used to define margin sizes. Remember that all margins are measured from the top left corner of the paper, not from the left border which is the DVI origin!

5.1.1 Option `MARGINS`, standard 0 mm

This option sets all margins (top, left, right, bottom) to the specified size. You will not be able to change some of the margin to another size. If you need different size for any margin, you have to set all of them separately and not use this option. Notice that almost everything may be calculated automatically, thus it is not necessary to specify all values. MARGINS

5.1.2 Option `TOPMARGIN`, default 1 in

This option sets the value of `\topmargin`. TOPMARGIN

5.1.3 Option `BOTMARGIN`, standard `TOPMARGIN`

This option specifies the size of the margin below the running footer. BOTMARGIN

5.1.4 Option `LEFTMARGIN`, default -1 in

This option sets the value of the margin at the left side of an odd page (and the margin at the right side of an even page). Negative value means that the value was not given. LEFTMARGIN

5.1.5 Option `RIGHTMARGIN`, default -1 in

This is the size of the right side of an odd page. Similarly the negative value means that the option was not specified. RIGHTMARGIN

5.2 Text dimensions

The following options are used to specify the height and width of the text.

5.2.1 Option `TEXTWIDTH`, default -1 in

This option specifies the width of the text without marginal notes. Its meaning is different for book covers. The option specifies the width of the cover without the extra spine (`XSPINE`), flap (`FLAP`) and trimming (`XTRIM` or `TRIM`). TEXTWIDTH

5.2.2 Option `TEXTHEIGHT`, default -1 in

This option specifies the total height of the text including the header and footer. TEXTHEIGHT

5.2.3 Option HEADHEIGHT, default 0 mm

This option specifies the height of the running head.

HEADHEIGHT

5.2.4 Option HEADSEP, default 0 mm

This is the value of `\headsep`.

HEADSEP

5.2.5 Option FOOTSKIP, default 0 mm

This option specifies the value of `\footskip`.

FOOTSKIP

5.2.6 Option TOPSKIP, default `\topskip`

This option enables changing the value of `\topskip`. It is rarely needed. However, the value of `\topskip` is usually less than `\baselineskip` which may cause problems in some page layout schemes. Since we have already written that font sizes must be defined before ZWPAGELAYOUT is loaded, you can safely specify `topskip=\baselineskip`.

TOPSKIP

5.2.7 Option STRICTHEIGHT, default *false*, standard *true*

The value of the L^AT_EX dimension `\textheight` is calculated from the values of the options and later adjusted so that together with `\topskip` it represents an integer number of lines. This may not be desirable in some cases. This option instructs the package not to perform any adjustment and accept the calculated value.

STRICTHEIGHT

5.2.8 Option ADJUSTFOOTSKIP, default *true*, standard *true*

If option STRICTHEIGHT is not given, the calculated value of `\textheight` is decreased to the nearest integer multiple of `\baselineskip`. The difference is then added to `\footskip` in order to preserve the total height. If ADJUSTFOOTSKIP is set to *false*, the difference is added to `\headsep` instead.

ADJUSTFOOTSKIP

The difference between these two options is shown in figure 1. Adjustment of `\footskip` was achieved by:

```
\usepackage[c8,margins=6mm,headheight=4mm,headsep=4mm,
  croplength=3mm,cropgap=2mm,
  cropmarks,cropframe,croptitle=Adjust\ footskip]{zwpagelayout}
```

The sample on the right side of the figure was created by:

```
\usepackage[c8,margins=6mm,headheight=4mm,headsep=4mm,adjustheadsep,
  croplength=3mm,cropgap=2mm,
  cropmarks,cropframe,croptitle=Adjust\ headsep]{zwpagelayout}
```

5.2.9 Option ADJUSTHEADSEP, default *true*, standard *true*

This is a complementary option to ADJUSTFOOTSKIP.

ADJUSTHEADSEP

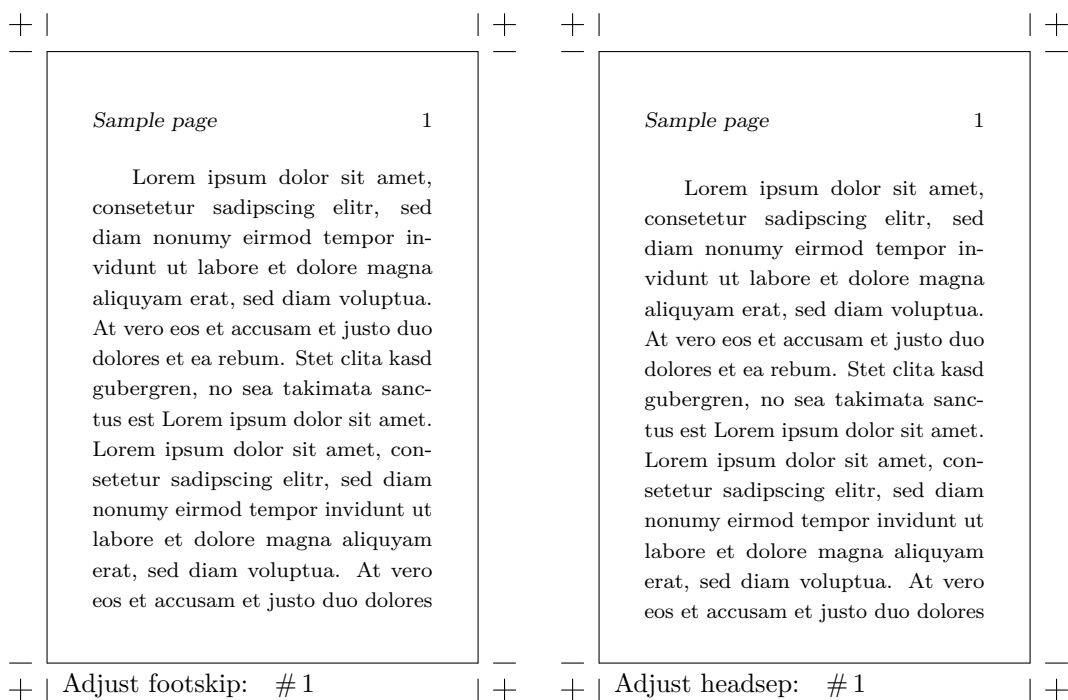


Figure 1: Compensation of `\textheight` by changing `\footskip` (on the left) and `\headsep` (on the right)

6 Crop marks

The crop marks must appear outside the print area. The package assumes that `\paperheight` and `\paperwidth` contain the page size after trimming. These dimensions will then be increased and the page will be shifted by setting `\hoffset` and `\voffset`. If you wish to print the crop marks using `ZWPAGELAYOUT`, you must not change values of `\hoffset` and `\voffset`.

The values of `\hoffset` and `\voffset` will be set to the sum of `CROPLENGTH` and `CROPGAP`, see the following text.

6.1 Basic crop marks options

Options described in this section define basic behaviour of the crop marks.

6.1.1 Option `ONLYCROPMARKS`, default *false*, standard *true*

It may happen that a document is already finished and proof-read and just the `ONLYCROPMARKS` crop marks have to be added. By specifying option `ONLYCROPMARKS` you instruct the package to ignore all page layout options and interpret the crop marks options only. You just have to ensure that `\paperheight` and `\paperwidth` are set to the dimensions of the trimmed page before the `ZWPAGELAYOUT` package is loaded and `\hoffset` and `\voffset` are left at their default values.

6.1.2 Option `CROPMARKS`, default *false*, standard *true*

This option asks for creation of the crop marks.

`CROPMARKS`

6.1.3 Option `NOCROPMARKS`, default *true*, standard *true*

This is a complementary option to `CROPMARKS`.

`NOCROPMARKS`

6.1.4 Option `CROPLENGTH`, default 5 mm

This option specifies the length of the crop mark.

`CROPLENGTH`

6.1.5 Option `CROPGAP`, default 5 mm

This is the space that must be left blank between the crop marks and the trimmed page.

`CROPGAP`

6.1.6 Option `SPINE`

This option is used for designing book covers. It specifies the width of the book spine. You can even request a zero width but negative values will have disastrous results.

`SPINE`

6.1.7 Option `XSPINE`

This option defines the extra space adjacent to the spine so that the book can be safely opened. This space should not be used for printing logos etc., because they will soon be damaged by frequent use of the book. The option is ignored unless `SPINE` is used.

`XSPINE`

6.1.8 Option `FLAP`

This is another option for book covers. It specifies the width of the flap. If `SPINE` is given, the book cover is supposed to have flaps of equal sizes on both sides. Sometimes you do not have flaps and the back cover remains empty, thus you do not like to prepare the design of it, you just wish to design the spine and the front cover. This can be achieved by specifying `FLAP` as the spine thickness and omitting `SPINE`.

`FLAP`

6.1.9 Options `TRIM`, `XTRIM`, `YTRIM`

The paper or canvas used for printing the book covers is folded to the inner side yet part of it will be visible. Such part is specified by `XTRIM` in horizontal direction and `YTRIM` in vertical direction. You can use `TRIM` to set both of them to the same value.

`TRIM`
`XTRIM`
`YTRIM`

6.1.10 Option `CROPFRAME`, default *false*, standard *true*

When designing a book cover we often wish to verify whether all elements are properly aligned. Option `CROPFRAME` can be used to print the frames. It is active only if `CROPMARKS` was specified.

`CROPFRAME`

6.1.11 Option `NOCROPFRAME`, default *true*, standard *true*

This is a complementary option to `CROPFRAME`.

`NOCROPFRAME`

6.1.12 Option `CROPSTYLE`

This option selects the style of the crop marks. Currently only the *default* style is defined, thus the option is only useful if you write your own extension.

`CROPSTYLE`

6.1.13 Option CROPTITLE

This option defines the text that should be printed on each page. It may e.g. be the title of the document. Remember that all spaces are gobbled when parsing the options. The spaces must therefore be specified as `_`.

6.1.14 Option CROPSEPARATOR

This is the separator between the title and the page number. Its default value is `: \quad`.

6.1.15 Option PAGENUMBERFIRST, default *false*, standard *true*

Each page contains the title followed by the separator and the page number. If you set this option to *true*, the order will be reversed.

6.1.16 Option PAGENUMBERLAST, default *true*, standard *true*

This is a complementary option to PAGENUMBERFIRST.

6.1.17 Option USEPAGENUMBERS, default *true*, standard *true*

This option requests printing the page numbers.

6.1.18 Option NOPAGENUMBERS, default *false*, standard *true*

This is a complementary option to USEPAGENUMBERS.

6.1.19 Option NOBLEEDCLIP, default *false*, standard *true*

The pages are clipped to the bleed box so that graphics does not collide with the crop marks. Since the package must work also with DVIPS, the PDF operators are not used and the crop marks area outside the bleed box is just overpainted with white rectangles. The COLOR package is thus required and will be loaded automatically. If you set NOBLEEDCLIP to *true*, the pages will not be clipped. It may be useful if you know that nothing extends to the crop marks area and the COLOR package may clash with other packages required by your document.

6.2 Color support for crop marks

The package is able to produce basic color support, namely it prints the names of separations. The color support is implemented via a few options. Color printing is performed using the COLOR package that is loaded automatically. The package does not use predefined color names.

Option COLOR asks for the color support. Without setting this option to *true* all other color options are silently ignored.

Option COLORMODEL defines the color model used. Its default value is `cm\yk`. You will rarely need to change it.

Option CROPCOLOR specifies the color to be used for the crop marks. Remember that the crop marks must be visible on all separations, thus it might not be sufficient to print them in black. As default their color is mixed of 100 % of all components of the current color model. Since the default model is `cm\yk`, the default value of this option is `{1,1,1,1}`. Notice that the syntax conforms to the requirements of the COLOR package.

Option `COLORS` assigns names to the color components of the current model. `COLORS` Specification of each color must be enclosed in curly brackets. The color name is followed by a colon and comma separated values conforming to the syntax of the `COLOR` package. It will be clear from the following examples.

Suppose that for some strange reason you prepare the document in the RGB space. Your specification will then be:

```
\usepackage[cropmarks,color,colormodel=rgb,cropcolor={1,1,1},
  colors={{RED:1,0,0},{GREEN:0,1,0},{BLUE:0,0,1}},
  croptitle=RGB\ example]{zwpagelayout}
```

Now we show a more realistic example. The document should be printed in custom colors. Since the true specification of custom colors requires much work and is rarely worth the trouble, we make use of the CMYK model. We will use cyan instead of Pantone 298 (blue), magenta instead of Pantone 213 (red), black will remain black and the yellow separation will be unused. The crop marks should not produce anything on the yellow separation and we should provide proper color names. The specification will therefore look as:

```
\usepackage[cropmarks,color,cropcolor={1,1,0,1},
  colors={{Pantone\ 298\ (blue):1,0,0,0},
    {Pantone\ 213\ (red):0,1,0,0},{Black:0,0,0,1}},
  croptitle=Example\ with\ custom\ colors]{zwpagelayout}
```

Notice that the `COLORMODEL` option was not specified. Its default value was used. The `CROPCOLOR` option left zero for the yellow separation.

7 Calculation of missing dimensions

The package can either calculate paper size from the page layout dimensions or calculate missing page layout dimensions if the paper size and some dimensions are known. It operates separately for the height and width. You can e.g. define the paper height and calculate the text height from it and the margins but specify all width layout dimensions and calculate the paper width.

When designing simple pages it is better to define the paper size and calculate some page layout dimensions. However, for bok covers it is recommended to calculate at least the paper width from the layout dimensions of the front cover, the spine and the flap width.

7.1 Calculation of the paper size

Remember that paper size can be calculated only if all page layout dimensions for the corresponding orientation (height, width) are specified. There is no diagnostics for warning you if some important options are missing, the result will just be wrong or the package may even report an error. All dimensions are considered *without* the area for the crop marks.

7.1.1 Paper height

Calculation of the paper height is very simple. The formula is:

$$\backslash\text{paperheight} = \text{TOPMARGIN} + \text{TEXTHEIGHT} + \text{BOTMARGIN} + 2 \text{ YTRIM}$$

Remember that `TEXTHEIGHT` includes also `\headheight`, `\headsep`, and `\footskip`. It is not necessary to specify `YTRIM`, the package checks the existence of this option and removes it from the formula if not given.

7.1.2 Paper width

Algorithm for calculating the paper width differs for simple pages and book covers. In the simple case the paper width is calculated as follows:

$$\backslash\text{paperwidth} = \text{LEFTMARGIN} + \text{TEXTWIDTH} + \text{RIGHTMARGIN}$$

The value of `\textwidth` will be equal to `TEXTWIDTH`. The same value will be stored in macro `\UserWidth`, see section 9.2.

If the book cover is designed, the `TEXTWIDTH` option refers to the width of the text at the front cover but `LEFTMARGIN` and `RIGHTMARGIN` are used to set `\oddsidemargin` and `\evensidemargin`. It is therefore better to set these options to zero, or alternatively to the same value as `FLAP` or `XTRIM` if the corresponding area will be left empty. The value of `\textwidth` will then be calculated from the final paper width by:

$$\backslash\text{textwidth} = \backslash\text{paperwidth} - \text{LEFTMARGIN} - \text{RIGHTMARGIN}$$

If the `FLAP` option is used without the `SPINE` option in order to emulate the front cover and spine with an empty back cover, the paper width is calculated as:

$$\backslash\text{paperwidth} = \text{LEFTMARGIN} + \text{FLAP} + \text{TEXTWIDTH} + \text{XTRIM} + \text{RIGHTMARGIN}$$

The value of `XTRIM` may be omitted. The value of `XSPINE` will be silently ignored. It is not allowed to have just the spine, the front cover and the flap while leaving the back cover empty.

The case of a book cover with the spine given is a bit more complicated:

$$\text{width} = \text{XSPINE} + \text{TEXTWIDTH} + \text{FLAP} + \text{XTRIM}$$

$$\backslash\text{paperwidth} = \text{LEFTMARGIN} + \text{SPINE} + 2 \text{ width} + \text{RIGHTMARGIN}$$

Options `XSPINE`, `FLAP`, and `XTRIM` need not be specified if these elements are not needed.

7.2 Calculation of page layout dimensions

Algorithm for calculating the page layout dimensions is intended for simple pages, not for book covers. Options `XTRIM`, `YTRIM`, `SPINE`, `XSPINE`, and `FLAP` are silently ignored but will be taken into account when producing the crop marks. You can still make use of this algorithm if you understand what you are doing and if you wish to do extra calculations yourself.

The dimensions may be overdetermined. In such a case the algorithm disregards one of the dimensions and calculates it.

7.2.1 Vertical dimensions

The algorithm first looks whether `TEXTHEIGHT` was given. If not it is assumed that the user wishes to have it calculated from the paper height and the margins. If the `BOTMARGIN` option was not set, it will be set to the same value as `TOPMARGIN`.

In the second step the package looks whether BOTMARGIN is defined, either by the user or from the previous step. If so, the text height is calculated, otherwise the bottom margin is calculated from the top margin and the text height. As a matter of fact, the bottom margin is never used by T_EX.

Finally the value of `\textwidth` is reduced by `\headheight`, `\headsep`, and `\footskip`. If STRICTHEIGHT is `\false`, the values are later adjusted according to options ADJUSTFOOTSKIP and ADJUSTHEADSEP so that `\textheight - \topskip` is an integer multiple of `\baselineskip`.

7.2.2 Horizontal dimensions

The sum of the horizontal dimensions must be equal to the paper width according to the relation:

$$\text{LEFTMARGIN} + \text{TEXTWIDTH} + \text{RIGHTMARGIN} = \text{\paperwidth}$$

If all three dimensions are specified, TEXTWIDTH is disregarded and calculated from the other dimensions. If any two dimensions are given, the missing one is calculated from the above formula. If only one dimension is given, it is assumed that both margins have the same size and the above formula is applied. It is even possible to omit all dimensions. In such a case the margins are assumed to have the same size as TOPMARGIN and the text width can thus be calculated.

8 Page reflection

We sometimes need to print the whole document as a mirror image. Although there are external tools that can provide such a job taking PDF or PS as input, it is useful to do everything in a single step. The package provides options REFLECTHORIZONTALLY for horizontal reflection and REFLECTVERTICALLY for vertical reflection, respectively. If you specify both, some drivers may interpret both of them and print rotated 180°, some interpret only one of them and some driver may get totally confused. Remember that these options are intended for printing only. The hypertext links made by the HYPERREF package will be wrong. If you wish to rotate parts of texts and preserve hyperlinks, use ROTATING instead.

REFLECT-
HORIZONTALLY
REFLECTVERTICALLY

9 Useful macros

The package offers two types of useful macros. The macros from the first group help in designing the document. However, as mentioned in the Introduction, the package may be deployed in already existing document just for adding the crop marks. In order to reduce the risk of conflicts with other packages these macros are unavailable if the ONLYCROPMARKS option is used. The second group contains macros that are primarily used in the crop mark generation. They are always visible.

9.1 Userspace macros

As written above, macros of this group are available only if the ONLYCROPMARKS option was not used. It is not planned to make any interface for providing them even with the ONLYCROPMARKS option. If the document is already finished, you do not need them. If you write a new document, it is preferable to use the whole package for defining the page layout. The macros will then be available.

T_EX should use integer arithmetic but some implementations, e. g. emT_EX, violate `\topskip`
`\Vcorr`

this rule. This makes processing the text faster but may have bad results. If you combine text with boxes and fixed size glues the height of which is an integer multiple of `\baselineskip`, the round-off errors may add a few scaled points, the page overflows and as a result is made one line shorter and underfull vbox is reported. For this reason tiny shrink is added to `\topskip`, its size is 11.00153pt minus 0.00763pt. Yet it may not be sufficient in some cases. We therefore provide vertical skip macro `\Vcorr` the size of which is shrinkable zero. The meaning of `\Vcorr` is `macro:->\vskip 10sp minus 50sp`.

We often need a vertical skip the size of which is a multiple of `\baselineskip`. Macro `\vb` serves this very purpose. It accepts an optional argument in square brackets which denotes the multiple of `\baselineskip`. The default value is 1. The command also contains compensation shrinkage. Some packages activate several characters, even those which could be used in numerical and dimension specifications. Hyphens and dots are temporarily deactivated within the optional argument of the `\vb` macro. You thus can comfortably specify negative as well as fractional dimensions.

L^AT_EX provides `\cleardoublepage` for moving the next text to the beginning of the odd page. However, you have no control over the page style of the inserted empty even page. This feature is enabled in the `\NewOddPage` macro. Its syntax is:

`\NewOddPage*[\langle style \rangle]`

Optional parameter `\langle style \rangle` defines the style of the empty even page that should be fed to the `\thispagestyle` command. The default is `empty`. The starred version displays a warning in the log file if an empty page was inserted.

This macro sets the contents of the message that should be displayed as a warning if an empty page was inserted as a result of `\NewOddPage*`. The macro acts as `\gdef`. This means that the definition is global and you can therefore have only one message unless you redefine it. The body of the message may contain macros. They will be expanded when the message is being written. You can use `\MessageBreak` in order to split the text into several lines.

Generally speaking, you should not start a chapter on an even page but there are cases when it is desirable. Imagine the situation when each chapter starts with a full page illustration on the left and with its title page on the right. In such cases you need to start at the even page but L^AT_EX does not contain any direct tool for doing it. This package provides the `\NewEvenPage` with the same syntaxe as `\NewOddPage`.

Similarly this macro serves for setting the message text that appears if an empty page was inserted as a result of the `\NewEvenPage*` macro.

9.2 Crop mark macros

T_EX offers zero-width horizontal boxes with contents overlapping to the right (`\rlap`) or to the left (`\llap`). The ZWPAGELAYOUT package frequently needs a zero-width box with the contents centered overlapping evenly to both sides. The package thus provides macro `\clap` and makes it available for the user.

It was written in the Introduction that the package allows to prepare the book cover before the exact dimensions are known. Later we just adapt the values of the package options and everything is recalculated automatically. As a matter of fact, it could not happen if we did not refer to selected dimensions in the document symbolically. Thus the values of some options are available in macros: `\CropFlap = FLAP`, `\CropSpine = SPINE`, `\CropXSpine = SPINE`, `\CropXtrim = XTRIM`, `\CropYtrim = YTRIM`, `\UserWidth = TEXTWIDTH`, `\UserLeftMargin = LEFTMARGIN`, `\UserRightMargin = RIGHTMARGIN`.

The use of these macros can be shown by an example. The book cover with flaps displayed in Figure 2 was prepared by the following code:

```
\input utf8-t1 % conversion from UTF-8 to T1 by encTeX
\documentclass{book}
\usepackage[papersize={,80mm},topmargin=5mm,
  botmargin,strictheight,
  leftmargin=0mm,flap=40mm,textwidth=50mm,spine=8mm,
  cropmarks,cropframe,croptitle=Cover]{zwpagelayout}
\usepackage[SC0,ISBN=978-80-7340-097-2]{ean13isbn}
\usepackage[T1]{fontenc}
\usepackage{lmodern,rotating}
\pagestyle{empty}
\parindent 0mm
\def\thePageNumber{Designed by Zdeněk Wagner}
\begin{document}
\small
\hbox to \textwidth{%
  \vbox to \textheight{\hsize \CropFlap \centering
    Back flap\vfill}\hss
  \vbox to \textheight{\hsize \UserWidth \centering
    Back cover\vfill \leavevmode \EANisbn}\hss
  \vbox to \textheight{\hsize \CropSpine \centering
    \vfill
    \begin{sideways}Spine\end{sideways}\vfill}\hss
  \vbox to \textheight{\hsize \UserWidth \centering
    Front cover\vfill \LaTeX}\hss
  \vbox to \textheight{\hsize \CropFlap \centering
    Front flap\vfill}}
\end{document}
```

Remember that these macros contain only the dimensions that were specified in the `\usepackage` command, not those that were calculated by the algorithm described in section 7.2. If a shortcut option was used, all real dimensions are defined, i.e. the `MARGINS` option fills all four margin options, `TOPMARGIN`, `LEFTMARGIN`, `RIGHTMARGIN`, `BOTMARGIN`, and the `TRIM` option fills both `XTRIM` and `YTRIM`. You can take advantage of it to make your document more general. You can e.g. define the book cover in two variants, with or without flaps, using this trick:

```
\ifcat$\CropFlap$
  % no flaps
\else
  \vbox to \textheight{\hsize\CropFlap
    Flap text
  }\hss
\fi
```

Macro `\thePageNumber` is used to print the page number together with the crop mark text. We have already shown that it may be suppressed by the `NOPAGENUMBERS` option. Another possibility is to redefine it. We can e.g. define the book title with the `CROPTITLE` option and use the text “Cover” instead of the page number when typesetting the cover. Usage of this macro is also demonstrated in Figure 2.



Figure 2: Sample of a book cover with flaps

10 Customizing crop styles

This section is intended for real \TeX pers. The package tries to provide a lot of options that enable to customize the default crop style. Try to live with them because here you are touching the very guts of the package. If you squeeze them badly, your document will suffer from strange problems the source of which will be difficult to trace. However, if you like tough challenges and your stomach is strong enough, then read on, but remember, you have been warned.

You ask for a different crop style by the `CROPSTYLE` option as described in section 6.1.12 on page 8. Assume that you load the package with:

```
\usepackage[cropmarks,cropstyle=special,...]{zwpagelayout}
```

You will then have to define macro `\cropstyle@special`. The package first patches the footer in all page styles. A zero-width, zero-height, zero-depth box is added to the beginning of the footer. The actual point is set to the lower left corner of the paper as defined by `\paperheight` and `\paperwidth`. If `NOBLEEDCLIP` was not used, the area outside the bleed box is erased. The current color is set according to `CROPCOLOR` option and the font is selected. Afterwards the `\cropstyle@special` macro is called.

As already noted, the intention is to discourage users from writing their own crop styles. We will therefore only mention that the package defines a few useful macros that can be used in the crop style definition. If your temptation to design your own crop style is really strong, you should better study the package internals yourself.

11 License

The package can be used and distributed according to the \LaTeX Project Public License version 1.3 or later the text of which can be found at the `License.txt` file in the `doc` directory or at <http://www.latex-project.org/lppl.txt>