

# The `crop` package

Melchior FRANZ

May 14, 2002

## Abstract

This article describes the `crop` package<sup>1</sup>, which provides different forms of cropmarks for trimming paper stacks, for camera alignment and for visualizing the page dimensions. There are options for centering the document page on the paper sheet, for marking the vertical and horizontal middle axis, for mounting pages on a physical sheet, for reflecting and inverting the whole document or printing it upside-down, and for suppressing either text or graphics output.

The package was originally developed for needs of the Austrian Red Cross/Federal Province of Vienna/Department of Radiation Protection.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>	<b>4</b>	<b>The macros</b>	<b>8</b>
			4.1	Preamble . . . . .	8
			4.2	Size options . . . . .	9
			4.3	Runtime options handling	10
			4.4	Axes and page info . . . .	11
			4.5	The marks . . . . .	12
			4.6	The kernel . . . . .	14
			4.7	Mounting . . . . .	15
			4.8	Page manipulation . . . .	16
			4.9	Color handling . . . . .	17
			4.10	The graphics commands .	18
			4.11	The device drivers . . . .	18
			4.12	Compatibility stuff . . . .	19
			4.13	Final settings . . . . .	19
			4.14	A <code>mount4</code> example . . . .	20
<b>2</b>	<b>How to use the package</b>	<b>2</b>			
	2.1 Conventional options . . .	2			
	2.2 Runtime options . . . . .	3			
	2.3 Loading . . . . .	5			
	2.4 Custom document paper size . . . . .	5			
	2.5 Custom printing paper sheet size . . . . .	6			
	2.6 Defining your own marks	6			
	2.7 The configuration file . . .	7			
<b>3</b>	<b>How the package works</b>	<b>8</b>			
	3.1 The kernel mechanism . .	8			
	3.2 Compatibility . . . . .	8			

## 1 Introduction

It is convenient to print documents for smaller logical paper sizes on paper of the printer's standard physical paper size. On the one hand this keeps from changing

---

<sup>1</sup>This file has version number 1.7, last revised 2002/05/14.

I'd like to thank ROLF NIEPRASCHK for his useful hints and suggestions, which influenced the package substantially. A big thank you also goes to WALTER SCHMIDT for his extensive tests and his expertise on compatibility issues with different devices.

paper stacks, on the other hand it allows printing close to the logical paper edge and even outside the logical page.

For trimming a whole paper stack or lining up the single pages on printing plates for photographic duplication a set of corner marks is required.

## 2 How to use the package

### 2.1 Conventional options

These options may only be used in the preamble and have to be stated as arguments to the `\usepackage` command (e. g. `\usepackage[frame]{crop}`).

**a0, a1, a2, a3, a4, a5, a6, b0, b1, b2, b3, b4, b5, b6, letter, legal, executive**

These options declare the printing paper dimensions. They are mandatory if the `center` option is used and optional if one of the `dvips`, `pdftex` and `vtex` options is used, in which case the paper size is passed to the respective DVI device driver.

**width, height** — Instead of using one of the pre-defined paper formats as described above, you can also set the paper dimensions directly. You can omit the ‘`true`’ specifier if you don’t plan to scale the document. Example:  
`\usepackage[cam,width=10truecm,height=13truecm]{crop}`

**center** This option centers the logical document page on the physical printer paper and requires therefore, that you declare the sheet size properly. Write e. g. `\usepackage[cam,a4,center]{crop}` to center a document of any size on ISO-A4 sheets. If no paper size is chosen you get an error message, but you can proceed nevertheless.

**landscape** — Use this option in addition to the `center` option if you want to center a document on *landscape oriented* paper. Note that it has nothing to do with the standard `landscape` option.

**dvips, pdftex, pdflatex, vtex, nodriver** — If you are working with `dvips`, `pdftex` or `vtex` you may want to pass the dimensions of the paper you are planning to print on to the respective program. Especially viewer programs like `gs` or `gv` make use of that *bounding box* information. This requires, of course, that you also choose one of the paper size options mentioned above (e. g. [`letter,dvips`]). If none of the driver options is given, `crop` automatically uses the `graphics` driver that your installation suggests (see the `graphics.cfg` file). The `nodriver` option suppresses this automatism. This is, for example, required if a document scaled with T<sub>E</sub>X’s `\mag` register is to be processed by `ghostscript`. `pdflatex` is a synonym for `pdftex`.

**mirror** This option reflects the whole document, provided that the DVI output driver handles POSTSCRIPT `\special`’s. It uses the standard graphics interfaces, if the `graphics` package could be found or the `color` package is included, or a matching interface file such as `dvips.def`. If no interface is defined, the package uses its internal, less portable macros.

**rotate** Rotates the document by 180° so that it appears upside-down. This may be useful to circumvent problems with printers, which do not print close

enough to the lower paper edge due to their paper feed mechanism. This option relies on PS, just like the `mirror` option.

**invert** Lets the whole document be printed white onto black background, if the `color` package can be loaded and the document is output with an output device that is capable of executing PS commands. All further color changing commands stated in the document are ignored. This option is ignored after a `notext` option request.

**notext** — This option uses the `color` package to turn text to white color, after which all further color switching commands are disabled. This makes the text disappear from the printout, although it remains in the output file. Note that neither cropmarks nor the info line are shown. See the description of the options `nographics` and `graphics` on page 4 for an explanation. This option is ignored after an `invert` option request.

## 2.2 Runtime options

These options may be used in the preamble like the ‘conventional’ options (see above), but also as arguments to the `\crop` command everywhere in the document (e. g. `\crop[frame]`).

**cam** This mode provides four different marks (see figure 1), one for each corner, which show the logical paper edges without touching them and can thus be printed on every page. These marks are mainly thought for camera alignment. The `\crop` command selects this mode if no other mode is requested.

**cross** This mode provides four two inch wide crosses (see figure 1), one at each corner that touch the logical paper edge. That’s the reason why they should be printed on an extra page to be used as a cover page while trimming the whole paper stack. (This is also the *Red Cross* mode ;-)

**frame** This mode draws a frame around the logical page and is mainly thought for visualizing the document page dimensions.

**off** This ‘option’ makes only sense in connection with the `\crop` command (i. e. at runtime). It disables all markings and is selected by default if the package is input without requesting any of the marks.

**odd, even** — Use these options to let the cropmarks be put on odd/even pages only. The marks must have been turned on before, using one of the options `cam`, `cross`, or `frame`, otherwise there will be no effect. Note that only the page number is considered. If you have two subsequent pages both with page number 1 and give the `odd` option, then both pages will have marks.

**axes, noaxes** — These options enable/disable the output of little marks which show the horizontal and vertical middle axis of the logical page and may be selected in addition to one of the above modes. These marks might be needed for punching. Notice that they are lost after trimming, since they lie outside the logical page. These marks are disabled by default.

**info, noinfo** — Show the page info consisting of filename, date, time, page number and page index on every sheet (see figure 1). The page index starts with #1 and is incremented with every page info line, so that it is more reliable than page numbers, which are not unique and may be negative or contain letters. It can also be seen as a cropmarks counter. This page information is enabled by default.

**font** The page info line uses `\normalfont` by default. If you are typesetting the document in non-latin glyphs or a decorative, but less legible font, you may want to request a specific font for that info. Just assign a font switching command to the `font` option parameter. This command may take one argument (like `\textsf{}`) or stand alone (like `\small`). You may even use more than one command, but note that just the last one is able to take the argument: `\crop[font=\small\textsf]`. You can, of course, still define a more complex command first, and assign that one: `\newcommand*\infofont[1]{\textcolor{blue}{\textsf{\small#1}}}`  
`\crop[font=\infofont]`

**mount1, mount2** — If more than one logical page is to be mounted on a physical sheet, you normally don't want marks to appear on the inner edges, where the pages touch each other. The `mount2` mode prints only the outer marks. There's also a `mount1` mode that is selected by default. These commands take a number as an optional argument serving as a page offset. Type `mount2` or `mount2=0` for odd pages right and `mount2=1` for odd pages left. Since further modes are likely to be document, driver, and printer dependent, it is up to you to implement them yourself. (See a `mount4` suggestion on page 20.)

**horigin, vorigin** — The top and left margin are by default 1 inch wide. This can be changed using the dimensions `\oddsidemargin`, `\evensidemargin` and `\topmargin`. It's more convenient, though, to let the `geometry` package define all these and further parameters. The options `horigin` and `vorigin` only move the marks and don't change the page contents. *Using these options is almost always a mistake, so use them only as a last resort!* Both options take a (mandatory) dimension. These dimensions describe the way from the reference point—the upper left corner of the text block—to the upper left corner of the page in a cartesian coordinate system. As both `horigin` and `vorigin` are by default `-1` inch, you would for example write `horigin=-.6in` to move the marks by 0.4 inch to the right.

**graphics, nographics** — Color printouts are often more expensive than black and white ones, while their text quality is sometimes reduced. Therefore it may be desirable to create two versions of a document, one with only text and one with only graphics. Now you can feed the concerned pages through a color printer to print the `notext` version, and then through a mono laser printer with the `nographics` version. The `graphics` option turns graphics on again. You may want to mark up every colored picture so that you can decide in the preamble, whether they shall be printed or not.

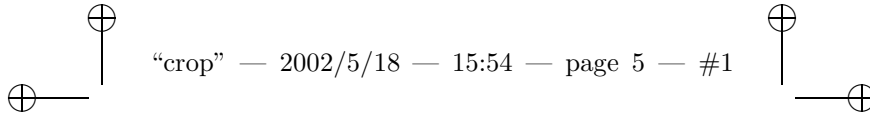


Figure 1: That’s what you see on top of a 9 cm wide document page when `cam` mode is requested: the marks, jobname, date, time, page number and cropmarks index.

```

% \newcommand* \colorgraphics{} % print them
\newcommand* \colorgraphics{\crop[nographics]} % don't!
...
{\colorgraphics
 \includegraphics{...}
 \caption{...}
}

```

### 2.3 Loading

Since all marks lie outside the logical page, the horizontal and vertical offset are to be set properly. Otherwise the marks are likely to be cut off by the DVI driver or the printer. Provided that you have declared the size of your printing paper, you can use the `center` option to center every logical page on the respective sheet. There’s, however, no harm in centering an A4 page on A4 paper, in which case both offsets are set to 0 pt (unless, of course, you have set `\mag ≠ 1000`).

<code>\documentclass[a5paper]{article}</code>	<code>\documentclass[a5paper]{article}</code>
<code>\usepackage[cam,a4,center]{crop}</code>	<code>\usepackage[a4,center]{crop}</code>
<code>\begin{document}</code>	<code>\begin{document}</code>
<code>...</code>	<code>...</code>
<code>\end{document}</code>	<code>\crop % or: \crop[<i>cross</i>], etc.</code>
	<code>...</code>
	<code>\end{document}</code>

You get corner markings at every page shipped out after a `cam`, `cross`, or `\crop` frame mode request until you turn them off by typing `\crop[off]`, or the actual grouping level ends. Typing `\crop` without argument(s) is equivalent to typing `\crop[cam,noaxes]`. Axis marks appear only together with one of the modes as listed above. If you only want one cover page for trimming, make sure that a page is actually output in the scope of `\crop`, for example:

```

\newpage
{\crop[cross,axes]\mbox{} \newpage}

```

### 2.4 Custom document paper size

The `crop` package respects any page layout that you specify by means of L<sup>A</sup>T<sub>E</sub>X dimensions. The following example uses the `geometry` package, which I strongly

recommend. Let's assume you want to print a CD booklet ( $4\frac{23}{32} \times 4\frac{3}{4}$  inch) on ISO-A4 paper:

```
\documentclass{article}
\usepackage[dvips=false, pdftex=false, vtex=false]{geometry}

\geometry{
  paperwidth=4.71875in,
  paperheight=4.75in,
  margin=2em,
  bottom=1.5em,
  nohead
}

\usepackage[cam,a4,center,dvips]{crop}

\begin{document}
...
\end{document}
```

Note that the `crop` package should always be requested after setting up the 'geometry'. See the `geometry` documentation for details. Always disable all of `geometry`'s driver options. While this isn't necessary in all cases, it doesn't hurt and it makes your document more portable. You never know how the local `geometry.cfg` file on other workstations looks like! Some of `crop`'s options depend very much on the used output device, thus, always specify the correct driver option when you load `crop.sty`.

## 2.5 Custom printing paper sheet size

If you want to use one of the `center`, `dvips`, `pdftex` or `vtex` options together with non-standard printing paper, you can set it via the `width` and `height` option, or simply add the respective paper definition to your `crop.cfg` file (see 2.7). Let's for example define a new `weird` paper format, whereby the first dimension shall describe the paper width. Don't forget to request `true` dimensions, otherwise you will get really weird results with scaled documents.

```
\DeclareOption{weird}{\CROP@size{12truecm}{34truecm}}
```

Now you can use your new printing paper format like the pre-defined ones.

```
\usepackage[frame,weird,center]{crop}
```

If you don't need that format regularly or don't want to depend on a `crop.cfg` file, then you might prefer to declare the dimensions in the document:

```
\usepackage[frame,width=12truecm,height=34truecm,center]{crop}
```

## 2.6 Defining your own marks

If you need a funny mode, you can easily define it with only a couple of macros. The `\cropdef` command defines the mode switch. It takes as arguments: the name of a macro providing the page info (optional; enclosed in brackets), four

`\cropdef`

macro names to be assigned to the upper left, the upper right, the lower left, and the lower right corner, each representing a `picture` with zero width and height, or `\relax`, and finally the mode name. The optional brackets may also be empty, if no page info is wanted, or contain the info code instead of a macro name.

```

\newcommand*\funnymarkA{%           % a little x
  \begin{picture}(0,0)
    \thinlines\unitlength1pt
    \put(-5,-5){\line(1,1){10}}
    \put(-5,5){\line(1,-1){10}}
  \end{picture}}

\newcommand*\funnymarkB{%           % a bullet
  \begin{picture}(0,0)
    \unitlength1pt
    \put(0,0){\circle*{5}}
  \end{picture}}

\newcommand*\funnyinfo{funny page info}
\cropdef[\funnyinfo]\relax\funnymarkA\relax\funnymarkB{funny}

```

You can select your own mode by typing `\crop[funny]`. Local definitions like these are ideally put into a local configuration file:

## 2.7 The configuration file

If you want to change the predefined settings or add new features, then create a file named `'crop.cfg'` and put it in a directory, where `TEX` can find it. This configuration file will then be loaded at the end of the `crop.sty` file, so you may redefine any settings or commands therein, select package options and even introduce new ones. But if you intend to give your documents to others, don't forget to give them the required configuration files, too! That's how such a file could look like:

```

% define a new printing paper size
\DeclareOption{special}{\CROP@size{22truecm}{37truecm}}

% introduce an option 'oldinfo', which restores the page
% information that was used in older package releases
\newcommand*\CROP@opt@oldinfo{%
\renewcommand*\CROP@@info{%
  \hskip\paperwidth\hskip12\p@
  \raise12\p@\hbox{\vbox{%
    \hbox{'\jobname'}\strut}%
    \hbox{\the\year/\the\month/\the\day\strut}%
    \hbox{page \thepage\strut}}}}}}

% make the internal time string (used in the page
% information) accessible in the whole document
\let\Time\CROP@time

```

```

% let's use a different font for the predefined page
% information (we could also have written
% \newcommand*\CROP@font{\small\textsf})
\crop[font=\small\textsf]
\endinput

```

## 3 How the package works

### 3.1 The kernel mechanism

$\text{\TeX}$  outputs a page via the `\shipout` command. The `crop` package redefines `\shipout` to insert the requested marks before it outputs the page contents. It is carefully designed to coexist peacefully with other packages, which use the same method (like the `everyshi` package by MARTIN SCHRÖDER, from whom I have in fact borrowed some ideas).

In addition to the cropmarks every page gets an info line containing the job-name, the current date and time, the page number and an index number printed on top. This line can be turned off (`noinfo`) and on (`info`) anywhere in the document.

### 3.2 Compatibility

The package works with all  $\text{\LaTeX} 2_{\epsilon}$  standard classes (tested with  $\text{\LaTeX} 2_{\epsilon}$  1997/12/01), it does not work with plain  $\text{\TeX}$ .

The `crop` package uses (and relies on) the internal  $\text{\LaTeX}$  tokens `\hb@xt@`, `\filename@parse`, `\@classoptionslist`, `\@ifundefined`, `\@height`, `\@depth`, `\filename@base` `\@width`, `\z@`, `\@ne`, `\z@skip`, `\p@`, `\c@page`, `\@namedef`, `\@nameuse`, `\strip@pt`, `\two@digits`, `\count@`, `\dimen@`, `\@for`, `\@empty`, `\@gobble` and `\@undefined`, all of which are expected to keep their current meaning in future  $\text{\LaTeX} 2_{\epsilon}$  releases. The `crop` package will, however, be supported at least for some years, so you needn't worry about it.

## 4 The macros

### 4.1 Preamble

```

\CROP@driver The options graphics and nographics depend on the graphics package, so we try
\CROP@font to load it here. We don't complain now if it can't be found, because we cannot say
yet, if one of these options is used in the document at all. Then we try to be clever
and look if there's a device dependent graphics driver already loaded, that we can
use. This information can, of course, be overridden by the driver options. The
\CROP@font macro is by default empty and can be changed via the font option.

```

```

1 (*package)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{crop}[2002/05/14 v1.7 cropmarks (mf)]
4 \ifx\stockwidth\undefined \newdimen\stockwidth \stockwidth\z@\fi
5 \ifx\stockheight\undefined \newdimen\stockheight \stockheight\z@\fi
6 \newcount\CROP@index \CROP@index\z@
7 \newcommand*\CROP@driver{}
8 \IfFileExists{graphics.sty}{%
9 \RequirePackage{graphics}%

```



```

10 \let\CROP@Gininclude@graphics\Gininclude@graphics
11 \ifx\Gin@driver\@empty\else
12     \filename@parse{\Gin@driver}%
13     \edef\CROP@driver{\filename@base}%
14 \fi
15 }-{}
16 \newcommand*\CROP@font{}

```

## 4.2 Size options

`\CROP@size` These options set different standard printing paper sizes, which are needed for centering and as hint for the `dvips`, `pdftex` or `vtex` program. Since the physical paper dimensions must not underlie a possible scaling, `true` dimensions are taken. `\CROP@opt@width` The landscape option exchanges the `\hoffset` and `\voffset` values. `\CROP@opt@height`

```

17 \newcommand*\CROP@size[2]{\stockwidth#1 \stockheight#2 }
18 \DeclareOption{landscape}{%
19     \def\CROP@size#1#2{\stockheight#1 \stockwidth#2 }
20 }
21 \DeclareOption{a0}{\CROP@size{841truecm}{1189truecm}}
22 \DeclareOption{a1}{\CROP@size{595truecm}{841truecm}}
23 \DeclareOption{a2}{\CROP@size{420truecm}{595truecm}}
24 \DeclareOption{a3}{\CROP@size{297truecm}{420truecm}}
25 \DeclareOption{a4}{\CROP@size{210truecm}{297truecm}}
26 \DeclareOption{a5}{\CROP@size{149truecm}{210truecm}}
27 \DeclareOption{a6}{\CROP@size{105truecm}{149truecm}}
28 \DeclareOption{b0}{\CROP@size{1000truecm}{1414truecm}}
29 \DeclareOption{b1}{\CROP@size{707truecm}{1000truecm}}
30 \DeclareOption{b2}{\CROP@size{500truecm}{707truecm}}
31 \DeclareOption{b3}{\CROP@size{353truecm}{500truecm}}
32 \DeclareOption{b4}{\CROP@size{250truecm}{353truecm}}
33 \DeclareOption{b5}{\CROP@size{176truecm}{250truecm}}
34 \DeclareOption{b6}{\CROP@size{125truecm}{176truecm}}
35 \DeclareOption{letter}{\CROP@size{8.5truein}{11truein}}
36 \DeclareOption{legal}{\CROP@size{8.5truein}{14truein}}
37 \DeclareOption{executive}{\CROP@size{7.25truein}{10.5truein}}
38 \newcommand\CROP@opt@width{\stockwidth\CROP@@}
39 \newcommand\CROP@opt@height{\stockheight\CROP@@}

```

`\CROP@center` The center option sets `\voffset` and `\hoffset` so that the document pages are centered on the printing paper sheet.

```

40 \DeclareOption{center}{\AtEndOfPackage{\CROP@center}}
41 \newcommand*\CROP@center{%
42     \ifdim\stockwidth=\z@
43         \PackageError{crop}{%
44             No printing paper size selected%
45         }{%
46             You have to select a paper size like ‘a4’ or
47             ‘height=20truecm,width=...’ to be able to use the
48             ‘center’ option.%
49         }%
50     \else
51         \voffset\stockheight
52         \advance\voffset-\paperheight

```

```

53     \voffset.5\voffset
54     \hoffset\stockwidth
55     \advance\hoffset-\paperwidth
56     \hoffset.5\hoffset
57 \fi
58 }

```

### 4.3 Runtime options handling

Every unknown option is passed to the macro `\CROP@execopt`.

```
59 \DeclareOption*{\CROP@execopt\CurrentOption}
```

`\crop` The `\crop` macro allows runtime option requests. Every argument of the optional argument list is passed to the macro `\CROP@execopt`. The options `cam` and `noaxes` are selected by default.

```

60 \newcommand*\crop[1][cam,noaxes]{%
61   \@for\CROP@@=#1\do{\CROP@execopt\CROP@@}%
62 }

```

`\CROP@execopt` Every execution of this macro with an argument  $n$  leads to the execution of a macro `\CROP@opt@n` or a warning if no such exists. Optional arguments (separated by an equal sign) are cut off and stored in `\CROP@@`. The macro tolerates even arguments for options that are not prepared to handle arguments (e.g. `cross=gabrage`), or more than one argument (e.g. `mount2=1=gabrage`). This is not a bug, it's a feature!

```

63 \newcommand*\CROP@execopt[1]{%
64   \def\CROP@##1=##2=##3\@nil{\def\CROP@{##1}\def\CROP@@{##2}}%
65   \expandafter\CROP@#1==\@nil%
66   \@ifundefined{CROP@opt@\CROP@}{%
67     \PackageError{crop}{%
68       Requested option ‘#1’ not provided
69     }{%
70       Note that the ‘*center’ options are obsolete. You have to
71       request\MessageBreak e.g. [a4,center] instead of
72       [a4center].
73     }%
74   }{%
75     \nameuse{CROP@opt@\CROP@}%
76   }%
77 }

```

`\cropdef` The `\cropdef` macro defines a mode switch (see section 2.6).

```

78 \newcommand*\cropdef[6][\CROP@@info]{%
79   \@namedef{CROP@opt@#6}{%
80     \CROP@on
81     \def\CROP@info{#1}%
82     \let\CROP@ulc#2
83     \let\CROP@urc#3
84     \let\CROP@llc#4
85     \let\CROP@lrc#5
86   }%
87 }

```

## 4.4 Axes and page info

`\CROP@@vaxis` The standard definitions for the axes option. The `\CROP@@vaxis` macro must have  
`\CROP@@haxis` zero height and depth.

```

88 \newcommand*\CROP@@vaxis{%
89   \hfil
90   \setbox\z@\hbox{%
91     \vtop{%
92       \hrule\@height12\p@\@depth-2\p@\@width.4\p@
93       \vskip\paperheight
94       \vskip4\p@
95       \hrule\@height\z@\@depth10\p@\@width.4\p@
96     }%
97   }%
98   \ht\z@\z@
99   \dp\z@\z@
100  \box\z@
101  \hfil
102 }
103 \newcommand*\CROP@@haxis{%
104   \vfil
105   \hb@xt@\paperwidth{%
106     \llap{%
107       \vrule\@height.2\p@\@depth.2\p@\@width10\p@\hskip2\p@
108     }%
109     \hfil
110     \rlap{%
111       \hskip2\p@\vrule\@height.2\p@\@depth.2\p@\@width10\p@
112     }%
113   }%
114   \vfil
115 }

```

`\CROP@time` This macro prints the jobname, the current date and time, the page number and  
`\CROP@@info` an index number at the top of the page.

```

\CROP@opt@font 116 \newcommand*\CROP@time{}
117 \bgroup
118   \count@\time
119   \divide\time60
120   \count@\ne\time
121   \multiply\time60
122   \advance\count@-\time
123   \xdef\CROP@time{\the\count@\ne:\two@digits{\count@}}
124 \egroup
125 \newcommand*\CROP@@info{%
126   \global\advance\CROP@index\@ne
127   \def\x{\discretionary}{-}{\hbox{\kern.5em---\kern.5em}}}%
128   \hskip10\p@
129   \advance\paperwidth-20\p@
130   \raise10\p@\vbox to\z@{%
131     \centering
132     \hsize\paperwidth
133     \vss
134     \normalfont

```

```

135     \let\protect\relax
136     \CROP@font{%
137         ‘‘\jobname’’\x
138         \the\year/\the\month/\the\day\x
139         \CROP@time\x
140         page\kern.5em\thepage\x
141         \#\the\CROP@index
142         \strut
143     }%
144 }%
145 }}
146 \newcommand*\CROP@opt@font{\let\CROP@font\CROP@@}

```

## 4.5 The marks

The following four macros provide different marks for the `cam` mode. They do not touch the logical page and can, thus, be printed on every single sheet.

`\CROP@@ulc` The `cam` mode corner mark for the upper left corner.

```

147 \newcommand*\CROP@@ulc{%
148     \begin{picture}(0,0)
149         \unitlength\p@\thinlines
150         \put(-30,0){\circle{10}}
151         \put(-30,-5){\line(0,1){10}}
152         \put(-35,0){\line(1,0){30}}
153         \put(0,30){\circle{10}}
154         \put(-5,30){\line(1,0){10}}
155         \put(0,35){\line(0,-1){30}}
156     \end{picture}%
157 }

```

`\CROP@@urc` The `cam` mode corner mark for the upper right corner.

```

158 \newcommand*\CROP@@urc{%
159     \begin{picture}(0,0)
160         \unitlength\p@\thinlines
161         \put(30,0){\circle{10}}
162         \put(30,-5){\line(0,1){10}}
163         \put(35,0){\line(-1,0){30}}
164         \put(0,30){\circle{10}}
165         \put(-5,30){\line(1,0){10}}
166         \put(0,35){\line(0,-1){30}}
167     \end{picture}%
168 }

```

`\CROP@@llc` The `cam` mode corner mark for the lower left corner.

```

169 \newcommand*\CROP@@llc{%
170     \begin{picture}(0,0)
171         \unitlength\p@\thinlines
172         \put(-30,0){\circle{10}}
173         \put(-30,-5){\line(0,1){10}}
174         \put(-35,0){\line(1,0){30}}
175         \put(0,-30){\circle{10}}
176         \put(-5,-30){\line(1,0){10}}

```

```

177     \put(0,-35){\line(0,1){30}}
178     \end{picture}%
179 }

```

`\CROP@@lrc` The cam mode corner mark for the lower right corner.

```

180 \newcommand*\CROP@@lrc{%
181     \begin{picture}(0,0)
182         \unitlength\p@\thinlines
183         \put(30,0){\circle{10}}
184         \put(30,-5){\line(0,1){10}}
185         \put(35,0){\line(-1,0){30}}
186         \put(0,-30){\circle{10}}
187         \put(-5,-30){\line(1,0){10}}
188         \put(0,-35){\line(0,1){30}}
189     \end{picture}%
190 }

```

`\CROP@opt@cam` Define the cam mode switch with four different marks.

```

191 \cropdef\CROP@@ulc\CROP@@urc\CROP@@llc\CROP@@lrc{cam}

```

`\CROP@@cross` This macro provides a 4 cm wide cross.

```

192 \newcommand*\CROP@@cross{%
193     \begin{picture}(0,0)
194         \unitlength2cm\thinlines
195         \put(-1,0){\line(1,0){2}}
196         \put(0,-1){\line(0,1){2}}
197     \end{picture}%
198 }

```

`\CROP@opt@cross` Define the cross mode switch with four times the same mark.

```

199 \cropdef\CROP@@cross\CROP@@cross\CROP@@cross\CROP@@cross{cross}

```

`\CROP@@frame` The frame mode draws a simple frame around the document page. The respective mark is designed to be used in the upper left corner. Since graphics commands expect numbers without dimensions, `\paperwidth` and `-height` are transformed to numbers (representing printer's points). This is done by stripping off the unit `pt`.

```

200 \newcommand*\CROP@@frame{%
201     \begin{picture}(0,0)
202         \unitlength\p@\thinlines
203         \put(0,0){\line(1,0){\strip@pt\paperwidth}}
204         \put(0,0){\line(0,-1){\strip@pt\paperheight}}
205         \put(\strip@pt\paperwidth,0){\line(0,-1){\strip@pt\paperheight}}
206         \put(0,-\strip@pt\paperheight){\line(1,0){\strip@pt\paperwidth}}
207     \end{picture}%
208 }

```

`\CROP@opt@frame` Define the frame mode switch with only one mark. The other corners may `\relax`.

```

209 \cropdef\CROP@@frame\relax\relax\relax{frame}

```

## 4.6 The kernel

`\CROP@shipout` These macros redefine the T<sub>E</sub>X primitive `\shipout` to insert the contents of the `\shipout` macro `\CROP@@@ship` on top of the box which contains the page contents ready for output, after which the original `\shipout` command is executed.

```
\CROP@shiplist 210 \let\CROP@shipout\shipout
\CROP@@@ship    211 \renewcommand*\shipout{
                212     \afterassignment\CROP@ship
                213     \setbox\@cclv=%
                214 }
                215 \newcommand*\CROP@ship{%
                216     \ifvoid\@cclv
                217     \expandafter\aftergroup
                218     \fi
                219     \CROP@@@ship
                220 }
                221 \newcommand*\CROP@shiplist{%
                222     \CROP@@@ship\box\@cclv
                223 }
                224 \newcommand*\CROP@@@ship{%
                225     \CROP@shipout\vbox{\CROP@shiplist}%
                226 }
```

`\CROP@shipadd` This macro adds a *page manipulation command* to the *shiplist*, which gets every ready page as argument.

```
227 \newcommand*\CROP@shipadd[1]{%
228     \bgroup
229     \toks@{\expandafter{\expandafter#1\expandafter{\CROP@shiplist}}}%
230     \xdef\CROP@shiplist{the\toks@}%
231     \egroup
232 }
```

`\CROP@kernel` `\CROP@kernel` essentially contains a `\vbox` with zero width and height. The `\CROP@opt@horigin` `\CROP@every` command—which normally equals `\relax`—allows to insert commands that modify the behavior of the selected mode (see the options `mount1` and `mount2`).

```
233 \newcommand*\CROP@kernel{%
234     \color@setgroup
235     \vbox to\z@{%
236         \vskip\CROP@vorigin
237         \hb@xt@\z@{%
238             \hskip\CROP@horigin
239             \CROP@every
240             \vbox to\paperheight{%
241                 \hb@xt@\paperwidth{%
242                     \setbox\z@\hbox{\normalfont\CROP@@@info}%
243                     \ht\z@\z@
244                     \dp\z@\z@
245                     \wd\z@\z@
246                     \box\z@
247                     \CROP@ulc
248                     \CROP@uedge
249                     \CROP@urc
```

```

250             }%
251             \CROP@ledge
252             \hb@xt@\paperwidth{%
253                 \CROP@llc
254                 \hfil
255                 \CROP@lrc
256             }%
257         }%
258         \hss
259     }%
260     \vss
261 }%
262 \color@endgroup
263 }
264 \newcommand*\CROP@opt@horigin{\let\CROP@horigin\CROP@@}
265 \newcommand*\CROP@opt@vorigin{\let\CROP@vorigin\CROP@@}

```

`\CROP@@@ship` These macros start and stop the kernel mechanism.

```

\CROP@on 266 \newcommand*\CROP@@@ship{}
\CROP@opt@off 267 \newcommand*\CROP@on{\let\CROP@@@ship\CROP@kernel}
\CROP@opt@odd 268 \newcommand*\CROP@opt@off{\let\CROP@@@ship\relax}
\CROP@opt@even 269 \newcommand*\CROP@opt@odd{%
270     \def\CROP@@@ship{\ifodd\c@page\CROP@kernel\fi}%
271 }
272 \newcommand*\CROP@opt@even{%
273     \def\CROP@@@ship{\ifodd\c@page\else\CROP@kernel\fi}%
274 }

```

`\CROP@@@info` Enable and disable the output of axis marks and page info.

```

\CROP@opt@info 275 \newcommand*\CROP@@@info{}
\CROP@opt@noinfo 276 \newcommand*\CROP@opt@info{\def\CROP@@@info{\CROP@info}}
\CROP@opt@axes 277 \newcommand*\CROP@opt@noinfo{\let\CROP@@@info\relax}
\CROP@opt@noaxes 278 \newcommand*\CROP@opt@axes{%
\CROP@uedge 279     \let\CROP@uedge\CROP@@vaxis
\CROP@ledge 280     \let\CROP@ledge\CROP@@haxis
281 }
282 \newcommand*\CROP@opt@noaxes{%
283     \let\CROP@uedge\hfil
284     \let\CROP@ledge\vfil
285 }

```

## 4.7 Mounting

`\CROP@opt@mount1` `\CROP@opt@mount2` Since `\newcommand` doesn't allow macro names to contain non-letters, we need a somewhat strange construction using `\csname`, `\endcsname`, and `\expandafter`. `\@namedef` would have worked, too, but it would not have made a check for re-definitions.

```

286 \expandafter\newcommand\expandafter*\csname CROP@opt@mount1\endcsname{%
287     \let\CROP@every\relax
288 }
289 \newcount\CROP@offset
290 \expandafter\newcommand\expandafter*\csname CROP@opt@mount2\endcsname{%
291     \CROP@offset=\ifx\CROP@@\emptyz\else\CROP@@\fi

```

```

292 \def\CROP@every{%
293 \count@\c@page
294 \advance\count@\CROP@offset
295 \ifodd\count@
296 \let\CROP@ulc\relax
297 \let\CROP@llc\relax
298 \else
299 \let\CROP@urc\relax
300 \let\CROP@lrc\relax
301 \let\CROP@info\relax
302 \fi
303 }%
304 }

```

## 4.8 Page manipulation

`\CROP@reflect` The mirror and rotate options add a macro to the *shiplist*, which then gets every output page and embeds it in a POSTSCRIPT environment. The PS commands are only interpreted by PS-aware output drivers and do not affect the outcome on all other drivers. Raw PS commands are output via the `graphics` package's POSTSCRIPT interface `\Gin@PS@raw`, or the built-in `\CROP@ps`. The latter issues a warning, because it is less portable.

```

305 \DeclareOption{mirror}{%
306 \AtBeginDocument{\CROP@shipadd\CROP@reflect\CROP@setps}
307 }
308 \newcommand*\CROP@reflect[1]{%
309 \vbox to\z@{%
310 \vskip\CROP@vorigin
311 \hb@xt@\z@{
312 \hskip\CROP@horigin
313 \CROP@ps{gsave currentpoint}%
314 \kern\paperwidth
315 \CROP@ps{currentpoint}%
316 \hss
317 }%
318 \vss
319 }%
320 \CROP@ps{translate -1 1 scale neg exch neg exch translate}%
321 \vbox{#1}%
322 \CROP@ps{grestore}%
323 }
324 \DeclareOption{rotate}{%
325 \AtBeginDocument{\CROP@shipadd\CROP@rotate\CROP@setps}
326 }
327 \newcommand*\CROP@rotate[1]{%
328 \hb@xt@\z@{%
329 \hskip\CROP@horigin
330 \vbox to\z@{%
331 \vskip\CROP@vorigin
332 \CROP@ps{gsave currentpoint}%
333 \kern\paperheight
334 \hb@xt@\z@{%
335 \kern\paperwidth

```



```

336         \CROP@ps{currentpoint}%
337         \hss
338     }%
339     \vss
340 }%
341 \hss
342 }%
343 \CROP@ps{translate 180 rotate neg exch neg exch translate}%
344 \vbox{#1}%
345 \CROP@ps{grestore}%
346 }
347 \newcommand*\CROP@setps{%
348     \ifx\Gin@PS@raw@undefined
349         \PackageWarning{crop}{internal PostScript interface used}%
350         \newcommand*\CROP@ps[1]{\special{ps: ##1}}%
351     \else
352         \PackageInfo{crop}{graphics/color PostScript interface used}{}%
353         \let\CROP@ps\Gin@PS@raw
354     \fi
355     \let\CROP@setps\relax
356 }

```

## 4.9 Color handling

`\CROP@invert` The invert option simply switches to black background and white text, after which it disables all color switching commands.

```

357 \DeclareOption{invert}{%
358     \AtEndOfPackage{\RequirePackage{color}}
359     \AtBeginDocument{\CROP@invert{black}}
360 }
361 \newcommand*\CROP@invert[1]{%
362     \ifx\color@undefined
363         \PackageWarning{crop}{%
364             The ‘color’ package could not be loaded, so I’m\MessageBreak
365             ignoring the ‘invert’ and ‘notext’ option
366         }%
367     \else
368         \pagecolor{#1}%
369         \color{white}%
370         \newcommand\CROP@@color[2][{}]{%
371             \DeclareRobustCommand\color{\CROP@@color}%
372             \DeclareRobustCommand\pagecolor{\CROP@@color}%
373             \DeclareRobustCommand\textcolor{\CROP@@color}%
374             \let\normalcolor\relax
375         \fi
376         \let\CROP@invert\relax
377 }
378 \DeclareOption{notext}{%
379     \AtEndOfPackage{\RequirePackage{color}}
380     \AtBeginDocument{\CROP@invert{white}}
381 }

```

## 4.10 The graphics commands

`\CROP@opt@nographics` The `nographics` option redefines the `\Ginclude@graphics` command from the graphics package, so that it outputs its argument as a phantom. This makes the image invisible but takes up the same amount of white space. The `graphics` option re-enables graphics.

```
382 \newcommand*\CROP@opt@nographics{%
383   \def\Ginclude@graphics##1{%
384     \phantom{%
385       \CROP@Ginclude@graphics{##1}%
386     }%
387   }%
388 }%
389 \newcommand*\CROP@opt@graphics{%
390   \let\Ginclude@graphics\CROP@Ginclude@graphics
391 }
```

## 4.11 The device drivers

`\CROP@init@dvips` These options are automatically selected (via definition in `graphics.cfg`) and can as well be chosen explicitly. They do nothing else than handing the given printer paper size over to the DVI driver. This wouldn't be necessary if there were some sort of standard. Sigh ...

```
392 \DeclareOption{vtex}{\def\CROP@driver{vtex}}
393 \DeclareOption{pdftex}{\def\CROP@driver{pdftex}}
394 \DeclareOption{pdflatex}{\def\CROP@driver{pdftex}}
395 \DeclareOption{dvips}{\def\CROP@driver{dvips}}
396 \DeclareOption{nodriver}{\def\CROP@driver{}}
397 \newcommand*\CROP@init@dvips{%
398   \PackageInfo{crop}{using dvips graphics driver}%
399   \AtBeginDvi{%
400     \ifdim\stockwidth=\z@
401     \else
402       \special{papersize=\the\stockwidth,\the\stockheight}%
403     \fi
404   }%
405 }
406 \newcommand*\CROP@init@pdftex{%
407   \PackageInfo{crop}{using pdf(la)tex graphics driver}%
408   \ifx\undefined\pdfpagewidth
409     \PackageWarning{crop}{implicit or explicit pdf(la)tex option
410       ignored:^^JThis doesn't seem to be pdftex!}%
411   \else
412     \AtBeginDocument{%
413       \ifdim\stockwidth=\z@
414       \else
415         \pdfpagewidth\stockwidth
416         \pdfpageheight\stockheight
417       \fi
418     }%
419   \fi
420 }
421 \newcommand*\CROP@init@vtex{%
```

```

422 \PackageInfo{crop}{using vtex graphics driver}%
423 \ifdim\stockwidth=\z@\else
424   \ifx\@undefined\mediawidth
425     \PackageWarning{crop}{implicit or explicit vtex option
426       ignored:^^This doesn't seem to be vtex!}%
427   \else
428     \AtBeginDocument{%
429       \mediawidth\stockwidth
430       \mediaheight\stockheight
431     }%
432   \fi
433 \fi
434 }

```

## 4.12 Compatibility stuff

`\CROP@compat` These options are just kept for compatibility reasons. They issue a warning that will become an error message in the next release. Finally they will be dropped altogether.

```

435 \newcommand*\CROP@compat{%
436   \PackageWarning{crop}{%
437     center options like 'a4center' are obsolete and
438     only\MessageBreak provided for compatibility reasons. They will
439     be removed\MessageBreak in future releases. Use the new options
440     'a4'\MessageBreak and 'center' separately instead.
441   }
442 }
443 \DeclareOption{landscapecenter}{%
444   \CROP@compat\ExecuteOptions{landscape,center}
445 }
446 \DeclareOption{a4center}{%
447   \CROP@compat\ExecuteOptions{a4,center}
448 }
449 \DeclareOption{a5center}{%
450   \CROP@compat\ExecuteOptions{a5,center}
451 }
452 \DeclareOption{b5center}{%
453   \CROP@compat\ExecuteOptions{b5,center}
454 }
455 \DeclareOption{lettercenter}{%
456   \CROP@compat\ExecuteOptions{letter,center}
457 }
458 \DeclareOption{legalcenter}{%
459   \CROP@compat\ExecuteOptions{legal,center}
460 }
461 \DeclareOption{executivecenter}{%
462   \CROP@compat\ExecuteOptions{executive,center}
463 }

```

## 4.13 Final settings

`\CROP@horigin` Switch off marks and axes, set one page per sheet, load the local configuration file,  
`\CROP@vorigin` and process the requested options. Finally: Exit.

2	1
0	3

Figure 2: Possible mount4 arrangement

Notice that we cannot simply use `\ExecuteOptions` to preselect options `off`, `noaxes`, `info`, and `mount1`, because it does not accept default options declared with `\DeclareOption*`. `\@nameuse` doesn't complain if the command sequence is undefined. We let this only be executed `\AtEndOfPackage`, because there are possibly commands from the center option in the queue that have to be processed first.

```

464 \newcommand*\CROP@horigin{-1truein}
465 \newcommand*\CROP@vorigin{-1truein}
466 \crop[off,noaxes,info,mount1]
467 \InputIfFileExists{crop.cfg}{%
468   \PackageInfo{crop}{Local config file crop.cfg used}
469 }{}
470 \ProcessOptions
471 \AtEndOfPackage{\@nameuse{CROP@init@\CROP@driver}}
472 \endinput
473 \endpackage

```

#### 4.14 A mount4 example

Since a `mount4` mode is likely to be subject to specific local needs, there's only a suggestion provided, which supports a page arrangement as shown in figure 2. First of all `\CROP@offset` is set to the value of the (optional) argument or zero. Then `\CROP@every` is defined first to set `\count@` to the page number increased by this offset:  $p = \text{pagenumber} + \text{offset}$ .

```

\expandafter\newcommand\expandafter*\csname CROP@opt@mount4\endcsname
  {\CROP@offset=\ifx\CROP@@\empty\z@\else\CROP@@\fi
  \def\CROP@every{\count@\c@page
  \advance\count@\CROP@offset

```

Now bits 0 and 1 are checked via `\ifodd` to get  $p$  modulo 4, after which the respective marks are deleted. The comments in the example use for simplicity C-Notation in which `'%` is the modulo or remainder operator, `'=='` the equal, and `'||'` the logical (inclusive) OR operator.

```

\ifodd\count@           %% if (p % 4 == 1 || p % 4 == 3)
  \let\CROP@ulc\relax\let\CROP@llc\relax
  \divide\count@2 \ifodd\count@   %%   if (p % 4 == 3)
  \let\CROP@urc\relax
  \let\CROP@inc\relax
\else                   %%   if (p % 4 == 1)
  \let\CROP@lrc\relax
\fi

```

```
\else                                     %% if (p % 4 == 0 || p % 4 == 2)
  \let\CROP@urc\relax\let\CROP@lrc\relax
  \let\CROP@info\relax
  \divide\count@2 \ifodd\count@          %% if (p % 4 == 2)
    \let\CROP@llc\relax
  \else                                     %% if (p % 4 == 0)
    \let\CROP@ulc\relax
  \fi
\fi}}
```