The \texttt{crop} package

Melchior FRANZ

May 20, 2003

Abstract

This article describes the \texttt{crop} package\footnote{This file has version number 2.19, last revised 2001/01/27. 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.}, which provides different forms of crop marks 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 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

1 Introduction 1 3.2 Dependencies . . . . . . 8

2 How to use the package 2 4 The implementation 9
  2.1 Conventional options . . . 2 4.1 Preamble . . . . . . . . . . . . 9
  2.2 Runtime options . . . . . 3 4.2 The device drivers . . . . . . 9
  2.3 Loading . . . . . . . . . . 5 4.3 Size options . . . . . . . 12
  2.4 Color support . . . . . . 6 4.4 Runtime options handling 12
  2.5 Custom document page size . . . . . . . . 6 4.5 Axes and page info . . . . . . 13
  2.6 Custom printing paper sheet size . . . . . . . . 6 4.6 The marks . . . . . . . . . . . 15
  2.7 Defining your own marks 7 4.7 The kernel . . . . . . . . . . . . . 16
  2.8 The configuration file . . . 7 4.8 Mounting . . . . . . . . . . . . . 18
  2.9 Defining your own marks 7 4.9 Page manipulation . . . . . . . 19
  4.10 Color handling . . . . . . . 20
  4.11 The graphics commands . . . . . . . . . . . . . . . . . . . . 22
  4.12 Final settings . . . . . . . 22

3 How the package works 8 4.13 A \texttt{mount4} example . . . . . . 23
  3.1 The kernel mechanism . . 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
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 (as in \usepackage[mirror]{crop}).

- \texttt{a0, a1, a2, a3, a4, a5, a6, b0, b1, b2, b3, b4, b5, b6, letter, legal, executive}
  These options declare the printing paper dimensions. One of them should be specified if the center option or one of the options \texttt{dvips}, \texttt{pdftex} and \texttt{vtex} is used. The size options do not define the logical document page size! See section 2.5 for how to achieve this.

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

- \texttt{center} This option centers the logical document page on the physical printer paper and therefore requires that you declare the sheet size properly. Write, for example, \usepackage[cam, a4, center]{crop} to center a document of any size on ISO-A4 sheets.

- \texttt{landscape} — Use this option in addition to the center option if you want to center a document on landscape oriented paper. It has nothing to do with \LaTeX{}’s \texttt{landscape} document class option.

- \texttt{dvips, pdftex, pdflatex, vtex, nodriver} — If you are working with \texttt{dvips}, \texttt{pdftex} or \texttt{vtex} you may want to pass the dimensions of the paper that you are planning to print on to the respective driver program. Especially viewer programs like \texttt{gs} or \texttt{gv} make use of this bounding box information. Unfortunately, this can’t be done in a generic way—there’s no standard. These options select driver specific methods to set the paper size and to rotate and reflect a page.

  The \texttt{crop} package tries to find out by itself which driver to use. You find its choice mentioned in the log file. Additionally, you can \texttt{suggest ([dvips])} or \texttt{enforce ([dvips!])} one of the drivers. In the latter case you only have to add an exclamation point to the driver option. The difference is, that a suggestion may get overruled by the package. Assume you have asked for \texttt{[dvips]}, but run the document file through \texttt{pdflatex}. In this case \texttt{crop} will automatically use the \texttt{pdflatex} driver. You can also force \texttt{crop} not to use any of the drivers by requesting the \texttt{nodriver} option. \texttt{pdflatex} is a synonym for \texttt{pdftex}.

- \texttt{mirror} This option reflects the whole document, provided that the selected output driver supports the \texttt{graphics} package’s \texttt{\reflectbox} command. It doesn’t have any effect on the DVI file.
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.

invert Lets the whole document be printed white onto black background, if the color package can be loaded and the document is output with a color aware device. All further color changing commands stated in the document are disabled. This option doesn’t invert pictures, nor does it really swap text and paper color. Red text on green will still become white text on black. invert is stronger than notext.

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. See the description of the options nographics and graphics on page 4 for an explanation. This option is ignored if option invert was also requested.

2.2 Runtime options
These options may be used in the preamble like the ‘conventional’ options (see section 2.1), but also as optional arguments to the \crop command everywhere in the document (as in \crop[frame]). Using this command without options implies \crop[cam,noaxes].

cam This mode provides four different marks (see figure 1), one for each corner. They indicate 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 4 cm wide crosses (see figure 1), one at each corner, that touch the logical paper edge. That’s why they should be printed on an extra page that will 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 crop marks be put on odd/even pages only. They automatically turn on cam marks if no other marks have been requested. Note that only the page number is considered. If you have two subsequent pages both with page number 1, and ask for the odd option, then both pages will have marks.

axes, noaxes — These options enable/disable the output of little marks that indicate the logical page’s horizontal and vertical middle axis and may be selected in addition to one of the main modes. These marks might be needed for punching. Note that they are lost after trimming, since they lie outside the logical page. These marks are disabled by default.
info, noinfo — Print the page information 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, hence being more reliable than page numbers, which are not unique and may be negative or contain letters. It can also be seen as a crop marks counter. Pages without crop marks aren’t counted. 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 like \textsf to the font option parameter, leaving the initial backslash away: [font=textsf]. This command may take one argument (like \textsf{}) or stand alone (like \small). You can, of course, define a more complex command first, and assign that one: \newcommand*\infofont\{\textcolor{blue}{\textsf{\small#1}}\}\crop[font=infofont]

color You can set the color of crop marks, axes and info text with this option, if the color package could be loaded. The option takes only color names, as in [color=red]. See the color package documentation for how to define custom colors.

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 option prints only the outer marks. There’s also a mount1 option that is selected by default. These commands take a number as an optional argument serving as 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 23.)

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 to a color printer to print the notext version, and then to a mono laser printer with the nographics version. The graphics option turns graphics
“crop” — 2003/6/21 — 0:32 — page 5 — #1

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 crop marks index.

on again. You may want to mark up all colored pictures so that you can decide in the preamble, whether they shall be printed or not.

\%\newcommand*{\colorgraphic} % print them;
\newcommand*{\colorgraphic}{\crop[no graphicals]} % don’t!
\...
{\colorgraphic
#includegraphics{...}
\caption{...}
}

2.3 Loading

Since all marks lie outside the logical page, the horizontal and vertical offset should 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 \neq 1000).

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

You get corner markings at every page shipped out after a cam, cross, or frame mode request until you turn them off by typing \crop[off], or the actual grouping level ends. Typing \crop without argument is equivalent to typing \crop[cam, no axes]. 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 Color support

The \texttt{crop} package always tries to load the \texttt{color} package. You can change the color of the physical page as usual by using the \texttt{\pagecolor} command in the preamble. But after that, within the \texttt{document} environment, \texttt{\pagecolor} is redefined to only color the logical page. The color of crop marks, axes marks and page info can independently be set via the \texttt{color} option. The options \texttt{invert} and \texttt{notext} override any color settings.

2.5 Custom document page size

The \texttt{crop} package respects any page layout that you specify by means of \LaTeX\ dimensions. The following example uses the \texttt{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:

\begin{verbatim}
\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}
\end{verbatim}

Note that the \texttt{crop} package should always be requested after setting up the ‘geometry’. See the \texttt{geometry} documentation for details. Always disable all of \texttt{geometry}’s driver options. While this isn’t necessary in every case, it doesn’t hurt and it makes your document more portable. You never know how the local \texttt{geometry.cfg} file on other workstations looks like!

2.6 Custom printing paper sheet size

If you want to use one of the \texttt{center}, \texttt{dvips}, \texttt{pdftex} or \texttt{vtex} options together with non-standard printing paper, you can set it via the \texttt{width} and \texttt{height} option, or simply add the respective paper definition to your \texttt{crop.cfg} file (see 2.8). Let’s for example define a new \texttt{weird} paper format, whereby the first dimension shall describe the paper width. Don’t forget to request \texttt{true} dimensions, otherwise you will get really weird results with scaled documents.

\begin{verbatim}
\DeclareOption{weird}{\CROP@size{12truecm}{34truecm}}
\end{verbatim}

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

\begin{verbatim}
\usepackage[frame,weird,center]{crop}
\end{verbatim}
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}

### 2.7 Defining your own marks

If you need a funny mode, you can easily define it with only a couple of macros.

The \texttt{\textbackslash cropdef} command defines the mode switch. It takes as arguments: the name of a macro providing the page info (optional; enclosed in brackets), four 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 \texttt{\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.

\begin{verbatim}
\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}{\funnymarkB}{funny}
\end{verbatim}

Now you can select your new mode by typing \texttt{\crop[funny]}. Each of the axis marks is a picture that you can easily replace by some custom definition. There’s no setup command like \texttt{\textbackslash cropdef}, though. The kernel provides two ‘hooks’ that can be used to add local extensions. These are macros that default to \texttt{\relax}. The first, \texttt{\textbackslash CROP@user@a}, is executed at every page, no matter if marks are shown or not, while the second, \texttt{\textbackslash CROP@user@b} is only executed at pages that contain crop marks. Local definitions and modifications are ideally put into a local configuration file:

### 2.8 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 can 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:

\begin{verbatim}
% define a new printing paper size
\DeclareOption{special}{\CROP@size{22truecm}{37truecm}}
\end{verbatim}
% 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[1]{\textsf{#1}})
\crop[font=textsf]
\endinput

3 How the package works

3.1 The kernel mechanism

\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 crop marks 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 Dependencies

3.2.1 latex.ltx

The package works with all \LaTeX standard classes (tested with \LaTeX 1997/12/01 and sporadically with later versions), it does not work with plain\TeX.

The crop package uses (and relies on) the internal \LaTeX tokens \hb@xt@,\nfilename@parse, \@classoptionslist, \@ifundefined, \@height, \@depth,\nfilename@base \@width, \z@, \@ne, \z@skip, \@p@, \c@page, \@namedef,\n\@nameuse, \@strip@pt, \@twodigits, \@count@, \@dimen@, \@for, \@empty,\n\@gobble and \@undefined, all of which are expected to keep their current mean-
ing in future \LaTeX releases. The crop package will, however, be supported at
least for some years, so you needn’t worry about it.

3.2.2 color.sty

crop’s color handling depends on the color package. The following internal
macros are used directly: \declaredcolor, \current@color, \set@color,\n\set@page@color (Tested with color.sty, version 1.0i as of 1999/02/16.)

3.2.3 graphics.sty

crop’s driver detection, as well as the options rotate mirror, and nographics de-
pend on the graphics package. The following internal macros are used directly:
\Gin@PS@raw, \Ginclude@graphics, \Gin@driver (Tested with graphics.sty,
version 1.0l as of 1999/02/16.)
4 The implementation

4.1 Preamble

\stockwidth \stockheight \CROP@index \CROP@font

Make sure that \stockwidth and \stockheight are \dimen registers that hold the physical paper size. They are initially set to the paper size, but will be changed by the size options. These registers are also used/provided by the memoir class and the hyperref package. The \CROP@font macro is by default empty and can be changed through the font option.

\CROP@stockcolor \CROP@pagecolor \CROP@needscolor

Try to load the color package. It is needed for the invert and the notext option and, of course, for the modified \pagecolor command. Changing the meaning of \current@color looks dangerous, but it is only done if the color package couldn’t be loaded, anyway.

\CROP@detdriver \CROP@Ginclude@graphics \CROP@ps

The options graphics and nographics depend on the graphics package, which, if configured appropriately, also tells us which output device is preferred on the
system. Show a warning, if the package couldn’t be loaded, because we have to use a less portable PS ‘driver’ then.

\let\CROP@detdriver@empty
\IfFileExists{graphics.sty}{%\RequirePackage{graphics}\let\CROP@Ginclude@graphics\ifx\Gin@driver@empty\else\filename@parse{\Gin@driver}\edef\CROP@detdriver{\filename@base}\fi\let\CROP@ps\Gin@PS@raw\let\CROP@ps\Gin@PS@raw\PackageWarning{crop}{I couldn’t find the ‘graphics’ package, so I’ll use my internal PostScript interface}\newcommand*{\CROP@ps}[1]{\special{ps: ##1}}\CROP@reqdriver

\CROP@reqdriver Define options that suggest ...
\let\CROP@reqdriver@empty
\DeclareOption{vtex}{\def\CROP@reqdriver{vtex}}\DeclareOption{pdftex}{\def\CROP@reqdriver{pdftex}}\DeclareOption{pdflatex}{\def\CROP@reqdriver{pdftex}}\DeclareOption{dvips}{\def\CROP@reqdriver{dvips}}\CROP@driver

\CROP@driver ... or enforce a graphics driver. Note the exclamation points!
\let\CROP@driver@empty
\DeclareOption{vtex!}{\def\CROP@driver{vtex}}\DeclareOption{pdftex!}{\def\CROP@driver{pdftex}}\DeclareOption{pdflatex!}{\def\CROP@driver{pdftex}}\DeclareOption{dvips!}{\def\CROP@driver{dvips}}\DeclareOption{nodriver}{\def\CROP@driver{none}}\DeclareOption{!}{\def\CROP@driver{none}}\CROP@evaldriver

\CROP@evaldriver If \CROP@driver wasn’t already set, decide \AtBeginDocument which graphics driver to use. A detected driver takes precedence over a ‘suggested’ one. Show a warning if the user’s choice is ignored.
\newcommand*{\CROP@evaldriver}{%\ifx\CROP@driver@empty\PackageInfo{crop}{requested driver: ‘\CROP@reqdriver’}\else\PackageWarning{crop}{requested driver: ‘\CROP@reqdriver’}\PackageWarning{crop}{detected driver: ‘\CROP@detdriver’}\fi\PackageInfo{crop}{detected driver: ‘\CROP@detdriver’}
\if\CROP@reqdriver\empty\else
  \if\CROP@reqdriver\empty\else
    \if\CROP@reqdriver\CROP@detdriver\else
      \PackageWarningNoLine{crop}{%
        You requested the \texttt{\CROP@reqdriver} driver
        but I think that \texttt{\CROP@detdriver} works better in the
        current context. You can force
        me to respect your decision by adding an exclamation point as in
        \texttt{[\CROP@reqdriver!]}%}
    \fi
  \fi
\fi
\fi
\let\CROP@driver\CROP@reqdriver
\else
  \let\CROP@driver\CROP@detdriver
\fi
\let\CROP@evaldriver\relax
\AtBeginDocument{\CROP@evaldriver}
\CROP@init@dvips
\CROP@init@pdftex
\CROP@init@vtex
\CROP@init@none
These macros prepare the \texttt{crop} package for one of the supported graphics drivers. They don’t do anything spectacular, they just select the proper \texttt{rotate} and \texttt{mirror} macro and hand the physical paper size over to the driver program.
\newcommand*{\CROP@init@dvips}{%
  \PackageInfo{crop}{using \texttt{dvips} graphics driver}%
  \AtBeginDvi{%
    \special{papersize=\the\stockwidth,\the\stockheight}%
  }%
}
\newcommand*{\CROP@init@pdftex}{%
  \PackageInfo{crop}{using \texttt{pdftex} graphics driver}%
  \pdfpagewidth\stockwidth
  \pdfpageheight\stockheight
  \let\CROP@reflect\CROP@genreflect
  \let\CROP@rotate\CROP@genrotate
}
\newcommand*{\CROP@init@vtex}{%
  \PackageInfo{crop}{using \texttt{vtex} graphics driver}%
  \mediawidth\stockwidth
  \mediaheight\stockheight
  \let\CROP@reflect\CROP@genreflect
  \let\CROP@rotate\CROP@genrotate
}
\newcommand*{\CROP@init@none}{%
  \PackageInfo{crop}{not using any graphics driver}%
}
4.3 Size options

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

\newcommand{\CROP@size}[2]{\stockwidth#1 \stockheight#2 }
\DeclareOption{landscape}{%
  \def{\CROP@size#1#2{\stockheight#1 \stockwidth#2 }}%
}
\DeclareOption{a0}{\CROP@size{841truemm}{1189truemm}}
\DeclareOption{a1}{\CROP@size{595truemm}{841truemm}}
\DeclareOption{a2}{\CROP@size{420truemm}{595truemm}}
\DeclareOption{a3}{\CROP@size{297truemm}{420truemm}}
\DeclareOption{a4}{\CROP@size{210truemm}{297truemm}}
\DeclareOption{a5}{\CROP@size{149truemm}{210truemm}}
\DeclareOption{a6}{\CROP@size{105truemm}{149truemm}}
\DeclareOption{b0}{\CROP@size{1000truemm}{1414truemm}}
\DeclareOption{b1}{\CROP@size{707truemm}{1000truemm}}
\DeclareOption{b2}{\CROP@size{500truemm}{707truemm}}
\DeclareOption{b3}{\CROP@size{353truemm}{500truemm}}
\DeclareOption{b4}{\CROP@size{250truemm}{353truemm}}
\DeclareOption{b5}{\CROP@size{176truemm}{250truemm}}
\DeclareOption{b6}{\CROP@size{125truemm}{176truemm}}
\DeclareOption{letter}{\CROP@size{8.5truein}{11truein}}
\DeclareOption{legal}{\CROP@size{8.5truein}{14truein}}
\DeclareOption{executive}{\CROP@size{7.25truein}{10.5truein}}
\newcommand{\CROP@opt@width}{\stockwidth\CROP@@}
\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 sheets.

\newcommand{\CROP@center}{\AtBeginDocument{\CROP@center}}
\DeclareOption{center}{\AtBeginDocument{\CROP@center}}
\newcommand{\CROP@opt@width}{\stockwidth\CROP@@}
\newcommand{\CROP@opt@height}{\stockheight\CROP@@}

4.4 Runtime options handling

Pass every unknown option to the macro \CROP@execopt.

\newcommand{\crop}[1][cam,noaxes]{%
  \for{\CROP@opt@:=1}{\do{\CROP@execopt\CROP@opt@}}%
}
Every execution of this macro with an argument \( n \) leads to the execution of a macro \( \text{\texttt{\textbackslash opt@}} n \) or a warning if no such exists. Optional arguments (separated by an equal sign) are cut off and stored in \( \text{\texttt{\textbackslash opt@}} \). The macro tolerates even arguments for options that are not prepared to handle arguments (e.g. \texttt{cross=garbage}), or more than one argument (e.g. \texttt{mount2=1=garbage}). This makes the design simpler and doesn’t hurt.

\begin{verbatim}
\newcommand*{\execopt}[1]{
def\opt@=#1\@nil{
def\opt{#1}
def\opt@@{#2}}\
\expandafter\opt@#1==\@nil\PackageError{crop}{Requested option ‘#1’ not provided}{Note that the ‘center’ options are obsolete. You have to request `MessageBreak e.g. `[a4,center] instead of [a4center].}
}
\end{verbatim}

The \texttt{\texttt{\textbackslash cropdef}} macro defines a mode switch (see section 2.7). It supports only corner marks and the page info, but not the axis marks, mainly for hysterical raisins.

\begin{verbatim}
\newcommand*{\cropdef}[6][\texttt{\texttt{\textbackslash info}}]{\@namedef{\texttt{\texttt{\textbackslash opt@}}#6}{\def\opt{#1}\let\ulc#2\let\urc#3\let\llc#4\let\lrc#5\let\marks=CROP@marks}}
\end{verbatim}

### 4.5 Axes and page info

The standard definitions for the \texttt{axes} option.

\begin{verbatim}
\newcommand*{\laxis}{\begin{picture}(0,0)\unitlength\p@\thinlines\put(-2,0){\line(-1,0){11}}\end{picture}}
\newcommand*{\raxis}{\begin{picture}(0,0)\unitlength\p@\thinlines\put(2,0){\line(1,0){11}}\end{picture}}
\newcommand*{\upaxis}{\begin{picture}(0,0)\unitlength\p@\thinlines\put(-2,0){\line(-1,0){11}}\put(2,0){\line(1,0){11}}\end{picture}}
\end{verbatim}
This macro prints the jobname, the current date and time, the page number and an index number at the top of the logical page.
4.6 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 \textbf{The cam mode corner mark for the upper left corner.}

\begin{verbatim}
\newcommand*{\CROP@@ulc}{%
  \begin{picture}(0,0)
  \unitlength\p@	hinlines
  \put(-30,0){\circle{10}}
  \put(-30,-5){\line(0,1){10}}
  \put(-35,0){\line(1,0){30}}
  \put(0,30){\circle{10}}
  \put(-5,30){\line(1,0){10}}
  \put(0,35){\line(0,-1){30}}
  \end{picture}%
}
\end{verbatim}

\CROP@@urc \textbf{The cam mode corner mark for the upper right corner.}

\begin{verbatim}
\newcommand*{\CROP@@urc}{%
  \begin{picture}(0,0)
  \unitlength\p@	hinlines
  \put(30,0){\circle{10}}
  \put(30,-5){\line(0,1){10}}
  \put(35,0){\line(-1,0){30}}
  \put(0,30){\circle{10}}
  \put(-5,30){\line(1,0){10}}
  \put(0,35){\line(0,-1){30}}
  \end{picture}%
}
\end{verbatim}

\CROP@@llc \textbf{The cam mode corner mark for the lower left corner.}

\begin{verbatim}
\newcommand*{\CROP@@llc}{%
  \begin{picture}(0,0)
  \unitlength\p@	hinlines
  \put(-30,0){\circle{10}}
  \put(-30,-5){\line(0,1){10}}
  \put(-35,0){\line(1,0){30}}
  \put(0,-30){\circle{10}}
  \put(-5,-30){\line(1,0){10}}
  \put(0,-35){\line(0,1){30}}
  \end{picture}%
}
\end{verbatim}

\CROP@@lrc \textbf{The cam mode corner mark for the lower right corner.}

\begin{verbatim}
\newcommand*{\CROP@@lrc}{%
  \begin{picture}(0,0)
  \unitlength\p@	hinlines
  \put(30,0){\circle{10}}
  \put(30,-5){\line(0,1){10}}
  \put(35,0){\line(-1,0){30}}
  \put(0,-30){\circle{10}}
  \put(-5,-30){\line(1,0){10}}
  \put(0,-35){\line(0,1){30}}
  \end{picture}%
}
\end{verbatim}
Define the \texttt{cam} mode switch with four different marks.

\begin{quote}
\verb|\cropdef\CROP@@ulc\CROP@@urc\CROP@@llc\CROP@@lrc{cam}|
\end{quote}

This macro provides a 4 cm wide cross.

\begin{quote}
\verb|\newcommand*\CROP@@cross{|
\verb|\begin{picture}(0,0)\|
\verb|\unitlength1cm\thinlines\|
\verb|\put(-2,0){\line(1,0){4}}\|
\verb|\put(0,-2){\line(0,1){4}}\|
\verb|\end{picture}|
\end{quote}

Define the \texttt{cross} mode switch with four times the same mark.

\begin{quote}
\verb|\cropdef\CROP@@cross\CROP@@cross\CROP@@cross\CROP@@cross{cross}|
\end{quote}

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

\begin{quote}
\verb|\newcommand*\CROP@@frame{|
\verb|\begin{picture}(0,0)\|
\verb|\unitlength\p@\thinlines\|
\verb|\put(0,0){\line(1,0){\strip@pt\paperwidth}}\|
\verb|\put(0,0){\line(0,-1){\strip@pt\paperheight}}\|
\verb|\put(\strip@pt\paperwidth,0){\line(0,-1){\strip@pt\paperheight}}\|
\verb|\put(0,-\strip@pt\paperheight){\line(1,0){\strip@pt\paperwidth}}\|
\verb|\end{picture}|
\end{quote}

Define the \texttt{frame} mode switch with only one mark. The other corners may \texttt{relax}.

\begin{quote}
\verb|\cropdef\CROP@@frame\relax\relax\relax\relax{frame}|
\end{quote}

\section*{4.7 The kernel}

These macros redefine the \TeX{} primitive \texttt{\shipout} to insert the contents of the macro \texttt{\CROP@shiplist} on top of the box which contains the page contents ready for output, after which the original \texttt{\shipout} command is executed.

\begin{quote}
\verb|\let\CROP@shipout\shipout\|
\verb|\renewcommand*\shipout{|
\verb|\afterassignment\CROP@ship\|
\verb|\setbox\@cclv=\%\|
\verb|}\|
\verb|\ifvoid\@cclv\expandafter\aftergroup\|
\verb|\fi\|
\verb|\newcommand*\CROP@ship{|
\verb|\ifvoid\@cclv|}
This macro adds a page manipulation command to the shiplist, which gets every ready page as argument.

\newcommand*{\CROP@shipadd}[1]{
  \begin{group}
    \toks@\expandafter{\expandafter#1\expandafter{\CROP@shiplist}}
    \xdef{\CROP@shiplist}{\the{\toks@}}
  \end{group}
}

\CROP@kernel\CROP@marks
\CROP@markcolor\CROP@user@a\CROP@user@b\CROP@opt@horigin\CROP@opt@vorigin
\CROP@kernel
  \vbox to\z@{
    \vskip{\CROP@vorigin}
    \hb@xt@\z@{
      \hskip{\CROP@horigin}
      \vbox to\paperheight{
        \let\protect\relax
        \hssize\paperwidth
        \CROP@hook
        \CROP@user@a
        \CROP@drawstockcolor
        \CROP@drawpagecolor
        \CROP@opt@marks
      }
    }
    \hss
  }
\newcommand*{\CROP@marks}{
  \CROP@setmarkcolor
  \CROP@user@b
  \CROP@ulc\null\hfill\CROP@@@info\CROP@upedge\hfill\null\CROP@lrc
  \vfill
  \CROP@llc\null\hfill\null\CROP@loedge\hfill\null\CROP@lrc
}

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

\newcommand*{\CROP@shipadd}[1]{
  \begin{group}
    \toks@\expandafter{\expandafter#1\expandafter{\CROP@shiplist}}
    \xdef{\CROP@shiplist}{\the{\toks@}}
  \end{group}
}

\CROP@kernel\CROP@marks
\CROP@markcolor\CROP@user@a\CROP@user@b\CROP@opt@horigin\CROP@opt@vorigin
\CROP@kernel
  \vbox to\z@{
    \vskip{\CROP@vorigin}
    \hb@xt@\z@{
      \hskip{\CROP@horigin}
      \vbox to\paperheight{
        \let\protect\relax
        \hssize\paperwidth
        \CROP@hook
        \CROP@user@a
        \CROP@drawstockcolor
        \CROP@drawpagecolor
        \CROP@opt@marks
      }
    }
    \hss
  }
\newcommand*{\CROP@marks}{
  \CROP@setmarkcolor
  \CROP@user@b
  \CROP@ulc\null\hfill\CROP@@@info\CROP@upedge\hfill\null\CROP@lrc
  \vfill
  \CROP@llc\null\hfill\null\CROP@loedge\hfill\null\CROP@lrc
}
These macros start and stop the output of crop marks.

Enable and disable the output of axis marks and page info.

4.8 Mounting

Since \newcommand doesn’t allow macro names to contain non-letters, we need a construction with \csname, \endcsname, and \expandafter. \namedef would have worked, too, but it would not have made a check for redefinitions.
4.9 Page manipulation

The mirror and rotate options add a macro to the shiplist, which then gets every output page and embeds it in a POSTSCRIPT environment (dvips) or lets the graphics package reflect or rotate it (pdftex). We could also use the generic operations \CROP@genreflect and \CROP@genrotate for the dvips mode. They would produce correct PS documents, the intermediate DVI document, however, would be unreadable.

\DeclareOption{mirror}{% 
  \AtBeginDocument{\CROP@shipadd\CROP@reflect} }
\newcommand*\CROP@reflect[1]{% 
  \vbox to\z@{\vskip\CROP@vorigin \hb@xt@\z@{}{\hskip\CROP@horigin \CROP@ps{gsave currentpoint} \kern\paperwidth \CROP@ps{currentpoint}\hss}}\vss \CROP@ps{translate -1 1 scale neg exch neg exch translate}#1\CROP@ps{grestore} %}
\newcommand*\CROP@genreflect[1]{% \leavevmode \dimen0\CROP@horigin \kern2\dimen0 \reflectbox{\hb@xt@\paperwidth{\vbox to\paperheight{#1\vbox{\hss}}}}\hs}}%
\DeclareOption{rotate}{% 
  \leavevmode 
  \dimen0\CROP@horigin 
  \kern2\dimen0 
  \reflectbox{% 
    \hsxt\paperwidth{% 
      \vbox to\paperheight{% 
        \vbox{\hs}} 
        \hs}}}%
  }%}
\DeclareOption{rotate}{%
4.10 Color handling

These macros care for the color of crop marks and of the logical and the physical page. The overlap value is the amount that the logical page is drawn over the page boundaries on each side. This is necessary to get good results on imprecise cutting machines.
The invert option simply switches to black background and white text, after which it disables all color switching commands. The notext option does the same with white text on white background. The \@gobble on the last line keeps notext from switching to white background and breaking a prior invert.
4.11 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.

\CROP@opt@graphics

4.12 Final settings

\CROP@horigin \CROP@vorigin

Switch off marks and axes, set one page per sheet, load the local configuration file, and process the requested options. Finally: Exit.

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 \AtBeginDocument, because there are possibly commands from the center option in the queue that have to be processed first.

\CROP@horigin{-1truein}
\CROP@vorigin{-1truein}
\crop[cam,off,noaxes,info,mount1]
\InputIfFileExists{crop.cfg}{%
\PackageInfo{crop}{Local config file crop.cfg used}%
}}
4.13 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@hook is defined first to set \count@ to the page number increased by this offset: \( p = \text{pagenumber} + \text{offset} \).

Now bits 0 and 1 are checked via \ifodd to get \( p \mod 4 \), 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.

\begin{verbatim}
\ifodd\count@ \%% if (p % 4 == 1 || p % 4 == 3) \let\CROP@ulc\relax \let\CROP@llc\relax \let\CROP@ledge\relax \divide\count@2 \ifodd\count@ \%% if (p % 4 == 3) \let\CROP@urc\relax \let\CROP@info\relax \let\CROP@upedge\relax \else \%% if (p % 4 == 1) \let\CROP@lrc\relax \let\CROP@loedge\relax \fi \else \%% if (p % 4 == 0 || p % 4 == 2) \let\CROP@urc\relax \let\CROP@lrc\relax \let\CROP@redge\relax \divide\count@2 \ifodd\count@ \%% if (p % 4 == 2) \let\CROP@llc\relax \end{verbatim}