Printing digital photographs with LaTeX

Frédéric Boulanger
Supélec – Service Informatique
Plateau de Moulon, 3 rue Joliot-Curie
F-91192 Gif-sur-Yvette cedex
Frederic.Boulanger@supelec.fr
http://wwwsi.supelec.fr/fb/

September 28, 2006

Résumé

Des appareils photographiques numériques et des imprimantes à jets d'encre de prix abordables permettent aujourd'hui d'obtenir des tirages papier d'une qualité équivalente, sinon meilleure, à celle obtenue avec un appareil argentique compact et un tirage standard. Toutefois, l'impression en quantité de photographies en nombre de copies variable sur du papier de taille standard devient rapidement malaisée. Il existe des programmes permettant de placer des photos dans un album pour l'imprimer ensuite, mais ils exigent généralement un placement manuel de chaque photographie. La classe de documents LaTeX photoprint est une tentative d'automatisation de l'impression de photographies sur du papier de taille standard. Elle s'appuie sur une liste de photographies associées au nombre de copies désirées et sur un découpage de chaque feuille en « emplacements d'impression ». Elle gère la rotation, la mise à l'échelle et le rognage des photographies pour les faire tenir au mieux dans les emplacements. Elle permet aussi d'obtenir un index des photographies qui facilite le choix des retirages, peut imprimer des marques repères pour le découpage, et afficher la date en surimpression.

Abstract

Affordable digital cameras and photo inkjet printers make it possible to obtain digital prints with a quality which is generally as good as, and often better than with an inexpensive conventional camera and standard processing. However, printing many photographs in various number of copies on standard-size paper may be cumbersome. There are programs to build albums and print them, but they generally force you to lay out photographs by hand on each page. The photoprint LaTeX document class is an attempt to automatize the printing of digital photographs on standard-size paper. It uses a list of photographs tagged with the number of copies to print, divides the paper into "print slots", and takes care of rotating, cropping or scaling the photographs to make them fit into the slots. It can also build an index of the photographs to help people choose the prints they want, add crop marks to the printed pages to help cut the prints or overprint the date on the prints.

1 Introduction

Printing photographs from a family wish list can make you busy during a whole week-end: you must find the photographs to print, lay out them on pages, taking the total number of copies of each into account, rotate them if they do not match the page orientation, resize and crop them so they match the paper size, wait for the printer, and finally cut the pages into prints. Of course, you can use special photo-paper with pre-cut margins, but it is more expensive than standard-size paper, yields less clean edges, works only for a given brand of printers because its margins are adapted to the print area of the printer, and it limits you to one print size.

Using standard-size paper (for instance A4 paper) gives you more freedom since you can use it for prints from 10×14 cm up to 20×28 cm. You can even produce smaller prints, for instance to make stamps printed on self-adhesive paper. But this freedom comes at the cost of the layout of the photographs on the paper.

Regular digital photographs have a $\frac{4}{3}$ aspect ratio, while A4 paper has a $\sqrt{2}$ aspect ratio. So, to use the largest part of the expensive sheets, the photos must be cropped. Some photos are in portrait orientation (their longer side is vertical), while others are in

landscape orientation (their longer side is horizontal), so the photos which don't have the same orientation as the paper must be rotated.

The photoprint LaTeX document class divides pages into "print slots" and lays out photographs on these pages. It reads the name of the photographs and the number of copies of each from a file. This file can be written by hand, created by a shell script, or generated by a helper Java application which provides a contactsheet-like view of the photographs from which you can select pictures and choose the number of copies.

Since most digital cameras yield JPEG files, it is much easier to use photoprint with pdflatex. However, nothing in photoprint is hardcoded for pdflatex, so if you provide bounding box files for your JPEG pictures so that graphicx can determine their size, you can use photoprint with ordinary latex.

2 Basic setup

Since photoprint is a regular LaTeX document class, you just write:

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

to use it with A4 paper in landscape orientation. Then, you have to specify the size of the printable area of your printer. Since this does not depend on the orientation of the paper, photoprint uses the top, left, bottom and right margins from the edges of the paper in portrait orientation to describe the print area. The command used to set the print area is:

`\photopaper` `\photopaper{`⟨*top*⟩`}{`⟨*left*⟩`}{`⟨*bottom*⟩`}{`⟨*right*⟩`}`

For instance:

```
\photopaper{0.5cm}{0.5cm}{1cm}{0.5cm}
```

says that your printer can print as close as 0.5 cm from the top edge, 0.5 cm from the left edge, 1 cm from the bottom edge, and 0.5cm from the right edge of a sheet of paper. These are the default setting if you don't use `\photopaper`.

You can consider that the `\photopaper` command takes a physical description of the print area in which top is the side which comes out of the printer first, bottom is the side which comes out last, and left and right are defined relatively to top and bottom, without consideration of the printing orientation.

The bottom margin is generally larger than the other margins because the printer must still be able to hold the paper between its rolls while printing the bottom of the page. If you print in landscape orientation, the bottom margin will be on the left of the printed page.

Next, you have to decide how you want to divide a sheet of paper. For this, you give the number of print slots per line (so it is a number of columns), and the number of print slots per column (the number of lines). You set the number of photographs on a line with the:

`\photosperline` `\photosperline{`⟨*number*⟩`}`

command, and the number of photographs in a column with the:

`\photospercol` `\photospercol{`⟨*number*⟩`}`

command. For instance:

```
\photosperline{2}
\photospercol{2}
```

sets four print slots. This is the default setting which yields about 10×14 cm prints on A4 paper.

To print a contact sheet, in landscape orientation, you could use:

```
\photosperline{5}
\photospercol{4}
```

which makes for 4 rows of 5 pictures on each page. However, since printing contact sheets is a quite common operation, photoprint has a `\contactsheet` command for this which does more than printing the pictures.

Last, in the body of your document, you will lay out the pictures, using as many pages as necessary, with the command:

`\printphotos` `\printphotos[`⟨*filename*⟩`]`

⟨*filename*⟩ is the name of the file which contains the list and number of copies of the pictures to print. It defaults to photoprint.job. To use the list photosfordad.job, write:

```
\printphotos[photosfordad.job]
```

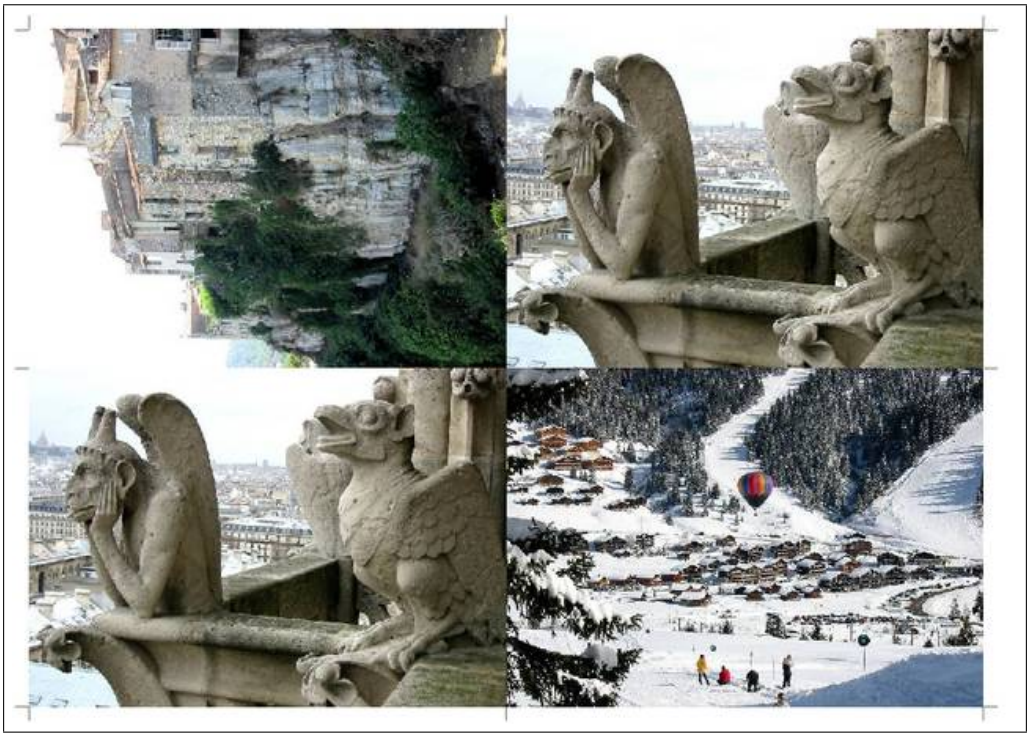


Figure 1: Result of `pdflatex photos.tex`

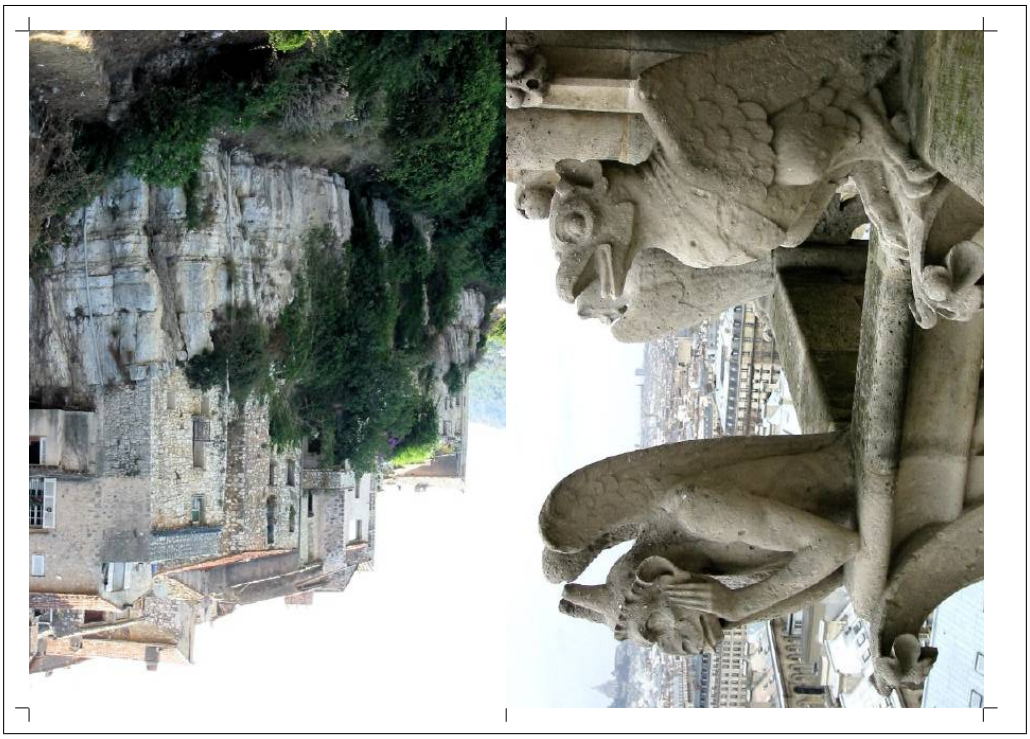


Figure 2: Result (first page only) of `pdflatex largephotos.tex`

The format of the photoprint.job file is very simple: each line contains the name of a picture file, optionally followed by a * and the number of copies to print. For instance, the following asks for one copy of 001-23-IMG_0375.jpg, two copies of 004-20-IMG_0949.jpg, one copy of 006-22-IMG_1126.jpg, and no copy of the other pictures:

```
001-23-IMG_0375.jpg*1
002-28-IMG_0425.jpg*0
003-27-IMG_0707.jpg*0
004-20-IMG_0949.jpg*2
005-20-IMG_1091.jpg*0
006-22-IMG_1126.jpg*1
```

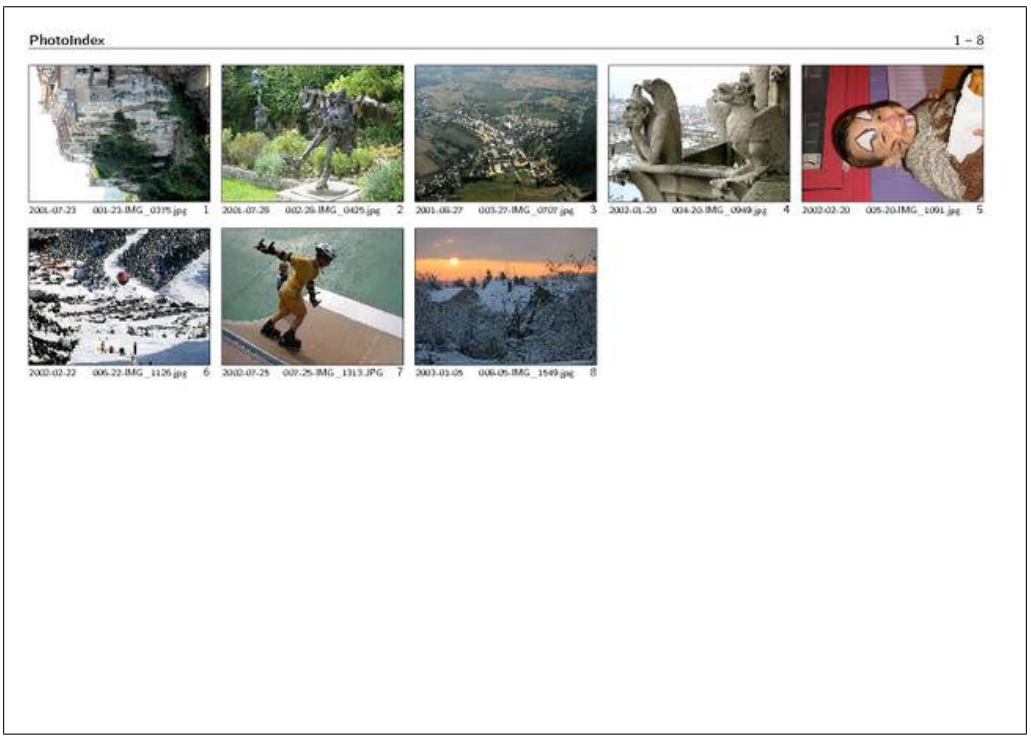


Figure 3: Result of `pdflatex contactsheet.tex`



Figure 4: Effect of `whitemargins` option

```
007-25-IMG_1313.JPG*0
008-05-IMG_1549.jpg*0
```

`\contactsheet` If you want contact sheets, use `\contactsheet` instead of `\printphotos`. `\contactsheet` will ignore any `\photosperline` and `\photospercol` settings and print four rows of 5 pictures per page, with a header, and the name and indice of each photo printed below it. You may obtain strange results if you don't use landscape orientation when printing contact sheets. When printing contact sheets, photoprint ignores the number of copies requested for each photo, and puts each photo once on the contact sheet.

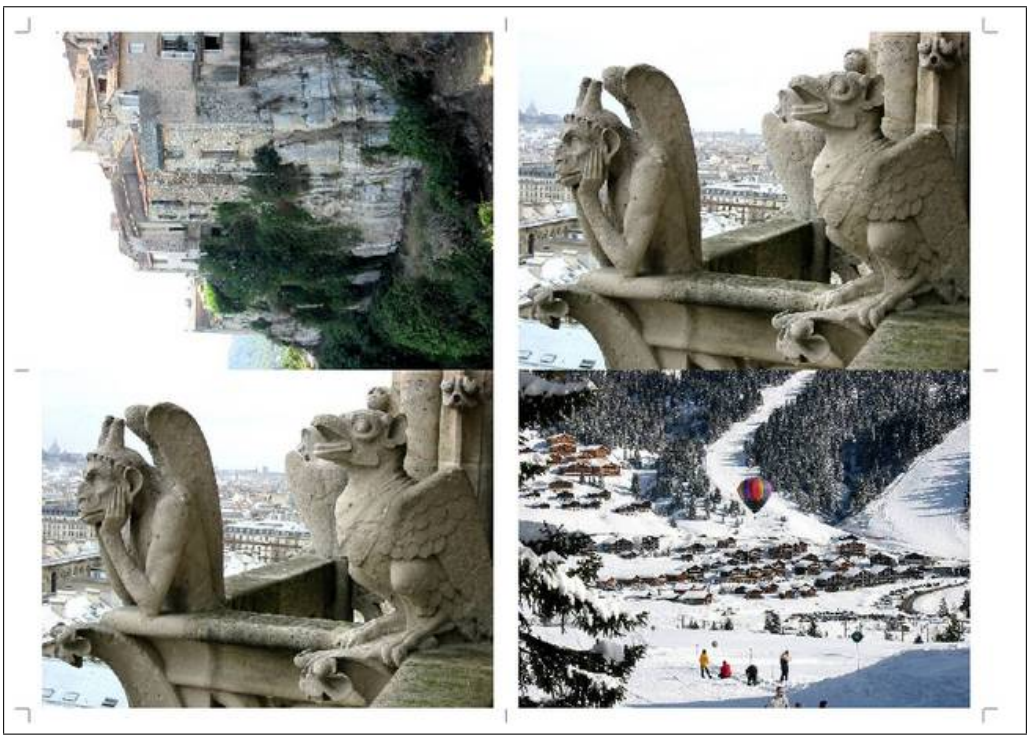


Figure 5: Effect of the `fullpictures` option



Figure 6: Effect (exaggerated) of the `showdate` option

3 Convenience files

Some settings are so common that two files can be used to print photos without writing any LaTeX commands. The file photos.tex prints four pictures by page in landscape orientation:

```
\documentclass[%
  a4paper,
  landscape,
  cropmarks
]{photoprint}
\photosperline{2}
\photospercol{2}
\begin{document}
```

```
  \printphotos%[name-of-photo-list]
\end{document}
```

With the contents of the photoprint.job file as shown above, running

```
pdflatex photos.tex
```

yields the result shown on figure 1.

The file largephotos.tex prints two pictures by page and yields 14×20 cm prints on A4 paper:

```
\documentclass[%
  a4paper,
  landscape,
  cropmarks
]{photoprint}
\photosperline{2}
\photospercol{1}
\begin{document}
  \printphotos%[name-of-photo-list]
\end{document}
```

With the contents of the photoprint.job file as shown above, running

```
pdflatex largephotos.tex
```

yields the result shown on figure 2.

The file contactsheet.tex prints contact sheets for the pictures listed in photoprint.job. It ignores the number of copies and makes each photo appear once on the contact sheet.

With the contents of the photoprint.job file as shown above, running

```
pdflatex contactsheet.tex
```

yields the result shown on figure 3.

For Unix systems (Linux, Mac OS X and others), photoprint is distributed with two shell scripts. The mkphoto script builds a photoprint.job file containing the list of all JPEG files in the current directory, with a number of copies set to 0. This script considers files as JPEG if their extension is `.jpg` or `.jpeg`, in either lower or upper case. The mkcs script builds a contact sheet from the photoprint.job file in the current directory. It uses the name of the current directory as the title of the contact sheet.

When installing photoprint, since docstrip adds a dot at the end of the name of both scripts when they are extracted, you should rename these scripts and make them executable with the `chmod` Unix command. For instance `chmod a+x mkcs` will make mkcs executable by any user.

4 Document class options

On the examples of the previous section, you can see black lines in the margins of the pages.

`cropmarks` This is the effect of the `cropmarks` option. These marks help adjust the position of the page on the paper cutter. The default is not to print crop marks.

`whitemargins` If you want white margins around your prints once they are cut, use the `whitemargins` option. The length of the smallest margin of the printable area is used as the white margin, making for an even white border around the prints when they are cut.

The effect of the `whitemargins` option is shown in Figure 4.

`fullpictures` Some photographs may contain important details even at their edges, and cropping them to adapt their aspect ratio may lead to bad results with important details removed. The `fullpictures` option prevents photoprint from cropping pictures. This makes less optimal use of the paper since the aspect ratio of the pictures can no longer be adapted to the print slots, as shown on figure 5, where the white band down the middle is wasted space.

`landscape` The `landscape` option makes photoprint use the paper in landscape orientation (largest edge is horizontal). It should always be used for contactsheets. When printing photographs, the `landscape` option does not change the aspect ratio of the prints. It only changes the way they are laid on the pages.

`showdate` The `showdate` option prints the date in the lower right corner of the pictures; it may be useful to know when a picture was taken. The date is read from a file which should be named

!Name_Map.can or !Name_Map.mav and contains information about the photographs. The name and format of this file are set for historical reasons from the tools I use to manage my digital photographs. Each line of the file describes a photograph with four items delimited by braces:

1. the number of seconds since january 1 1904 (the epoch on the Macintosh). This item is not used by photoprint and may be left empty (but is must be present);
2. the file name of the photograph;
3. the file name of the unmodified picture (if it was kept as a backup). This item is not used by photoprint and may be left empty;
4. the date and time at which the photograph was taken, in ISO format:
   `YYYY-MM-DDThh:mm:ssZ`
   where `YYYY` is the year, `MM` the month (1 for January, 12 for December), `DD` is the day of month, `T` the separator between date and time in the ISO format, `hh` the hour (0–23), `mm` the minutes (0–59), `ss` the seconds (0–59), and `Z` is the time zone. The letter `Z` indicates UTC time. It may be replaced by the signed offset of the time with regard to Universal Time (for instance `+01:00` is the offset of the French local time in winter). Everything after the `T` is ignored by photoprint.

For instance, the file used to store information about the example pictures has the following contents:

```
{3078749666} {001-23-IMG_0375.jpg} {} {2001-07-23T16:14:26+01:00}
{3079183448} {002-28-IMG_0425.jpg} {} {2001-07-28T16:44:08+01:00}
{3081773514} {003-27-IMG_0707.jpg} {} {2001-08-27T16:11:54+01:00}
{3094393350} {004-20-IMG_0949.jpg} {} {2002-01-20T17:42:30+01:00}
{3097077318} {005-20-IMG_1091.jpg} {} {2002-02-20T19:15:18+01:00}
{3097226436} {006-22-IMG_1126.jpg} {} {2002-02-22T12:40:36+01:00}
{3110471962} {007-25-IMG_1313.JPG} {} {2002-07-25T19:59:22+01:00}
{3124602926} {008-05-IMG_1549.jpg} {} {2003-01-05T09:15:26+01:00}
```

The effect of the `showdate` option is shown on figure 6 where unusal settings have been chosen to make the date very visible (`\datecolor{red}` and `\datestyle{\Huge\sffamily\bfseries}`).

Other options (for instance `a4paper`) are passed to the article document class.

5 Finer tuning

Several commands allow changing the default behavior of photoprint:

`\datecolor` `\datecolor{`⟨***color***⟩`}` sets the color used to print the date on pictures when the `showdate` option is active. The default is `black`.

`\datestyle` `\datestyle{`⟨***style***⟩`}` sets the text style used for the date on pictures when the `showdate` option is active. *style* should be a series of text style declarations. The default is `\sffamily\small`.

`\infostyle` `\infostyle{`⟨***style***⟩`}` sets the text style which is used to print information about the pictures on contact sheets (date and name). *style* should be a series of text style declarations. The default is `\sffamily\footnotesize`.

`\cstitle` `\cstitle` If this command is defined, its value will be used as the title of the contact sheet by the `\contactsheet` command. For instance:

```
\newcommand{\cstitle}{Summer 2002}
\contactsheet
```

will produce a contact sheet with title “Summer 2002”.

`\photohmargin` `\photohmargin{`⟨***length***⟩`}` sets the amount of horizontal white space left around pictures. Consecutive pictures will be separated horizontally by twice this length. The default value is 0 pt (or the smallest margin of the print area when the `whitemargins` option is active).

`\photovmargin` `\photovmargin{⟨length⟩}` sets the amount of vertical white space left around pictures. Consecutive pictures will be separated vertically by twice this length. The default value is 0 pt (or the smallest margin of the print area when the `whitemargins` option is active).

`\photomargins` `\photomargins{⟨length⟩}` sets the amount of white space left around pictures, both vertically and horizontally. Consecutive pictures will be separated by twice this length. The default value is 0 pt (or the smallest margin of the print area when the `whitemargins` option is active).

6 Requirements

The photoprint document class relies on the standard LaTeX document class article, on the standard LaTeX packages calc, ifthen, graphicx, color and fontenc. It also needs the geometry package.

Althought photoprint does not make any assumption about the TeX engine used, and uses the graphicx package to include pictures, it is much easier to use with pdftex since digital photographs are generally in JPEG format.

7 The tasks of photoprint

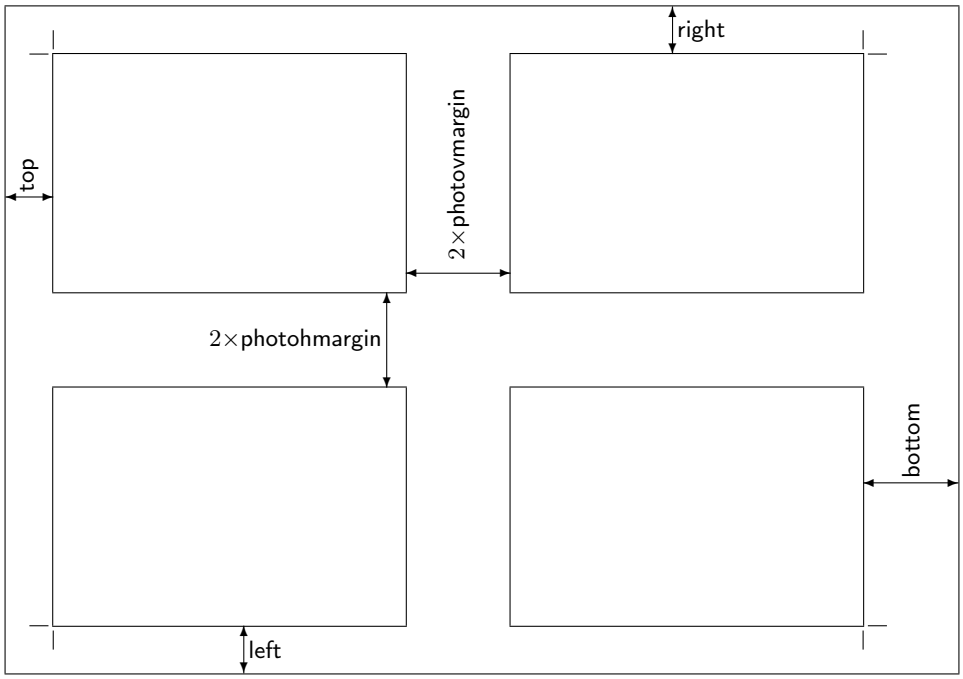


Figure 7: Dividing the print area

The class photoprint performs several tasks to prepare pages of prints from the list of photographs and the selected layout:

1. from the dimensions of the print area, the orientation of the paper, and the number of photographs to print per page, photoprint computes the size and position of the "print slots" (the areas where pictures will be placed).

2. from the list of photographs and the number of copies to print, photoprint builds a list of pictures to lay out.

3. for each picture in this list, photoprint compares the orientation of the picture and the orientation of the current print slot. If they differ, the picture is rotated by 90 °.

4. the picture is then scaled to fit into the print slot.

5. when the last print slot on a page has been used, a new page is started.



Print slot is too narrow



Print slot is too shallow

Figure 8: Fitting a picture into a print slot

7.1 Determining print slots

photoprint considers only the print area of a page, and divides it according to the required number of photos per line and per column. The horizontal and vertical margins around prints are also taken into account to determine the size of the print slots. Figure 7 shows how print slots are determined in landscape orientation, with two photos per line and per column.

7.2 Reading the list of photos

The default name for the file which contains the list of photographs is photoprint.job. In this file, each line should be formated as:

$$\langle filename \rangle * \langle num \rangle$$

or as:

$$\langle filename \rangle$$

where $\langle filename \rangle$ is the name of a picture file, and $\langle num \rangle$ is the requested number of copies for this photograph. If the "$*\langle num \rangle$" part is ommited, a value of one is assumed.

photoprint stops reading the file as soon as it reads an empty string, so the file should not contain empty lines excepted at its end. There is no special character to mark a line as a comment.

7.3 Putting the photos in the print slots

The print slots are used starting from the upper left one, and ending with the lower right one, from top to bottom and from left to right. Each picture is put in the first unused slot on the page.

For each picture, photoprint compares the orientation (landscape or portrait) of the print slot and the orientation of the picture. If they differ, the picture is rotated by 90 ° counter-clockwise.

Then, photoprint computes the width the picture would have if it were scaled by a factor s_h so that its height would be exactly the height of the print slot. This width is called the *scaled width* of the picture. photoprint also computes the height the picture would have if it was scaled by a factor s_w so that its width were exactly the width of the print slot. This height is called the *scaled height* of the picture.

If the scaled width of the picture is larger than the width of the print slot, photoprint scales the picture by s_h and trims its left and right edges by half the difference between the scaled width and the width of the slot.

If the scaled height of the picture is larger than the height of the print slot, photoprint scales the picture by s_w and trims its top and bottom edges by half the difference between the scaled height and the height of the slot.

In both cases, the picture fills the print slot, but it is cropped, as shown on figure 8.

However, when the `fullpictures` option is on, photoprint uses the smallest of s_h and s_w so that the picture fits entirely inside the print slot, between equal white bands to fill the slot.

7.4 Contact sheets

Contact sheets are a special case in which photoprint reserves space at the top of the page to display the title of the contact sheet and the picture range (there are 20 pictures on each page of a contact sheet). It also reserves space between the rows of pictures to display their names, indices and date if available.

8 Related software

Photoprint started as a Java application, but I never managed to make it print my picture the right way. This is why I started writing the photoprint document class (everybody knows that it is much easier to develop in LaTeX than in Java...)

However, the Java Photoprint application still exists and allows to choose pictures to print and to set the number of copies of each. So I added a menu item to save the current selection of pictures to a file whose default name is ... photoprint.job.

This Java application has never been released, but I plan to make it available as a helper to photoprint. It may even launch pdftex and open the result in Acrobat so that people who don't like command-line tools feel more at ease. Keep an eye on

`http://wwwsi.supelec.fr/fb/Development.html`

for a release.

9 The implementation

photoprint works only in LaTeX2e, and is a document class. We load the ifthen package early since it is used to process options.

```
⟨*class⟩
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesClass{photoprint}
              [2004/02/22 v1.2]

\RequirePackage{ifthen}
```

We need some boolean values to remember which options are active.

```
\newboolean{photo@cropmarks}
\newboolean{photo@whitemargins}
\newboolean{photo@landscapepaper}
\newboolean{photo@fullpictures}
\newboolean{photo@showdate}

```

We declare the supported options, and pass other options to the article class. The `landscape` option is also passed to the geometry package.

```
\DeclareOption{cropmarks}{\setboolean{photo@cropmarks}{true}}
\DeclareOption{whitemargins}{\setboolean{photo@whitemargins}{true}}
\DeclareOption{fullpictures}{\setboolean{photo@fullpictures}{true}}
\DeclareOption{showdate}{\setboolean{photo@showdate}{true}}
\DeclareOption{landscape}{%
  \setboolean{photo@landscapepaper}{true}%
  \PassOptionsToClass{\CurrentOption}{article}%
  \PassOptionsToPackage{\CurrentOption}{geometry}%
}
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
\ProcessOptions\relax

```

Then, we load the article class and all the packages we need. calc is used for easier computations, geometry is used to set the margins, graphicx is used to include the photographs, and color is used to set the color of the date when the `showdate` option is active. We use the T1 font encoding to avoid issues with special characters in file names. For instance, if a file name (read from the list of photos) contains an underscore, it will be displayed correctly on the contact sheet only in T1 encoding. If we use OT1, we get something which looks like a degree symbol instead.

```
\LoadClass{article}
\RequirePackage{calc}
```

```
%\RequirePackage{geometry}
\RequirePackage{graphicx}
\RequirePackage{color}
\RequirePackage[T1]{fontenc}
```

We tell graphicx that `.JPG` files should be handled the same as `.jpg` files.

```
\DeclareGraphicsRule{.JPG}{jpg}{*}{}
```

We set `\parindent` to 0 to avoid unwanted space before the photos, and we choose an empty page style.

```
\setlength{\parindent}{0pt}
\pagestyle{empty}

% \end{macrocode}
%
% The \cs{photo@dir} macro contains the path to the directory which
% contains the pictures. When its is empty, the local directory is
% used (more precisely, the pictures are searched according to the
% use of \cs{graphicspath} macro of the graphics package).
%
% \begin{macro}{\photodir}
% The \cs{photodir} macro sets the value of \cs{photo@dir}.
%
% \begin{macrocode}
\newcommand*{\photo@dir}{}

\newcommand*{\photodir}[1]{%
  \renewcommand*{\photo@dir}{#1}%
}

% \end{macrocode}
% \end{macro}
%
% The following counters hold the number of photos per line
% and per column.
%
% \begin{macrocode}
\newcounter{photo@perline}
\newcounter{photo@percol}

```

These counters are set by the `\photosperline` and `\photospercol` macros.

`\photosperline`

```
\newcommand{\photosperline}[1]{%
  \setcounter{photo@perline}{#1}%
}

```

`\photospercol`

```
\newcommand{\photospercol}[1]{%
  \setcounter{photo@percol}{#1}%
}

```

The `\photo@date@color` macro contains the color used to display the date when the `showdate` option is active.

`\datecolor` It is set by the `\datecolor` macro.

```
\newcommand{\photo@date@color}{black}
\newcommand{\datecolor}[1]{%
  \renewcommand{\photo@date@color}{#1}%
}

```

The `\photo@date@style` macro contains the style used to display the date when the `showdate` option is active.

\datestyle It is set by the \datestyle macro.

```
\newcommand{\photo@date@style}{\sffamily\small}
\newcommand{\datestyle}[1]{%
  \renewcommand{\photo@date@style}{#1}%
}
```

The \photo@info@style macro contains the style used to display the information about photos in the contact sheet.

\infostyle It is set by the \infostyle macro.

```
\newcommand{\photo@info@style}{\sffamily\footnotesize}
\newcommand{\infostyle}[1]{%
  \renewcommand{\photo@info@style}{#1}%
}
```

The following lengths are the margins of the printing area with respect to the edges of a page in portrait orientation.

```
\newlength{\photo@top}
\newlength{\photo@left}
\newlength{\photo@bottom}
\newlength{\photo@right}
```

\photopaper The \photopaper macro sets the margins of the printing area. These margins are given in the order top, left, bottom, right and are measured with respect to the portrait orientation of the page.

```
\newcommand{\photopaper}[4]{% top left bottom right
  \setlength{\photo@top}{#1}
  \setlength{\photo@left}{#2}
  \setlength{\photo@bottom}{#3}
  \setlength{\photo@right}{#4}
  \calc@photomargins
}
```

The \calc@photomargins macro computes and sets the margins from the effective orientation of the paper and the size of the printing area.

photoprint uses a picture environment to place the photographs on each page. This environment must be opened at the beginning of a page, and closed at the end. The boolean photo@inpicture is used to know whether we are inside a picture environment or not.

```
\newboolean{photo@inpicture}
\setboolean{photo@inpicture}{false}
```

The index of the current photography is hold in the photo@num counter. The index of the first photography on the current page is stored in the photo@firstnum counter and is used to print the range of photographs on each page of a contact sheet.

```
\newcounter{photo@num}
\newcounter{photo@firstnum}
```

\photo@list is the stream used to read the name and number of copies of the photographs from the photoprint file.

```
\newread\photo@list
```

\printphotos The \printphotos macro reads the photoprint file and places the photographs on pages according to the document settings. Its optional argument is the name of the photoprint file, which defaults to photoprint.job.

```
\newcommand{\printphotos}[1][photoprint.job]{%
```

The first thing to do is to compute the margins and the spacing of photos according to the number of photos per line and per column, the dimensions of the printing area and the margins around pictures.

```
  \calc@photodim
```

Then, if we must print the date on the pictures, or if we are making a contact sheet, we check for a file named !Name_Map.can or !Name_Map.mav and read it to get the date of the pictures. If no such file can be found, we give a warning and disable the display of dates.

```
\ifthenelse{\boolean{photo@showdate}\OR\boolean{photo@contactsheet}}{%
  \IfFileExists{!Name_Map.can}{%
    \photo@read@map[!Name_Map.can]%
  }{%
    \IfFileExists{!Name_Map.mav}{%
      \photo@read@map[!Name_Map.mav]%
    }{%
      \ClassWarningNoLine{photoprint}{%
        No map file were found. Dates won't be shown.
      }%
      \setboolean{photo@showdate}{false}%
    }%
  }%
}{}%
```

After intializing the `photo@num` counter, we open the photoprint file and read it. The photographs are placed on the pages as their name is read from the file.

```
\setcounter{photo@num}{0}%
\immediate\openin\photo@list=#1
\photo@read
```

If the last photography is on an uncomplete page, the `picture` environment is still open. So we draw the crop marks if requested and then close the `picture` environment.

```
  \ifthenelse{\boolean{photo@inpicture}}{%
    \photo@drawticks
    \end{picture}%
  }{}%
}

```

We use the `photo@contactsheet` boolean to know whether we are building a contactsheet or preparing regular prints.

```
\newboolean{photo@contactsheet}

```

`\contactsheet` The `\contactsheet` macro is similar to `\printphotos` but builds a contact sheet instead of print sheets. It uses the same photoprint file as `\printphotos`, but ignores the number of copies and puts each photo once on the contactsheet.

`\contactsheet` uses a predefined layout and shows information (name and date) about each picture.

```
\newcommand{\contactsheet}[1][photoprint.job]{%
```

Since we use the `\printphotos` macro to build contact sheets, we must disable the `whitemargins` and `cropmarks` options, and inform `\printphotos` that we are building a contact sheet.

```
  \setboolean{photo@contactsheet}{true}%
  \setboolean{photo@whitemargins}{false}%
  \setboolean{photo@cropmarks}{false}%
```

We reduce the printing area to be able to put a title at the top and some information about the last row of pictures at the bottom. Since the printing area is defined in portrait orientation, what we call the top and bottom of the contact sheet depends on the orientation of the pages.

After changing these dimensions, we must update the lengths used by photoprint with `\calc@photomargins`.

```
  \ifthenelse{\boolean{photo@landscapepaper}}{%
    \addtolength{\photo@right}{1cm}%
    \addtolength{\photo@left}{0.5cm}%
  }{%
    \addtolength{\photo@top}{1cm}%
    \addtolength{\photo@bottom}{0.5cm}%
  }%
  \calc@photomargins
```

For contact sheets, we put 5 photographs per line and 4 per column (this is optimized for landscape A4 paper). We set the margins around the thumbnails to 2 mm horizontally, and to 4 mm vertically. The last margin is used to print the name, date and index of each photo below its thumbnail.

We then use `\printphotos` to layout the photos.

```
  \photosperline{5}%
  \photospercol{4}%
  \photohmargin{2mm}%
  \photovmargin{4mm}%
  \printphotos[#1]%
}
```

To make computations easier in the picture environment, we convert lengths to counters. `\photo@unitlength` is the length unit as a LaTeX length. `photo@unitconv` is the value of this length once stored in a LaTeX counter.

```
\newlength{\photo@unitlength}
\newcounter{photo@unitconv}
\setlength{\photo@unitlength}{0.1mm}
\setcounter{photo@unitconv}{\photo@unitlength}
```

`\photo@tmplen` is used to store length temporary. `photo@hgap` and `photo@vgap` are the horizontal and vertical photo margin converted to units of the `picture` environment.

```
\newlength{\photo@tmplen}
\newcounter{photo@hgap}
\newcounter{photo@vgap}
```

`\photohmargin` The `\photohmargin` sets the horizontal margin around photographs.

```
\newcommand{\photohmargin}[1]{%
  \setlength{\photo@tmplen}{#1}%
  \setcounter{photo@hgap}{\photo@tmplen / \value{photo@unitconv}}%
}
```

`\photovmargin` The `\photovmargin` sets the vertical margin around photographs.

```
\newcommand{\photovmargin}[1]{%
  \setlength{\photo@tmplen}{#1}%
  \setcounter{photo@vgap}{\photo@tmplen / \value{photo@unitconv}}%
}
```

`\photomargins` The `\photomargins` sets both the vertical and the horizontal margins around photographs to the same value.

```
\newcommand{\photomargins}[1]{%
  \photovmargin{#1}%
  \photohmargin{#1}%
}
```

The default is not to put margins around photographs.

```
\photomargins{0pt}
```

The `\photo@minmargin` length is the smallest margin of the printing area. It is used as the margin around photographs where the `whitemargins` option is active.

```
\newlength{\photo@minmargin}
```

`\calc@photomargins` The `\calc@photomargins` sets the margins of the document using the geometry package, and computes the smallest margin of the printing area.

Since the dimensions of the printing area are defined in portrait orientation, the parameters of the `\geometry` command depend on the effective orienttaion of the pages.

```
\newcommand{\calc@photomargins}{%
  \setlength{\headheight}{0pt}%
```

```
  \setlength{\headsep}{0pt}%
  \setlength{\footskip}{0pt}%
  \ifthenelse{\boolean{photo@landscapepaper}}{
    \setlength{\topmargin}{\photo@right-1in}%
    \setlength{\textheight}{\paperheight-\photo@right-\photo@left}%
    \setlength{\oddsidemargin}{\photo@top-1in}%
    \setlength{\evensidemargin}{\photo@top-1in}%
    \setlength{\textwidth}{\paperwidth-\photo@top-\photo@bottom}%
%     \geometry{%
%       nohead,nofoot,
%       top=\photo@right,
%       left=\photo@top,
%       bottom=\photo@left,
%       right=\photo@bottom
%     }
  }{
    \setlength{\topmargin}{\photo@top-1in}%
    \setlength{\textheight}{\paperheight-\photo@top-\photo@bottom}%
    \setlength{\oddsidemargin}{\photo@left-1in}%
    \setlength{\evensidemargin}{\photo@left-1in}%
    \setlength{\textwidth}{\paperwidth-\photo@left-\photo@right}%
%     \geometry{%
%       nohead,nofoot,
%       top=\photo@top,
%       left=\photo@left,
%       bottom=\photo@bottom,
%       right=\photo@right
%     }
  }
  \setlength{\photo@minmargin}{\photo@top}%
  \ifthenelse{\photo@left < \photo@minmargin}{%
    \setlength{\photo@minmargin}{\photo@left}%
  }{}%
  \ifthenelse{\photo@bottom < \photo@minmargin}{%
    \setlength{\photo@minmargin}{\photo@bottom}%
  }{}%
  \ifthenelse{\photo@right < \photo@minmargin}{%
    \setlength{\photo@minmargin}{\photo@right}%
  }{}%
}

```

The next counters contain the dimensions of the pages and of the printing slots in units of the `picture` environment.

```
\newcounter{paper@width}
\newcounter{paper@height}
\newcounter{photo@width}
\newcounter{photo@height}

```

We use the `photo@landscape` counter to store the orientation of the print slots. Its value is 1 if the width of the print slots is larger than their height (landscape orientation), and -1 in the other case. This convention allows us to use a product to determine if a photo has the same orientation as the prin slot. Using a boolean would have made things a bit more complex.

```
\newcounter{photo@landscape}

```

The next counters contain the x and y coordinate, and the number of the line and of the row of the current print slot.

```
\newcounter{photo@x}
\newcounter{photo@y}
\newcounter{photo@line}
\newcounter{photo@row}

```

The next counters contain the x and y coordinate, and the number of the line and of the row of the current print slot.

\calc@photodim The \calc@photodim macro computes the dimensions of the print slots and initializes the coordinates of the first print slot to use.

```
\newcommand{\calc@photodim}{%
```

If the whitemargins option is active, we set the margins around the photographs to the smallest margin of the print area.

```
  \ifthenelse{\boolean{photo@whitemargins}}{%
    \photomargins{\photo@minmargin}%
  }{}%
```

We convert the dimensions of the print area to counters for use in a picture environment.

```
  \setcounter{paper@width}{\textwidth / \value{photo@unitconv}}%
  \setcounter{paper@height}{\textheight / \value{photo@unitconv}}%
  %
```

The width of a print slot is the width of the print area minus the width of the spacing between photos, divided by the number of photos per line. The horizontal spacing between photos is twice the horizontal margin since each photo must have its own margin once it is cut.

```
  \setcounter{photo@width}{\value{paper@width}}%
  \addtocounter{photo@width}{-2*(\value{photo@perline}-1)*\value{photo@hgap}}
  \setcounter{photo@width}{\value{photo@width}/\value{photo@perline}}%
  %
```

The height of a print slot is the height of the print area minus the height of the spacing between photos, divided by the number of photos per column. The vertical spacing between photos is twice the vertical margin since each photo must have its own margin once it is cut.

```
  \setcounter{photo@height}{\value{paper@height}}%
  \addtocounter{photo@height}{-2*(\value{photo@percol}-1)*\value{photo@vgap}}
  \setcounter{photo@height}{\value{photo@height}/\value{photo@percol}}%
  %
```

Once we know the dimensions of the print slots, it is easy to determine if the photos will be printed in landscape or portrait orientation.

```
  \ifthenelse{\value{photo@width}>\value{photo@height}}{%
    \setcounter{photo@landscape}{1}%
  }{%
    \setcounter{photo@landscape}{-1}%
  }%
```

Last, we prepare for using the first print slot which is the leftmost one ($x = 0$) at the top of the page (y is at the top minus the height of the printing slot). This slot is on the first row of the first line.

```
  \setcounter{photo@x}{0}%
  \setcounter{photo@y}{\value{paper@height}-\value{photo@height}}%
  \setcounter{photo@line}{1}%
  \setcounter{photo@row}{1}%
}

```

The photo@ex counter is used to store the number of copies requested for the current photography.

```
\newcounter{photo@ex}

```

\parse@photo The \parse@photo macro is used to parse a line of the photoprint file. Each line contains the name of the picture file and the number of copies to print. The name and the number are separated by a *. Since the number part is optional, this macro is always used with *\end after its parameter.

When the number is omited, we set it to 1. When it is given, we have an extra * at its end, so we use another macro to remove it and set the photo@ex counter.

```
\def\parse@photo#1*#2\end{
  \gdef\photo@name{#1}%
  \if |#2|\relax
    \setcounter{photo@ex}{1}%
  \else
    \set@photo@ex#2%
```

```
  \fi
}
```

`\set@photo@ex` The `\set@photo@ex` macro is used to set the `photo@ex` counter to the value read in the photoprint file. This value has a trailing `*` which is eliminated by the argument list of this macro.

```
\def\set@photo@ex#1*{%
  \setcounter{photo@ex}{#1}%
}
```

`\photo@read` The `\photo@read` macro reads the photoprint file until its end or an empty line is reached.

```
\newcommand{\photo@read}{%
  \let\next\photo@read
  \ifeof\photo@list
    \immediate\closein\photo@list
    \let\next=\relax
  \else
    {%
      \endlinechar=-1
      \immediate\global\read\photo@list to \photo@name
    }%
    \if |\photo@name|
      \immediate\closein\photo@list
      \let\next=\relax
    \else
      \expandafter\parse@photo\photo@name*\end
      \ifphoto@contactsheet
        \setcounter{photo@ex}{1}%
      \else
        \message{Photo "\photo@name" x \thephoto@ex^^J}%
      \fi
      \stepcounter{photo@num}%
      \place@image
    \fi
  \fi
  \next
}

\newsavebox{\photo@box}
\newcounter{this@photo@width}
\newcounter{this@photo@height}
\newcounter{this@photo@landscape}
\newcounter{photo@angle}
\newcounter{photo@tmpcount}
\newcounter{photo@date@x}
\newcounter{photo@date@y}
\newcounter{photo@date@width}

\newlength{\photo@offset}
\newlength{\photo@width}
\newlength{\photo@height}
\newlength{\this@photo@width}
\newlength{\this@photo@height}
\newlength{\scaled@photo@width}
\newlength{\scaled@photo@height}

\newlength{\photo@trimleft}
\newlength{\photo@trimbot}
\newlength{\photo@trimright}
\newlength{\photo@trimtop}

\newboolean{photo@too@wide}

\newcommand{\place@image}{%
  \ifthenelse{\value{photo@ex}>0}{%
```

```
\global\let\next@place=\place@image
\ifthenelse{\(\value{photo@line}=1\)\and\(\value{photo@row}=1\)}{%
  \setlength{\unitlength}{\photo@unitlength}%
  \thispagestyle{empty}%
  \setcounter{photo@firstnum}{\value{photo@num}}%
  \begin{picture}(\value{paper@width},\value{paper@height})%
  \setboolean{photo@inpicture}{true}%
}{}%
\edef\photo@fname{\photo@dir\photo@name}%
\savebox{\photo@box}{\includegraphics{\photo@fname}}%
\setlength{\this@photo@width}{\wd\photo@box}%
\setlength{\this@photo@height}{\ht\photo@box}%
\setcounter{this@photo@width}{\this@photo@width / \value{photo@unitconv}}%
\setcounter{this@photo@height}{\this@photo@height / \value{photo@unitconv}}%
\ifthenelse{\this@photo@width > \this@photo@height}{%
  \setcounter{this@photo@landscape}{1}%
}{%
  \setcounter{this@photo@landscape}{-1}%
}%
\setcounter{photo@tmpcount}{%
  \value{photo@landscape}*\value{this@photo@landscape}%
}%
\ifthenelse{\value{photo@tmpcount} < 0}{%
  \setcounter{photo@angle}{90}%
  \setcounter{photo@tmpcount}{\value{this@photo@width}}%
  \setcounter{this@photo@width}{\value{this@photo@height}}%
  \setcounter{this@photo@height}{\value{photo@tmpcount}}%
  \setlength{\photo@tmplen}{\this@photo@width}%
  \setlength{\this@photo@width}{\this@photo@height}%
  \setlength{\this@photo@height}{\photo@tmplen}%
}{%
  \setcounter{photo@angle}{0}%
}%
\setlength{\unitlength}{\photo@unitlength}%
\setlength{\photo@width}{\value{photo@width}\unitlength}%
\setlength{\photo@height}{\value{photo@height}\unitlength}%
%
\setlength{\scaled@photo@width}{%
  \this@photo@width*\ratio{\photo@height}{\this@photo@height}%
}%
\setlength{\scaled@photo@height}{%
  \this@photo@height*\ratio{\photo@width}{\this@photo@width}%
}%
\ifthenelse{%
  \(\(\scaled@photo@width>\photo@width\)\AND\NOT\boolean{photo@fullpictures}\)
  \OR
  \(\(\scaled@photo@width<\photo@width\)\AND\boolean{photo@fullpictures}\)
}{%
  \setboolean{photo@too@wide}{true}%
}{%
  \setboolean{photo@too@wide}{false}%
}
\ifthenelse{\boolean{photo@too@wide}}{%
  % photo is too wide
  \setlength{\photo@offset}{%
    (\this@photo@width-\photo@width*\ratio{\this@photo@height}{\photo@height})/2
  }%
  \ifthenelse{\value{photo@angle}=0}{%
    \setlength{\photo@trimleft}{\photo@offset}%
    \setlength{\photo@trimbot}{0pt}%
  }{%
    \setlength{\photo@trimleft}{0pt}%
    \setlength{\photo@trimbot}{\photo@offset}%
  }%
}{%
  % photo is too high
  \setlength{\photo@offset}{%
    (\this@photo@height-\photo@height*\ratio{\this@photo@width}{\photo@width})/2
  }%
```

```
401       \ifthenelse{\value{photo@angle}=0}{%
402         \setlength{\photo@trimleft}{0pt}%
403         \setlength{\photo@trimbot}{\photo@offset}%
404       }{%
405         \setlength{\photo@trimleft}{\photo@offset}%
406         \setlength{\photo@trimbot}{0pt}%
407       }%
408     }
409     \setlength{\photo@trimright}{\photo@trimleft}%
410     \setlength{\photo@trimtop}{\photo@trimbot}%
411     \edef\photo@fname{\photo@dir\photo@name}%
412     \put(\thephoto@x,\thephoto@y){%
413       \includegraphics[%
414         angle=\thephoto@angle,%
415         trim=\photo@trimleft{} \photo@trimbot{} \photo@trimright{} \photo@trimtop{},%
416         width=\photo@width,%
417         height=\photo@height,%
418         clip%
419       ]{\photo@fname}}%
420     \ifthenelse{\boolean{photo@contactsheet}}{%
421       \put(\thephoto@x,\thephoto@y){\framebox(\thephoto@width,\thephoto@height){}}%
422       \setcounter{photo@tmpcount}{\value{photo@y}-10}%
423       \put(\thephoto@x,\thephoto@tmpcount){%
424         \makebox(\thephoto@width,0)[ts]{%
425           {\photo@info@style
426            \photo@get@date{\photo@name}\hss
427            % \photo@name may contain special characters
428            \escapechar=-1{\expandafter\string\csname\photo@name\endcsname}\hss
429           }\textsf{\thephoto@num}%
430         }%
431       }%
432     }{%
433       \ifthenelse{\boolean{photo@showdate}}{%
434         \ifthenelse{\value{photo@angle}>0}{%
435           \setcounter{photo@date@x}{\value{photo@x}+\value{photo@width}-20}%
436           \setcounter{photo@date@y}{\value{photo@y}}%
437           \setcounter{photo@date@width}{\value{photo@height}-20}%
438           \put(\thephoto@date@x,\thephoto@date@y){%
439             \rotatebox{90}{%
440               \makebox(\value{photo@date@width},0)[br]{%
441                 \photo@date@style{%
442                   \textcolor{\photo@date@color}{\photo@get@date{\photo@name}}%
443                 }%
444               }%
445             }%
446           }%
447         }{%
448           \setcounter{photo@date@x}{\value{photo@x}}%
449           \setcounter{photo@date@y}{\value{photo@y}+20}%
450           \setcounter{photo@date@width}{\value{photo@width}-20}%
451           \put(\thephoto@date@x,\thephoto@date@y){%
452             \makebox(\value{photo@date@width},0)[br]{%
453               \textcolor{\photo@date@color}{%
454                 \photo@date@style\photo@get@date{\photo@name}%
455               }%
456             }%
457           }%
458         }%
459       }{}%
460     }%
461     %
462     \addtocounter{photo@ex}{-1}%
463     \stepcounter{photo@row}%
464     \ifthenelse{\value{photo@row}>\value{photo@perline}}{%
465       \setcounter{photo@row}{1}%
466       \stepcounter{photo@line}%
467       \setcounter{photo@x}{0}%
468       \addtocounter{photo@y}{-\value{photo@height}}%
469       \addtocounter{photo@y}{-2*\value{photo@vgap}}%
```

```
    }{%
      \addtocounter{photo@x}{\value{photo@width}}%
      \addtocounter{photo@x}{2*\value{photo@hgap}}%
    }%
    \ifthenelse{\value{photo@line}>\value{photo@percol}}{%
      \setcounter{photo@row}{1}%
      \setcounter{photo@line}{1}%
      \setcounter{photo@x}{0}%
      \setcounter{photo@y}{\value{paper@height}-\value{photo@height}}%
      \photo@drawticks
      \end{picture}\newpage
      \setboolean{photo@inpicture}{false}%
    }{}%
  }{%
    \let\next@place=\relax
  }%
  \next@place
}

\newcounter{photo@tick@x}
\newcounter{photo@tick@y}
\newcounter{photo@tick@line}
\newcounter{photo@tick@col}
\newcommand{\photo@drawticks}{%
  \ifthenelse{\boolean{photo@cropmarks}}{%
    \setcounter{photo@tick@col}{-1}%
    \setcounter{photo@tick@x}{-\value{photo@hgap}}%
    \setcounter{photo@tick@y}{-\value{photo@vgap}-1}%
    \whiledo{\value{photo@tick@col}<\value{photo@perline}}{%
      \put(\thephoto@tick@x, \thephoto@tick@y){\line(0,-1){40}}%
      \stepcounter{photo@tick@col}%
      \addtocounter{photo@tick@x}{2*\value{photo@hgap}}%
      \addtocounter{photo@tick@x}{\value{photo@width}}%
    }
    \setcounter{photo@tick@line}{-1}%
    \setcounter{photo@tick@x}{\value{paper@width}+\value{photo@hgap}+1}%
    \setcounter{photo@tmpcount}{-\value{photo@hgap}-1}%
    \setcounter{photo@tick@y}{-\value{photo@vgap}}%
    \whiledo{\value{photo@tick@line}<\value{photo@percol}}{%
      \put(\thephoto@tmpcount, \thephoto@tick@y){\line(-1,0){40}}%
      \put(\thephoto@tick@x, \thephoto@tick@y){\line(1,0){40}}%
      \stepcounter{photo@tick@line}%
      \addtocounter{photo@tick@y}{2*\value{photo@vgap}}%
      \addtocounter{photo@tick@y}{\value{photo@height}}%
    }%
    \setcounter{photo@tick@col}{-1}%
    \setcounter{photo@tick@x}{-\value{photo@hgap}}%
    \setcounter{photo@tick@y}{\value{paper@height} + \value{photo@vgap} + 1}%
    \whiledo{\value{photo@tick@col}<\value{photo@perline}}{%
      \put(\thephoto@tick@x, \thephoto@tick@y){\line(0,1){40}}%
      \stepcounter{photo@tick@col}%
      \addtocounter{photo@tick@x}{2*\value{photo@hgap}}%
      \addtocounter{photo@tick@x}{\value{photo@width}}%
    }%
  }{}%
  \ifthenelse{\boolean{photo@contactsheet}}{%
    \setcounter{photo@tick@y}{\value{paper@height}+50}%
    \setcounter{photo@tick@x}{0}%
    \put(\thephoto@tick@x,\thephoto@tick@y){\line(1,0){\value{paper@width}}}%
    \addtocounter{photo@tick@y}{10}%
    \put(\thephoto@tick@x,\thephoto@tick@y){\textsf{\textbf{\large PhotoIndex}}}%
    \put(\thephoto@tick@x,\thephoto@tick@y){%
      \makebox(\value{paper@width},0)[br]{%
        \textsf{\large\thephoto@firstnum{} -- \thephoto@num}%
      }%
    }%
    \put(\thephoto@tick@x,\thephoto@tick@y){%
      \makebox(\value{paper@width},0)[b]{%
        \textsf{\large\bfseries\csname cstitle\endcsname}%
```

```
      }%
    }%
  }{}%
}
%%
% Map reading
%
\newread\photo@mapfile

\newcommand{\photo@readmap@entry}{%
  \let\next@read=\relax
  \ifeof\photo@mapfile
    \immediate\closein\photo@mapfile
  \else
    {\endlinechar=-1\immediate\global\read\photo@mapfile to \photo@map@entry}%
    \if|\photo@map@entry|
      \immediate\closein\photo@mapfile
    \else
      \let\next@read=\photo@readmap@entry
      \photo@parse@entry\photo@map@entry
    \fi
  \fi
  \next@read
}

\newcommand{\photo@parse@entry}[1]{%
  \expandafter\photo@eval@entry#1
}

\newcommand{\photo@eval@entry}[4]{%
  \expandafter\expandafter\expandafter
  \def\expandafter\expandafter
    \csname #2\endcsname\expandafter{\photo@parse@isodate#4\end}%
}

\newcommand{\photo@parse@isodate}{}
\def\photo@parse@isodate#1T#2\end{%
  #1%
}

\newcommand{\photo@get@date}[1]{%
  \expandafter\csname#1\endcsname
}

\newcommand{\photo@read@map}[1][!Name_Map.can]{%
  \immediate\openin\photo@mapfile=#1
  \photo@readmap@entry
}

% Default settings
\photopaper{0.5cm}{0.5cm}{1cm}{0.5cm} % default for Canon S800
\photosperline{2}
\photospercol{2}
⟨/class⟩
```

10 Helper files

10.1 Creating a contactsheet

The file contactsheet.tex can be used to create a contactsheet using the photoprint.job file in the current directory, for instance with the command: `pdflatex contactsheet`.

```
⟨*contactsheet⟩
\documentclass[a4paper,landscape]{photoprint}
\begin{document}
  \contactsheet%[name-of-photo-list] % defaults to photoprint.job
\end{document}
⟨/contactsheet⟩
```

10.2 Printing regular size photos

The file photos.tex can be used to print four photos per page (what makes 10 cm by 13 cm prints on A4 paper) using the photoprint.job file in the current directory, for instance with the command: `pdflatex photos`.

599 ⟨*photos⟩
```
600 \documentclass[a4paper,landscape,cropmarks]{photoprint}
601 \photosperline{2}
602 \photospercol{2}
603 \begin{document}
604   \printphotos%[name-of-photo-list] % defaults to photoprint.job
605 \end{document}
```
606 ⟨/photos⟩

10.3 Printing large photos

The file largesphotos.tex can be used to print two photos per page (what makes 14 cm by 18 cm prints on A4 paper) using the photoprint.job file in the current directory, for instance with the command: `pdflatex largephotos`.

607 ⟨*largephotos⟩
```
608 \documentclass[a4paper,landscape,cropmarks]{photoprint}
609 \photosperline{2}
610 \photospercol{1}
611 \begin{document}
612   \printphotos%[name-of-photo-list] % defaults to photoprint.job
613 \end{document}
```
614 ⟨/largephotos⟩

10.4 Building a photoprint.job file

A quick way to build a photoprint.job file is to run the mkphoto shell script in a directory. This will add all JPEG pictures to the file, with a number of copies set to 0. It can be edited later to set the effective number of copies to print.

Note that docstrip will extract this file as mkphoto., so you will have to remove the dot at the end of its name. You will also have to make it executable, for instance with the Unix command `chmod 555 mkphoto`.

```
<*mkphoto>
cat /dev/null > photoprint.job
for f in *.[jJ][pP][gG] *.[jJ][pP][eE][gG]
do
  echo "$f*0" >> photoprint.job
done
</mkphoto>
```

For Windows users, Peter Zimmerman has written a PPrlist.bat batch file that does the same thing as mkphoto:

```
<*PPrlist>
@echo off
REM PPrlist.bat
REM Author: Peter Zimmerman
REM Create jpg-filelist for PhotoPrint.cls
if exist "Photoprint.jbu" del    Photoprint.jbu > nul
if exist "Photoprint.job" echo   "Backup Photoprint.job --> Photoprint.jbu"
if exist "Photoprint.job" rename Photoprint.job Photoprint.jbu
for %%e in (*.jpg) do echo %%e*1 >> Photoprint.job
if exist "Photoprint.job" echo   "Photoprint.job created."
</PPrlist>
```

10.5 Contactsheet with name of directory

To create a contactsheet which shows the name of the directory containing the pictures (useful to know where to look when someone wants copies of the pictures), you can use the following mkcs shell script.

Note that docstrip will extract this file as mkcs., so you will have to remove the dot at the end of its name. You will also have to make it executable, for instance with the Unix command `chmod 555 mkcs`.

```
<*mkcs>
pdflatex \\def\\cstitle{‘basename "$PWD"‘} \\input contactsheet
rm contactsheet.aux contactsheet.log
</mkcs>
```