

The png2pdf program

Dipl.-Ing. D. Krause

June 24, 2005

Contents

1	Overview	2
2	Installation	3
2.1	Installation from source on UNIX systems	3
2.1.1	Prerequisites	3
2.1.2	Installation procedure	3
2.2	Executable setup for Windows	4
3	Usage	5
3.1	Common dkapp options	5
3.1.1	Help and version information	5
3.1.2	Permanent options handling	5
3.2	Png2pdf options	6
4	Examples	9
4.1	Overview	9
4.2	Explicit background color	10
4.3	Colored background	11
4.4	Background effects	12
4.4.1	Mixing against background color	12
4.4.2	Adding an image mask	13
4.4.3	Using alpha channel	13
4.5	Image interpolation	14
4.6	Beamer presentation example	15

1 Overview

The png2pdf program can be used to convert PNG image files to PDF files. If the PNG file contains an alpha channel one can mix the image against a background color and/or create a PDF image mask or PDF transparency data.

The program is mainly intended for pdfLaTeX users who want to create flyers or posters having colored background.

2 Installation

2.1 Installation from source on UNIX systems

2.1.1 Prerequisites

The following software packages should be installed before you start to install png2pdf:

- **zlib** (required)
A general purpose compression library¹.
- **libpng** (required)
A library for dealing with PNG images².
- **libbz2** (recommended)
Another compression library³.
- **dklibs** (required)
Dirk Krause's libraries⁴.

The png2pdf software can be retrieved on the SourceForge project website⁵. Documentation is available via the project homepage also hosted at SourceForge⁶.

2.1.2 Installation procedure

zlib, libpng, libbz2 and dklibs

These libraries are installed as usual by running

```
1 ./configure
2 make
3 make install
```

¹<http://www.gzip.org/zlib>

²<http://www.libpng.org/pub/png/libpng.html>

³<http://sources.redhat.com/bzip2>

⁴<http://sourceforge.net/projects/dklibs>

⁵<http://sourceforge.net/projects/png2pdf>

⁶<http://png2pdf.sourceforge.net>

png2pdf

Make sure the environment variables CFLAGS and LDFLAGS are set properly to point into the “include” and “lib” subdirectories of the installation directories for the libraries listed above. The CFLAGS variable should contain something like “-I/usr/local/include”, the LDFLAGS variable “-L/usr/local/lib”.

Both variables must not point to the directory created while unpacking the dklibs source archive and used to build dklibs! Reason: There are two different versions of the “dkconfig.h” file. An initial version is shipped with the dklibs distribution, it includes the “config.h” file created by the configure script. During installation a final version is written to “/usr/local/include/dkconfig.h”, this file must be used by applications linked against the dklibs libraries. If the initial version is used instead, it loads the “config.h” file but finds the application’s file instead of the file created for the dklibs project. Many #define instructions will be missing in this case!

After making sure to have the correct values for CFLAGS and LDFLAGS run

```
1 ./configure
2 make
3 make install
```

to install the software.

2.2 Executable setup for Windows

For Windows users there is a setup.exe available via the dklibs project website ⁷. In the file releases section you find dklibs-win32-*.*. * containing two files dklibs-win32-*. * .*-setup.exe. Choose the highest version number for the recent version. The “user” file installs the dklibs-based software and documentation, the “dev” file also installs all used sources. *Note:* The “dev” file is significantly larger.

⁷<http://sourceforge.net/projects/dklibs>

3 Usage

3.1 Common dkapp options

The program uses the dklibs library set for preferences handling, localization, logging etc. At program startup the permanent options are read first before processing command line options. The options used are the result of combining the permanent options and the command line options.

3.1.1 Help and version information

- -h
--help
shows a help text.
- -v
--version
shows version information.

3.1.2 Permanent options handling

- -c *options*
--configure *options*
configures permanent options (preferences).
- -u
--unconfigure
removes all permanent options.
- -C
--show-configuration
shows the permanent options.
- -r *options*
--reset *options*
overrides the permanent options for this program invocation and uses the command line options only.

3.2 Png2pdf options

- -p 1.2
-p 1.3
-p 1.4
--pdf-level=1.2
--pdf-level=1.3
--pdf-level=1.4
chooses the PDF format version.
- -m *r,g,b*
--mix-background=*r,g,b*
sets a default background color for color mixing. This background color is used if the PNG file does not contain a background chunk or if the -s option is used (see below).
-m-
--mix-background=-
turns explicit mixing off and leaves this up to the libpng library.
- -s
--mix-specified
ignores the background chunk contained in the PNG file. The program uses the background specified in the -m option (see above) regardless whether or not the PNG file contains a background chunk.
-s-
--mix-specified=off
disables this behaviour.
- -a
--transparency
passes alpha channel data from the PNG file to the PDF file. This feature is available for PDF format 1.4 only.
-a-
--transparency=off
denies writing of alpha channel information to the PDF file.
- -i
--image-mask
converts the alpha channel from the PNG file into a PDF image mask. This requires PDF format 1.3 or above. By default all pixels having opacity 0 are masked out. The -l option (see below) inverts the level so only pixels with

full opacity are drawn, all others are masked out.

-i-

--image-mask=off

disables image mask generation.

- -l

--invert-levels

can be used in conjunction with -i to invert the image mask level.

-l-

--invert-levels=off

switches back to normal image mask level.

- -t

--alpha-transparency

indicates that the alpha channel contains transparency data.

By default alpha channel data is interpreted as opacity.

- -t-

--alpha-transparency=off

switches back to default behaviour (alpha channel is opacity).

- -b

--fill-background

can be used in conjunction with -a or -i to fill the image area with the default background color before applying the image.

-b-

--fill-background=off

turns background filling off.

- -n

--interpolation

adds an entry to the image dictionary indicating that viewers should attempt an image interpolation to obtain smoother transitions.

Use with care. Image interpolation is not supported for all output devices.

Different viewers may use different algorithms. So the representation of the image may vary depending on the viewer software and output device.

- -f

--file-time-check

inspects the modification time of source PNG and destination PDF file when

png2pdf is run on a directory.

-f-

--file-time-check=off

turns modification time inspection off (and forces a conversion for all files).

4 Examples

4.1 Overview

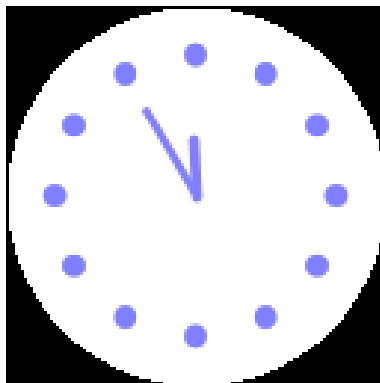


Figure 1: The original image, PNG file directly included by pdfL^AT_EX

In the examples section we use the image `fbt.png`. This image contains a clock showing 5 before 12. The hour dots and the center of the clock are fully opaque, the white center becomes more and more transparent when moving from the image center to the image borders. See figure 1. The image was included using

```
1 \begin{figure}[ht]
2   {\centering
3    \includegraphics[width=5cm]{fbt.png}
4    \caption{The original image}
5    \label{img:fbt}
6   }
7 \end{figure}
```

The LaTeX source must be processed using pdfLaTeX three times to make sure to have all labels and bookmarks recognized. The next step in processing is to run `ppower4` to have the background effects. Among others the packages `color`, `ifthen`, `ifpdf`, `background` and `pause` are needed for the examples.

4.2 Explicit background color

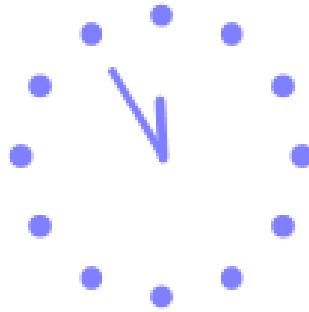


Figure 2: Now mixed against white background

Using the commands

```
1 bmeeps -p2 -c -er8 fbt.png x1.eps
2 png2pdf -m 255,255,255 -s fbt.png x1.pdf
```

we create an EPS and a PDF version of the image. The image is included using

```
1 \begin{figure}[ht]
2   {\centering
3   \ifpdf
4   \includegraphics[width=5cm]{x1.pdf}
5   \else
6   \includegraphics[width=5cm]{x1.eps}
7   \fi
8   \caption{Now mixed against a white background}
9   \label{img:b2}
10  }
11 \end{figure}
```

4.3 Colored background

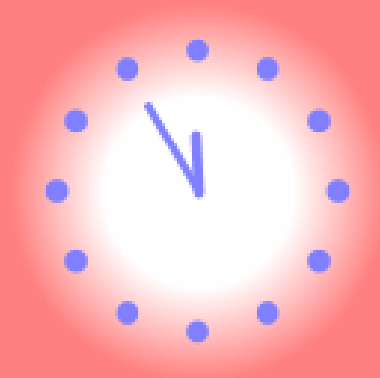


Figure 3: Now mixed against light red background

Now we create a light red background using

```
1 \pagecolor{bgred}
```

The images files are created by running

```
1 bmeps -p3 -c -e8f -am255,128,128 s fbt.png x2.eps
2 png2pdf -m 255,128,128 -s fbt.png x2.pdf
```

This image was included using

```
1 \begin{figure}[ht]
2 {\centering
3 \ifpdf
4 \includegraphics[width=5cm]{x2.pdf}
5 \else
6 \includegraphics[width=5cm]{x2.eps}
7 \fi
8 \caption{Now mixed against light red background}
9 \label{img:b3}
10 }
11 \end{figure}
```

4.4 Background effects

4.4.1 Mixing against background color

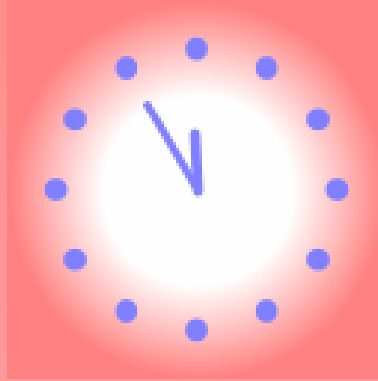


Figure 4: Still the same image mixed against light red background

On this page we have a color effect, the top of the page is light red as on the previous page, the bottom is lighter. Including the images from section 4.3 on the page before results in a visible rectangle background, see figure 4. Depending on the PDF level we can use we have two choices:

- Mixing against a constant background and adding an image mask (PDF level 1.3 and above) or
- Passing the alpha channel to the PDF file (PDF level 1.4).

4.4.2 Adding an image mask

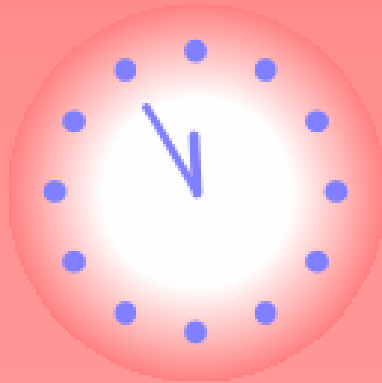


Figure 5: Image containing image mask

We create a PDF file containing an image mask by

- 1 `png2pdf -i -m 255,128,128 -s fbt.png x3.pdf`
- 2 `cp x2.eps e3.eps`

The result can be seen in figure 5.

4.4.3 Using alpha channel

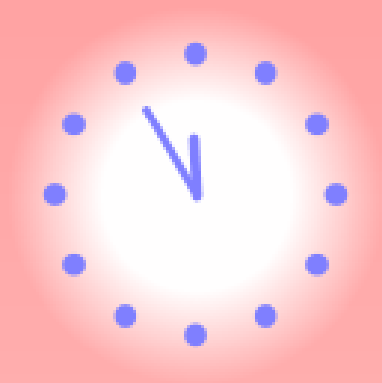


Figure 6: Image containing alpha channel

Best quality is achieved when passing the PNG file's alpha channel to the PDF file by running

- 1 `png2pdf -a -m 255,128,128 -s fbt.png x4.pdf`

The result can be seen in figure 6.

4.5 Image interpolation

This section shows the impact of the “-n” option used to add an “interpolate image” entry to the image directory. The logo was converted to PDF twice, the first example (figure 7) comes without image interpolation, the second examples (figure 8) uses interpolation.

To view the difference zoom into the page to have the images filling the screen width.



Figure 7: Image without interpolation



Figure 8: Interpolated image

4.6 Beamer presentation example

To embed partial transparent PDF graphics into beamer presentations, use “\usepackage{graphicx}” and “\includegraphics{...}”.

The example uses an image png2pdf-logo.pdf produced by

```
1  png2pdf -a -n png2pdf-logo.png png2pdf-logo.pdf
```

which is placed over a color gradient background.

```
1 \documentclass{beamer}
2 \mode<presentation>{\usetheme{Madrid}\setbeamercovered{transparent}}
3 \usepackage[german]{babel}
4 \usepackage[latin1]{inputenc}
5 \usepackage[T1]{fontenc}
6 \usepackage{ifpdf}
7 \usepackage{graphicx}
8 \usepackage{color}
9 \ifpdf
10 \hypersetup{pdfpagemode=FullScreen}
11 \fi
12 \title[Beamer and png2pdf]{Using png2pdf with the beamer class}
13 \author[Krause]{D.~Krause}
14 \subject{png2pdf}
15 \begin{document}
16 \beamertemplateshadingbackground{yellow!50}{magenta!50}
17 \begin{frame}
18 \frametitle{Image over transient background}
19 \includegraphics[width=5cm]{x4.pdf}
20 \end{frame}
21 \end{document}
```

The result can be seen in [beamtest.pdf](#).