

The `pdfcrack` manual

Marc Boyer-Gandon
IRIT/CNRS – ENSEEIHT
France
Marc.Boyer@enseeiht.fr

June 11, 2008

Contents

1	Introduction	1
1.1	Why <code>pdfcrack</code> ?	2
1.2	Basic idea	2
2	Using it	2
2.1	Installation	2
2.2	A very short tutorial	2
2.3	Limitations	4
2.4	Options	4
3	How does it works?	5
4	Knowns bugs and limitations	6
5	Why this name ?	6
6	Other solutions	6

1 Introduction

What is `pdfcrack`? The `pdfcrack` is a *hack* that allow to use `psfrag` and `pdflatex`.

The `pdfcrack` is not a *complete* solution: it does not handle all \LaTeX files, and you will sometimes need to modify your \LaTeX source files if you want to use it. But it can help you to save time.

1.1 Why pdfcrack?

If you want to generate a pdf file from a L^AT_EX one, you either can do it directly, using pdf_lat_ex, or by first generating a postscript¹ file and converting this postscript to a pdf file.

The psfrag package allow the user to replace some text in a postscript figure by another L^AT_EX text. To use psfrag, you must use a filter from dvi to postscript. So, if you compile with pdf_lat_ex, the text substutions are lost. Nevertheless, you sometimes want to use pdf_lat_ex, to be able to add hyperlinks in your pdf, or because your postscript to pdf filter produces a ugly text.

1.2 Basic idea

The basic idea of pdfcrack is, from your L^AT_EX source file, to produce the figures in pdf format, *with the psfrag replacements*. Then, you can compile with pdf_lat_ex, including the pdf figures.

2 Using it

2.1 Installation

Put the script pdfcrack.sh in a directory included in your path, and pdfcrack.sty somewhere where T_EX will find it².

Moreover, the pdfcrack.sh script uses a lot of other scripts and software. (cut, dvips, epstopdf, grep, head, latex, ps2ps, ps2epsi, sort, tail, and, first of all, a bourne shell). They all are installed with common Unix/Linux distribution, so, you should not have to care about it.

2.2 A very short tutorial

MAKE A BACKUP OF YOUR FILES BEFORE USING THIS SCRIPTS !

Assuming you have a L^AT_EX file, with figures included with

```
\begin{figure}[htbp]
\centering
\psfrag{Fp(x)}{ $\mathcal{F}(x)$ }
\psfrag{Gp(x)}{ $\mathcal{G}(x)$ }
\includegraphics[width=\textwidth]{BasicIdea}
\caption{The basic idea of \pdfcrack}
\label{fig:BasicIdea}
\end{figure}
```

you simply have to:

¹With L^AT_EX; you generate a dvi file, and, with some filter, like dvips, you get a postscript one.

²The variable TEXINPUTS defines where your T_EXtool looks for inputs.

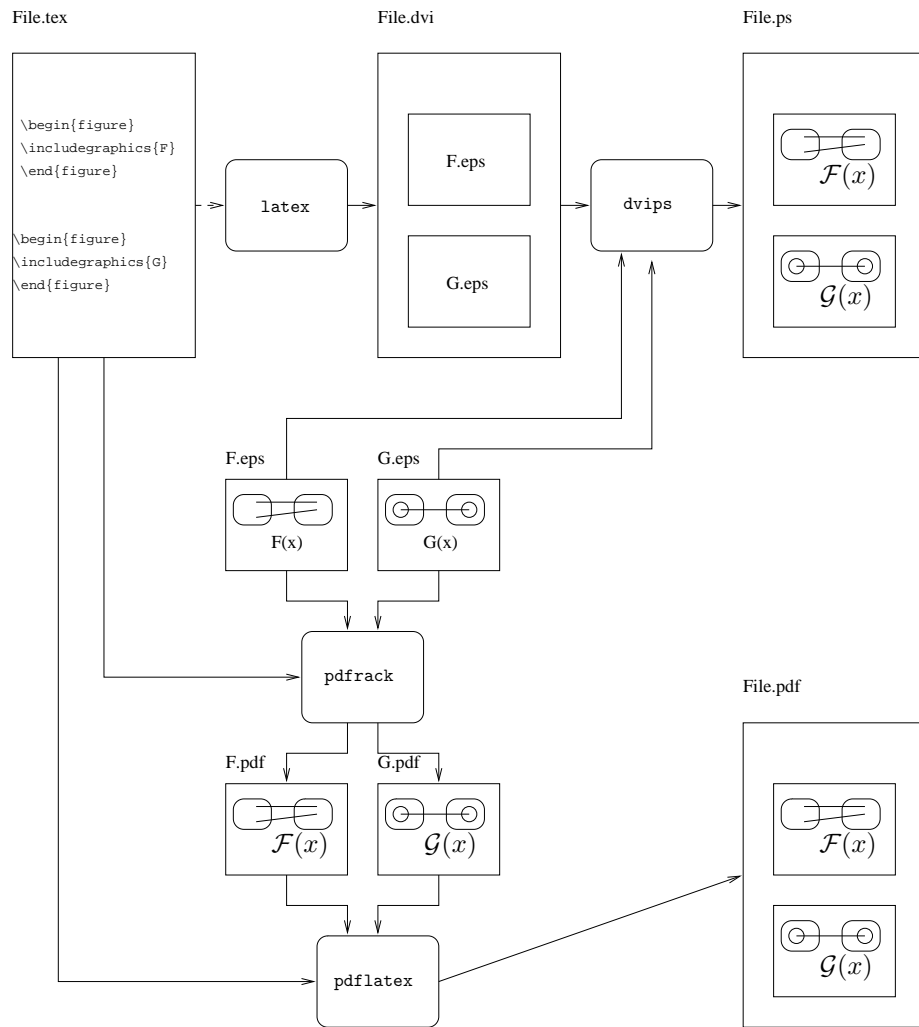


Figure 1: The basic idea of `pdfrack`

- include the `pdfrack` package `\usepackage{pdfrack}`
- replace the `\includegraphics` command by `\pdfrackincludegraphics`

Then, just run the command `pdfrack.sh` on you file. It should produce some output like:

```
-----
---> Converting figure BasicIdea
Using LaTeX: producing dvi
This is e-TeX, Version 3.14159-2.1 (Web2C 7.4.5)
entering extended mode
Method 2: Trying conversion with dvips -|-> ps2eps -|-> eps2eps -|-> epstopdf
Converting dvi -> ps (dvips)
Converting ps -> eps (ps2eps)
    Cleaning eps with eps2eps filter
Converting eps -> pdf (epstopdf)
Removing temporary files
```

for each figure, and generates a standalone pdf figure *with the psfrag replacements*.

Then, you can run `pdflatex` on your file.

Often, the bounding box of the generated file is not very good. Try different values for the `-m` options, and try to use or no the `-p` and `-e` options. It offers you 16 possibilities. If none work, you are unlucky.

See also the `README` file to have more details.

2.3 Limitations

The `pdfrack` package is based on a bourne-shell script that parse the file to find the `\begin{figure}` and `\end{figure}` tags. So, you must use this to way of including figure (or change the script).

Note that the `%` is is general seen as a comment, even if prefixed with `\`. So, avoid to use it.

If your `LATEX`code is too complex (if your figures environments are in included files for example), then, `pdfrack` will ne be able to handle it. Then, you should write a simpler `LATEX`file, with your figure(s) and use this file to generate the pdf figures.

2.4 Options

`-h` help

`-m` method: integer in 0..3 (default is 2) the number of the method used to convert `dvi` to `pdf` if one method fails, try another, and add `-p` and/or `-e` options :-)

`-p` filter each postscript file with `ps2ps`

- e filter each encapsulated postscript file with `eps2eps`
- i force use of `ps2epsi` instead of `ps2eps`
- k keep (all temporary files are kept, usefull for debug)
- H own header file (default is to extract from file.tex file)

3 How does it works?

The core of `pdfrack` is a `pdfrack.sh` script that try, for each figure in your \LaTeX source, to produce a small \LaTeX file with only the figure. Then, this file is compiled with \LaTeX , converted into `postscript`, and then, we have a `postscript` figure, with the replacements.

Now, the system should convert a `postscript` file to a `pdf` one, with the right bounding box... I am not at all a guru of `postscript` (neither `pdf`), so, I try to use some tools like `ps2ps`, `ps2eps`, `ps2pdf` and so on. There is an option in the script `-m` that allow the user to chose one or the other.

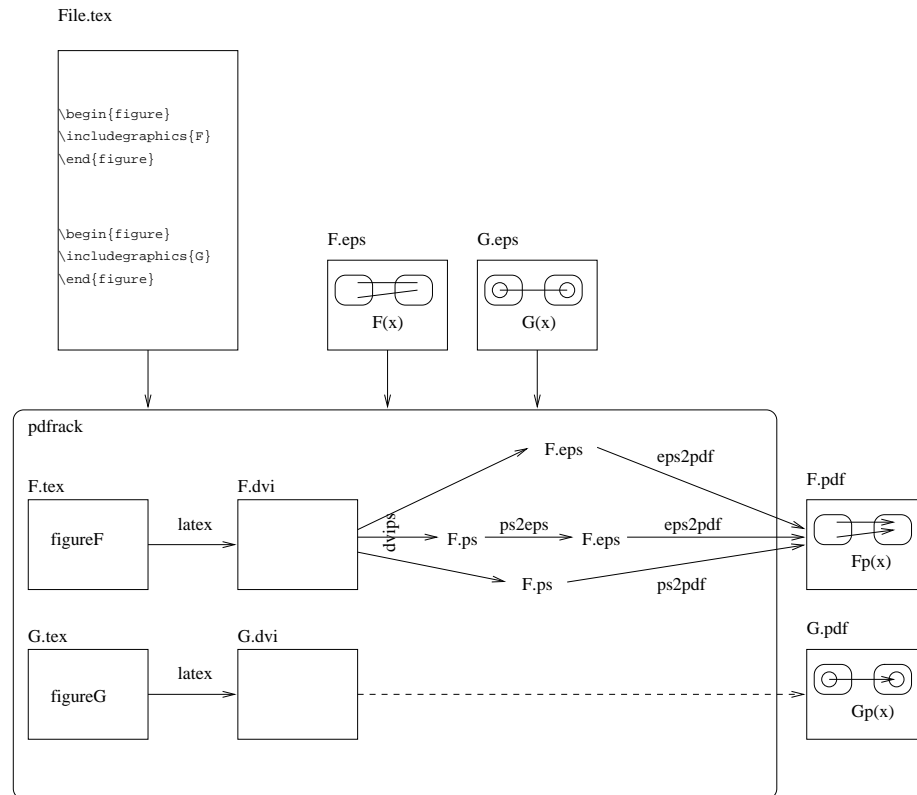


Figure 2: Inside `pdfrack`

4 Knowns bugs and limitations

1. **Only a subpart of my figure appears** The main problem is the computation of the “Bounding box” of the postscript figure. If the default method fails, try the `-m` option, with different values. Adding the `-e` may also help.
2. **The figure is alone on a full page. Its seems to big.**
See item 1
3. **I am a Windows user**
Neither am I³... This solution is based on some well known Unix command-line tools, like the bourne shell, `ps2eps`, `cut`, `tail`, `grep`... To use my script on windows, you should install a solution like Cygwin.
4. **My problem is not in the known bugs list !**
Just email me.

5 Why this name ?

I think a good solution for using `pdflatex` and `psfrag` should be named `pdfrag` or `pdffrag`. I do not think that my solution is really good. It is just a hack. Then `pdfcrack` comes from:

```
pdf
+ psfrag
+   hack
-----
pdfcrack
```

6 Other solutions

There are, of course, other solutions:

DrawAt Matthijs Douze (another member from ENSEEIHT) has developed DrawAt, a solution for MacOS X only. <http://www.enseeiht.fr/~douze/drawat/index.html>

unpsfrag Félix Valado Pomarinho has developed a perl script: `unpsfrag`

fragmaster Tilman Vogel developed `fragmaster`, another perl-oriented solution. <http://www.tat.physik.uni-tuebingen.de/~vogel/fragmaster>

³This is not always true: when I am forced to write Word document, I use a Windows machine.

pstoedit and [X]fig With `[X]fig`, you can already make a figure and, setting the special flag to a text zone, you already can put on your figures some LaTeX code, and export in the combined mode either in postscript or pdf.

In the combined Postscript/LaTeX, if your file is named `figure.fig`, it creates a `figure.pstex` file which is the postscript version of your fig figure *without the special-tagged texts*, and a `figure.pstex.t` file which just include the `figure.pstex` (with the `\includegraphics` command) *and* adds the LaTeX text at the right place.

In the combined Pdf/LaTeX, this is the same except that `figure.pstex` is named `figure.pdf` and `figure.pstex.t` is named `figure.pdf.t` and includes `figure.pdf`.

If you like to avoid the graphic interface, this can be done in command line with `fig2dev`.

```
fig2dev -L pstex_t figure.fig figure.pstex_t
fig2dev -L pstex figure.fig figure.pstex
```

If you do not have a fig figure, you can transform a postscript file into a fig one with `pstoedit` with command line like:

```
pstoedit -dis -f fig example.eps > example.fig
```

figfrag With `figfrag` (<http://www.ctan.org/tex-archive/graphics/figfrag/>) you can use the `psfrag` feature in your fig picture and create a standalone eps figure with the `psfrag` replacements.