

The pstool package

Zebb Prime & Will Robertson

vo.2 2008/08/03

Contents

I	Documentation	1
1	Processing modes	1
2	Cropping graphics	2
3	Todo	2
II	Implementation	2
4	Command parsing	5
4.1	User commands	5
5	The figure processing	6

Part I

Documentation

1 Processing modes

Default commands:

`\psfig` Process this figure if no PDF, or new EPS;

`\psfig*` Always process this figure; and maybe,

`\psfig!` Never process this figure?

The following package options override the above: `[process=all]`, `[process=none]` (the default is `[process=auto]`).

2 Cropping graphics

Graphics are cropped to the appropriate size with the preview package. Sometimes, however, this will not be good enough when an inserted label protrudes from the natural bounding box of the figure. A good way to solve this problem is to use the pdfcrop program (requires a Perl installation under Windows). This can be activated in pstool with the [pdfcrop] package option.

3 Todo

1. Higher commands (\psfragfig, \matlabfig, \mathfig)
2. Automatic detection for PDF older than EPS (Windows; shells other than bash?)
3. Generalise "olding" code for multiple files.
4. mylatex integration

Part II

Implementation

```
1 \ProvidesPackage{%  
    pstool}[2008/08/03_v0.2_Wrapper_for_processing_PostScript/psfrag_figures]
```

Initialisations

```
\if@pstool@always@ 2 \newif\if@pstool@always@  
\if@pstool@never@ 3 \newif\if@pstool@never@  
\if@pstool@oldpdf@ 4 \newif\if@pstool@oldpdf@  
\if@pstool@pdfcrop@ 5 \newif\if@pstool@pdfcrop@  
    \pstool@out 6 \newwrite\pstool@out
```

Package options

```
7 \RequirePackage{xkeyval}  
pdfcrop 8 \DeclareOptionX{pdfcrop}{\@pstool@pdfcrop@true}  
  
process 9 \define@choicekey*{pstool.sty}{process}[\@tempa\@tempb]{%  
    all,none,auto}{%  
10 \ifcase\@tempb\relax  
11 \@pstool@always@true  
12 \or  
13 \@pstool@never@true
```

```

14 \or
15 \fi}

```

```

16 \ProcessOptionsX

```

External packages

```

17 \RequirePackage{%
    catchfile,color,ifpdf,ifplatform,inversepath,graphicx,suffix}

```

These are cute:

```

\OnlyIfFileExists 18 \providecommand\OnlyIfFileExists[2]{\IfFileExists{#1}{#2}{}}
\NotIfFileExists 19 \providecommand\NotIfFileExists[2]{\IfFileExists{#1}{#2}{}}

```

Command line abstractions between platforms:

```

20 \edef\pstool@cmdsep{\ifwindows\string&\else\string;\fi\space}
21 \edef\pstool@rm{\ifwindows\del\else\rm--\fi}

```

```

\pstool@try@rm 22 \newcommand\pstool@try@rm[1]{%
23 \for\@tempa:=#1\do{%
24 \OnlyIfFileExists{\@tempa}{\immediate\write18{%
    \pstool@rm"\@tempa"}}}}

```

It is necessary while executing commands on the shell to write the exit status to a temporary file to test for failures in processing.

```

\pstool@statusfile 25 \def\pstool@statusfile{status-deleteme.txt}
\pstool@writestatus 26 \def\pstool@writestatus{%
27 \ifwindows
28 echo\pstool@percentsign\ERRORLEVEL\@percentchar\space>\%
    \pstool@statusfile
29 \else
30 echo$?>\pstool@statusfile
31 \fi}

```

Write more to the file to ensure the buffer is flushed and the file is written to disk properly (allowing it to be read by \CatchFileEdef). (Maybe even a touch would be enough?)

```

\pstool@flushstatus 32 \def\pstool@flushstatus{%
33 \@tempcnta=0
34 \loop
35 \advance\@tempcnta by 1
36 \immediate\write18{%

```

```

37      cd_\ip@directpath"\pstool@cmdsep
38      echo_\@percentchar_buffer_text">>_\pstool@statusfile}%
39      \ifnum\@tempcnta<10\repeat}

```

```

\pstool@retrievestatus 40 \def\pstool@retrievestatus#1{%
41   \pstool@flushstatus
42   \CatchFileEdef{#1}{\ip@directpath\pstool@statusfile}{}%
43   \pstool@try@rm{\ip@directpath\pstool@statusfile}% uses \@tempa
      internally
44 }

```

use `ls` (or `dir`) to detect if the EPS is newer than the PDF:

```

\pstool@datefiles 45 \def\pstool@datefiles{%
46   \edef\pstool@filenames{\ip@lastelement.eps\space_\%
      \ip@lastelement.pdf\space}%
47   \immediate\write18{%
48     \ifwindows
49     \else
50       cd_\ip@directpath"\pstool@cmdsep
51       ls_t_\ip@lastelement.eps"_\ip@lastelement.pdf">_\%
      \pstool@statusfile
52     \fi}%
53   \pstool@retrievestatus\@tempb
54   \ifx\@tempb\pstool@filenames
55     \@pstool@oldpdf@true
56   \else
57     \@pstool@oldpdf@false
58   \fi}

```

Generic function to execute a command on the shell and pass its exit status back into L^AT_EX. Any number of `\pstool@exe` statements can be made consecutively followed by `\pstool@endprocess`, which also takes an argument. If *any* of the shell calls failed, then the execution immediately skips to the end and expands `\pstool@error` instead of the argument to `\pstool@endprocess`.

```

\pstool@exe 59 \def\pstool@exe#1{%
60   \immediate\write18{%
61     cd_\ip@directpath"\pstool@cmdsep
62     #1\pstool@cmdsep
63     \pstool@writestatus}%
64   \pstool@retrievestatus\@tempb
65   \ifnum\@tempb>_0
66     \PackageWarning{pstool}{%
      Execution_failed_during_process:^^J_\#1^^J}%
67   \expandafter\pstool@abort

```

```
68 \fi}
```

Edit this definition to print something else when graphic processing fails.

```
\pstool@error 69 \def\pstool@error#1{\fbox{\color{red}%
\ttfamily\An\error\occured\processing\this\graphic.}}

\pstool@abort 70 \def\pstool@abort#1\pstool@endprocess{\pstool@error}
71 \let\pstool@endprocess\@firstofone
```

4 Command parsing

User input is `\psfig` (with optional `*` or `!` suffix) which turns into one of the following three macros depending on the mode.

```
\pstool@alwaysprocess 72 \newcommand\pstool@alwaysprocess[3] [] {%
73 \inversepath*{#2}% calculate filename, path & inverse path
74 \pstool@process[#1]{#2}{#3}}
```

```
\pstool@neverprocess 75 \newcommand\pstool@neverprocess[3] [] {%
76 \includegraphics[#1]{#2}}
```

For regular operation, which processes the figure only if the command is starred, or the PDF doesn't exist.

```
\pstool@maybeprocess 77 \newcommand\pstool@maybeprocess[3] [] {%
78 \inversepath*{#2}% calculate filename, path & inverse path
79 \IfFileExists{#2.pdf}{%
80 \pstool@datefiles
81 \if@pstool@oldpdf@{\expandafter\@firstoftwo
82 \else\expandafter\@secondoftwo\fi{%
83 \pstool@process[#1]{#2}{#3}%
84 }{%
85 \includegraphics[#1]{#2}}%
86 }{%
87 \pstool@process[#1]{#2}{#3}%
88 }}
```

4.1 User commands

Finally, define `\psfig` as appropriate for the mode:

```
89 \ifpdf
90 \if@pstool@always@
91 \let\psfig\pstool@alwaysprocess
```

```

\psfig 92 \WithSuffix\def\psfig!{\pstool@alwaysprocess}
\psfig* 93 \WithSuffix\def\psfig*{\pstool@alwaysprocess}
94 \else\if@pstool@never@
95 \let\psfig\pstool@neverprocess
\psfig 96 \WithSuffix\def\psfig!{\pstool@neverprocess}
\psfig* 97 \WithSuffix\def\psfig*{\pstool@neverprocess}
98 \else
99 \let\psfig\pstool@maybeprocess
\psfig 100 \WithSuffix\def\psfig!{\pstool@neverprocess}
\psfig* 101 \WithSuffix\def\psfig*{\pstool@alwaysprocess}
102 \fi\fi
103 \else
104 \let\psfig\pstool@neverprocess
\psfig 105 \WithSuffix\def\psfig!{\pstool@neverprocess}
\psfig* 106 \WithSuffix\def\psfig*{\pstool@neverprocess}
107 \fi

```

5 The figure processing

```

\pstool@process 108 \newcommand{\pstool@process}[3][{}]{%
109 \pstool@write@processfile{#1}{#2}{#3}%
110 \pstool@exe{latex_\shell-escape_\output-format=dvi
111 -interaction=batchmode_\ip@lastelement-process.tex"}%
112 \pstool@exe{dvips_\ip@lastelement-process.dvi}%
113 \if@pstool@pdfcrop@
114 \pstool@exe{ps2pdf_\ip@lastelement-process.ps}_"%
\ip@lastelement-process.pdf"%
115 \pstool@exe{pdfcrop_\ip@lastelement-process.pdf}_"%
\ip@lastelement.pdf"%
116 \else
117 \pstool@exe{ps2pdf_\ip@lastelement-process.ps}_"%
\ip@lastelement.pdf"%
118 \fi
119 \pstool@endprocess{\includegraphics[#1]{#2}}

```

The file that is written for processing is set up to read the preamble of the original document and set the graphic on an empty page (cropping to size is done either here with preview or later with pdfcrop).

```

\pstool@write@processfile 120 \def\pstool@write@processfile#1#2#3{%
121 \immediate\openout\pstool@out_\#2-process.tex\relax
122 \immediate\write\pstool@out{%
123 \unexpanded{%
124 \pdfoutput=0% force DVI mode if not already

```

Input the main document; redefine the document environment so only the preamble is read:

```

125         \let\origdocument\document
\document 126         \def\document{\endgroup\endinput}%
127     }\noexpand
128         \input{\ip@inversepath\jobname}%

```

Now the preamble of the process file: (restoring document's original meaning)

```

129     \if@pstool@pdfcrop@\else
130         \noexpand\usepackage[active,tightpage]{preview}
131     \fi
132     \unexpanded{%
133         \let\document\origdocument
134         \pagestyle{empty}% remove the page number
135         \begin{document}

```

And the document body to place the graphic on a page of its own:

```

136         \centering\null\vfill}%
137     \if@pstool@pdfcrop@\else
138         \noexpand\begin{preview}%
139     \fi
140         \unexpanded{#3}% this is the "psfrag" material
141         \noexpand\includegraphics[#1]{\ip@lastelement}%
142     \if@pstool@pdfcrop@\else
143         \noexpand\end{preview}%
144     \fi
145     \unexpanded{%
146         \vfill\end{document}}%
147     }%
148 \immediate\closeout\pstool@out}

```

<eof>