PSTricks

pst2pdf

Running a PSTricks document with pdflatex and pst-exa;
v.0.02

January 19, 2011

Package author(s):
Herbert Voß

Contents

Table 1: Possible optional arguments for the Perl script `pst2pdf`

name	*values*	*default*	*description*
`-imageDir`	literal	`imgages/`	the directory for the created images
`-Iext`	literal	`.pdf`	the extension for `\includegraphics`, can be empty, then `\includegraphics` decides which image is used.
`-DPI`	integer	`75`	the dots per inch for a created png file, if possible
`-Iscale`	real	`1`	the value for the option `scale` in `\includegraphics`. Important when using a greater dpi value.
`-tempDir`	literal	`.`	the temporary directory for the temp files
`-verbose`	boolean	`1`	for a long `pst2pdf` log
`-clear`	boolean	`0`	delete all temporary files
`-noImages`	boolean	`0`	create no images, build only the pdf with the alread existing images
`-runBibTeX`	boolean	`0`	runs `bibtex`
`-runBiber`	boolean	`0`	runs `biber` if a file with extension `.bcf` exists

`pst2pdf` is a Perl script for running a PSTricks document in a last run with pdflatex. `pst-exa` is a package that supports the printing of code and output of PSTricks examples when running in pdf mode.

Thanks to:
Pablo Gonzales Luengo; Rolf Niepraschk

1 Introduction

`PSTricks` as `PostScript` -related package uses the programming language `PostScript` for internal calculations. This is an important adavantage, because floating point arithmetic is no problem. Nearly all mathematical calculation can be done when running the DVI-file with Ghostscript. However, creating a PDFfile in a direct way with `pdflatex` is not possible. `pdflatex` cannot understand the `PostScript` related stuff. Instead of running `pdflatex` one can use the Perl script `pdf2eps`, it extracts all `PSTricks` -related code into single documents with the same preamble as the original main document. Then the script runs this document, clips all whitespace arounf the image and creates a `.pdf`, `.eps`, and `.png` image of the `PSTricks` related code. In a last run which is the `pdflatex` the `PSTricks` code in the main dcouemnt is replaced by the created images.

Table 2: Possible optional arguments for the package `pst-exa`

name	*values*	*default*	*description*
`pos`	`l,r,b,t`	`l`	position of the image, maybe left, right, bottom ot top of the code.
`halign`	`l,r,c`	`c`	the horizontal alignment of the image.
`valign`	`l,r,c`	`c`	the vertical alignment of the image.
`frame`	see lst		option is passed to `\lstinputlisting` from the package `listings`.
`width`	length	`0.5\linewidth`	the width of the example box.
`sep`	length	`1em`	separation between image and code.
`imageDir`	literal	`images/`	directory for the created images and tex files.

2 Running the Perl script

The genral syntax for the Perl script is simple

`pst2pdf` *file* `.tex` `options`

The options listed in Table 1 refer only to the script and not the LaTeX file.

After the `pst2pdf` run there exists a pdf file called `\jobname-pdf.pdf`. And when not using the `-clear` option also the corresponding TeX file `\jobname-pdf.tex`. The preamble of the document should contain all code which is important to the `PSTricks` code.

3 `PSTricks` code

The per scripts scans the files for `pspicture` and `postscript` environments, which are then taken with its contents from the main file to create stand alone documents with the same preamble as the main document. The `pspicture` environment can be nested, the `postscript` one not! But it can contain an environment `pspicture`, but not vice versa. The `postscript` environment should always be used, when there is some code before a `pspicture` environment or for some code which is not inside of a `pspicture` environment.

4 The package `pst-exa`

The package `pst-exa` was created to realize examples with printed code and output side by side or on top of each other. The package looks in the image directory for the source code of the examples and inserts only the code between the environment `document`, which is the sequence `\begin{document} ... \end{document}`.

The package provides the environment `PSTexample` with the optional arguments listed in Table 2.

5 Examples

The package contains some example files for uning the script without and with the package `pst-exa`.

test1.tex running `pst2pdf test1`. The test file contains a jpg-image, which is only possible with pdflatex.

test2.tex same as `test1`, but with using `pst-exa` and example–code combination.

test3.tex another example

6 List of all optional arguments for `pst-exa`

Key	Type	Default
`pos`	`ordinary`	`l`
`halign`	`ordinary`	`c`
`valign`	`ordinary`	`c`
`frame`	`ordinary`	
`width`	`ordinary`	`0.5\linewidth`
`vsep`	`ordinary`	`1em`
`sep`	`ordinary`	`1em`
`imageDir`	`ordinary`	`imgages/`

References

[1] Denis Girou. Présentation de PSTricks. *Cahier GUTenberg*, 16:21–70, April 1994.

[2] Michel Goosens, Frank Mittelbach, Sebastian Rahtz, Denis Roegel, and Herbert Voß. *The LaTeX Graphics Companion*. Addison-Wesley Publishing Company, Reading, Mass., 2007.

[3] Laura E. Jackson and Herbert Voß. Die Plot-Funktionen von `pst-plot`. *Die TeXnische Komödie*, 2/02:27–34, June 2002.

[4] Nikolai G. Kollock. *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*. IWT, Vaterstetten, 1989.

[5] Herbert Voß. Die mathematischen Funktionen von PostScript. *Die TeXnische Komödie*, 1/02, March 2002.

[6] Herbert Voß. *PSTricks – Grafik für TeX und LaTeX*. DANTE – Lehmanns, Heidelberg/Hamburg, 5. edition, 2008.

[7] Timothy van Zandt. *PSTricks - PostScript macros for generic TeX*. http://www.tug.org/application/PSTricks, 1993.

[8] Timothy van Zandt. *multido.tex - a loop macro, that supports fixed-point addition*. CTAN:/graphics/pstricks/generic/multido.tex, 1997.

[9] Timothy van Zandt. *pst-plot: Plotting two dimensional functions and data*. CTAN:graphics/pstricks/generic/pst-plot.tex, 1999.

[10] Timothy van Zandt and Denis Girou. Inside PSTricks. *TUGboat*, 15:239–246, September 1994.

Index