The pdftricks2 package

Christian Reibiger
inspired by the pdftricks package

June 7, 2013

Contents

1 Introduction

This package provides automatization facilities to use the pstricks package with *pdflatex*. It is intended to be of easy use and interoperability with *latex* and *pdflatex*, namely both show the same result. To use this package you only have to say `\usepackage{pdftricks2}` instead of `\usepackage{pstricks}`.

The pdftricks2 package circumvents the problem that *pdflatex* can not handle *ps*-code by writing an extra *latex* file that contains only the figure. This files are converted to dvi to ps and then to pdf. Finally the resulting pdf-pictures are included in the text as a graphic. This process can be handled automatically by the package but it needs to have the possibility to execute system commands (start *pdflatex* with option `--enable-write18` or `-shell-escape`). If you do not want to grant this rights you still can use the package but you have to execute a shell script the package writes yourself to produce the pdf-pictures. In this case you have to run *pdftex* again to include the pictures in your document.

The package tries to show the same results when used with *latex* as for *psflatex*. However there are some side effects of using this package as described below. Lightnings (↯) on the margin of the page mark parts where changes to pstricks syntax or important flaws are stated.

Although this package is inspired by the pdftricks package, it is coded from scratch. The pdftricks package has some flaws that are circumvented by pdftricks2. For example you have to gather all configurations you need for your pictures in one place. Furthermore it is not possible to share configuration code between your figures and the main text. Also the pdftricks package does not use the provided limits of the figure you produce but rather assumes it fits on a standard page and cuts everything away that does not look like being important.

2 Installation

Just copy the file ‘pdftricks2.sty’ to your tex directory (for example ‘~/texmf/tex/latex/pdftricks2’ on a linux or ‘C:\Programs\MikTeX\tex\latex\pdftricks2’ on a windows system). After that you have to update LaTeX's hash table by invoking 'texhash' on a linux system or by executing 'Start ≫ Programs ≫ MikTeX ≫ Maintenance ≫ Settings, General ≫ Refresh FNDB' on a windows system.

3 Usage

`\usepackage{pdftricks2}`

To use this package you have to load it via `\usepackage{pdftricks2}`. The package loads automatically the pstricks package, so there is no need to say `\usepackage{pstricks}`. All options you pass to pdftricks2 that are not recognized are passed to pstricks. So you can give the pstricks options just to the pdftricks2 package.

`\usepackageps, \usepackagepsonly`

After loading the pdftricks2 package you can load a *package* you want to use for your text and your figures by saying `\usepackageps{`*package*`}`. Also you have the possibility to load ↯ packages only for the figures and not in your other text by saying `\usepackagepsonly{`*package*`}` but this may have side effects if you use pdftricks2 with *latex* because they get included in the text as well.

`\psset, \newpsstyle, \addtopsstyle, \degrees, \radians, \SpecialCoor`

Also you can use the pstricks commands that change the configuration of drawing outside of ↯ a `pspicture`-environment. But you must be aware that their effect is always global. Inside of `pspicture`-environments the commands stated above obey the usual grouping rules.

`\configps, configpsblock`

In addition to this commands there are new ones to share definitions between your text and the figures. They are intended to be used for custom commands you want to use in text and in figures. The command `\configps{`*code*`}` executes the given *code* in the text and in every figure. The same is true for the *code* in `\begin{configpsblock}`*code*`\end{configpsblock}` but this second version is better for a large amount of code.

`\configpsonly, configpsonlyblock`

As in case of package loading these commands provide the possibility to include some *code* only in the figures. They are used the same way as `\configps` and `configpsblock`

`pspicture`

A `pspicture`-environment can be started as usual by saying `\begin{pspicture}`. This is fine if you keep your drawing inside the picture. However, you should say `\begin{pspicture*}` to clip your drawing to the box of the `pspicture`. Keep in mind that *pdflatex* will clip your drawing anyway. Therefore the use of the `pspicture` is discouraged and pdftricks2 ↯ will warn you when you use it. For technical reasons it is not possible to continue your LaTeX code right after `\end{pspicture*}`, instead you have to end the line there. If you do not want to have a space after your `pspicture`-environment you can end the line with ↯ a comment. In contrast to the current pstricks implementation you can not use the `shift` attribute to produce a `pspicture` that is not aligned by its lower border. Instead you have to use the old pstricks syntax `\begin{pspicture*}[`*a*`]` to produce a `pspicture` that's baseline is at the a-th part of its vertical size. Therefore it is not possible to slip configuration commands for pstricks in the optional argument of the `pspicture`-environment.

4 Package options

In the following the options that are processed by the pdftricks2 package are stated. All other options you provide are passed to the pstricks package.

`info`
: When this option is set pdftricks2 prints out additional information via package warnings. Especially it shows what files it will use and when it opens and closes them.

`shell, noshell`
: This options control whether the package tries to execute system commands for translating the figures itself or not. The default is to use the shell if possible. When no shell is used pdftricks2 writes a script that handles the translation of the figures. You just have to execute it as will be described in a package warning.

`windows, unix`
: For some system commands it is necessary to distinguish between windows and unix systems. The package tries to guess the system you have by using the ifplatform package. However, you can disable the automatic detection by giving the platform explicitly. This is mainly intended to be used if problems occur.

`figpath=`*`path`*
: Normally pdftricks2 creates the additional files it needs in the same directory as the one you run *pdflatex* from. This may be a bit messy. Therefore you can give an alternate *path* where the temporary files are stored. But this *path* has to exist. The path you can choose has to be given in unix style (it is converted to windows style internally if needed) and must meet the requirements for paths imposed by LaTeX.

`scriptpaths, noscriptpaths`
: This options enable you to control whether pdftricks2 writes the *path* given via `figpath` to the script and figure files or whether it omits them when writing these files. By default the paths are written to the files which is necessary if the translation is handled by the package itself. However, if you translate them by hand it might be useful to omit the paths.

5 Changelog

5.1 Version 1.01

Fixed the commands `\usepackageeps` and `\usepackagepsonly`.