

LTXIMG v1.1

TIKZ|PGF|PSTRICKS
TO IMAGE FORMAT

Pablo González Luengo
pablgonz at yahoo dot com

April 21, 2015

Abstract

ltximg is a Perl script that automates the process to extract and convert all PGF|TiKZ|Pstricks environments from input file to image formats (pdf,png,jpg,eps,ppm,svg) and source code for environments in individual files using Ghostscript and other software. By default search and extract environments using pdf_lTeX.

Contents

1	Required Software	1
2	Run and options	2
3	How it works	2
3.1	Environment Supports	2
3.2	Comment and ignore	3
3.3	Create source and images	3
4	Example	3

1 Required Software

For the full operation of the ltximg script you need the following opensource programs (available for windows and linux).

- Perl (version 5.18 or higer).
- Ghostscript (version 9.16).
- pdftops (optional, for images in EPS format).
- pdftoppm (optional, for images in PPM format).
- pdf2svg (optional, for images in SVG format).

2 Run and options

the syntax for ltximg script is simple: For T_EXLive users:

```
ltximg file.tex --options
```

For MikT_EX users:

```
perl ltximg file.tex --miktex --options
```

Table 1: Options for ltximg

short	long	default	<i>description</i>
-h	--help		Display help from command line and exit.
-l	--license		Display license information and exit.
-v	--version	1.1	Display version of script and exit.
-d	--dpi=<int>	150	The dots per inch for images.
-j	--jpg		Create .jpg files (need Ghostscript).
-p	--png		Create .pngfiles (need Ghostscript).
-e	--eps		Create .eps files (need pdftops).
-s	--svg		Create .svg files (need pdf2svg).
-P	--ppm		Create .ppm files (need pdftoppm).
-a	--all		Create (.pdf,eps,jpg,png) images.
-c	--clear		Delete all temp and aux files.
-o	--output		Create a file-out.tex whitout PGF TiKZ PST code.
-m	--margins=<int>	0	Margins in bp for pdfcrop.
-np	--nopreview	off	Create images files whitout preview package.
-ns	--nosource	off	Delete all source for images files
	--miktex	-shell-escape	Use -enable-write18 for MiK _T E _X users.
	--xetex	off	Using X _E L _A T _E X for create images.
	--latex	off	Using L _A T _E X for create images.
	--luatex	off	Using Lua _L A _T E _X for create images.
	--nopdf	off	Don't create a PDF image files.
	--other=<string>	other	Search other environment for export.
	--ignore=<string>	ignore	Skip verbatim environment.
	--imgdir=<string>	images	The folder for images.

3 How it works

The script works in two steps for create image, source code and output file *without PGF/TIKZ/PST* environments.

3.1 Environment Supports

ltximg export and convert this environment:

<code>\begin{pspicture}</code>	<code>\begin{pspicture*}</code>	<code>\begin{postscript}</code>	<code>\begin{circuitikz}</code>
<code><environment content></code>	<code><environment content></code>	<code><environment content></code>	<code><environment content></code>
<code>\end{pspicture}</code>	<code>\end{pspicture*}</code>	<code>\end{postscript}</code>	<code>\end{circuitikz}</code>
<code>\begin{tikzpicture}</code>	<code>\begin{pgfpicture}</code>	<code>\begin{gantchart}</code>	<code>\begin{forest}</code>
<code><environment content></code>	<code><environment content></code>	<code><environment content></code>	<code><environment content></code>
<code>\end{tikzpicture}</code>	<code>\end{pgfpicture}</code>	<code>\end{gantchart}</code>	<code>\end{forest}</code>

<code>\begin{tikzcd}</code>	<code>\begin{circuitikz}</code>	<code>\begin{dependency}</code>	<code>\begin{other}</code>
<code>\langle environment content \rangle</code>	<code>\langle environment content \rangle</code>	<code>\langle environment content \rangle</code>	<code>\langle environment content \rangle</code>
<code>\end{tikzcd}</code>	<code>\end{circuitikz}</code>	<code>\end{dependency}</code>	<code>\end{other}</code>

3.2 Comment and ignore

The first step `ltximg` script create a image dir (`images/`) and read all input file in memory, processing is as follows, being assumed that our file is `test.tex`:

1. Create a copy file in memory and change problematic environments (`verbatim`, `verbatim~`, `lstlisting`, `LTXexample`, `Verbatim`, `comment`, `alltt`, `minted`, `tcblisting`, `xcomment` and `ignore`)
2. Change problematic inline `verbatim` from various package (`lstlisting`, `LTXexample`, `Verbatim`, `comment`, `alltt`, `minted`, `tcblisting`, etc) include line `whit %`.

3.3 Create source and images

1. If script is call `whitout -nopreview` option (default), adds the following lines to the beginning of the `test.tex` in memory and save file `test-fig.tex` in `images` dir :

```
\AtBeginDocument{
\RequirePackage[active,tightpage]{preview}
\renewcommand\PreviewBbAdjust{-60pt -60pt 60pt 60pt}%
\newenvironment{postscript}{}{}%
\PreviewEnvironment{postscript}%
```

2. If script is call `whit -nopreview` option, all environment code its put inside the :

```
\begin{postscript}
...
\end{postscript}
```

and separate in individual files (`test-fig-1.tex`, `test-fig-2.tex`, etc) in `image` dir and join in `test-fig.tex`. The `postscript` environment can be used to place any material to be exported if that is not supported or fails the `-other` option.

3. Now, the script call `(pdf/luaxe)latex` in `test-fig.tex` and `pdftocrop` in `test-fig.pdf` and create image files.

4 Example

```
ltximg test.tex -e -p -j -c -o --imgdir=pics
```

produce `test-out.tex` whitout `PGF|TikZ|PST` environments and create "pics" dir whit all images (`pdf,eps,png,jpg`) and source (`.tex`) for all related parts using `pdfLATEX` whit `preview` package and cleaning all `tmp` files.

Support bundling for short options:

```
ltximg test.tex -epjco --imgdir=pics
```