The cachepic package

Tomasz M. Trzeciak
t34www@gmail.com

September 23, 2009

1 Introduction

It might be sometimes desirable to convert fragments of your document, e.g. drawings or diagrams, into external graphic files. Such a need may arise when you want to use some specialised package but your document have to compile on a system without said package.

The purpose of cachepic package is to simplify and automate conversion of document fragments into external eps or pdf file(s). The package consists of two parts: the LaTeX 2_ε style implementing the document level interface and a command line tool (written in Lua) for generation of external graphics.

2 Requirements

The inclusion of already generated graphics requires only the `cachepic.sty` style file and two standard packages: graphicx (obviously) and verbatim (for its comment environment). Both should be present in every LaTeX distribution.

The graphics generation step requires the full cachepic package, the preview package and a Lua interpreter to execute the conversion script. For Windows platform there is also a batch script wrapper provided for command line use.

3 Document interface

The package is loaded with the usual `\usepackage{cachepic}`. Currently there are no package specific options available. There are two commands and one environment provided for marking-up document fragments intended for externalization as graphic files.

`\cachepic` The main command of this package is `\cachepic{`⟨*file name*⟩`}{`⟨*fragment*⟩`}`. It takes a balanced LaTeX code ⟨*fragment*⟩, which is typeset and output to a file ⟨*file name*⟩ during the post-processing run as described in Section 4.

Table 1: Options of `cachepic` command line tool.

option	description
pdf	graphics will be output in (E)PDF format; this is the default option
eps	graphics will be output in EPS format
all	recreate all graphics, already existing ones will be overwritten
multi	keep all graphics in a sigle file ⟨*main file*⟩`-cachepic.pdf` rather then produce separate files (but graphics in separate files take precedence); this option cannot be used together with `-eps`
tight	no 0.5 bp margin around the graphic
notex	don't do any typesetting, only graphic postprocessing; the typesetting has to be done separately and requires the preview package with at least the `active` and `cachepic` options
nopic	generate no graphics, only ⟨*main file*⟩`.cachepic` file
help,h,?	display help message

`\cacheinput` The second command, `\cacheinput{`⟨*file name*⟩`}`, is just a shorthand for `\cachepic{`⟨*file name*⟩`}{\input{`⟨*file name*⟩`}}`.

`cachepicture` Finally, for longer stretches of code there is an environment analogue of the `\cachepic` command:

```
\begin{cachepicture}{⟨file name⟩}
  ⟨fragment⟩
\end{cachepicture}
```

During normal document compilation each ⟨*fragment*⟩ is typeset as usual unless there exists a file ⟨*file name*⟩`.eps` (in dvi mode) or ⟨*file name*⟩`.pdf` (in pdf mode). If such a file exists, it will be included (with the standard `\includegraphics` command) and it will be used in place of the corresponding ⟨*fragment*⟩.

4 Graphics generation

Graphics files are produced by running a command line tool called (unsurprisingly) `cachepic`. This tool automates the whole process of graphic generation. It can be called from the command line (or through appropriately configured TEX editor) as follows:

`cachepic [options]` ⟨*main file*⟩

All available options are gathered in Table 1. Options start with "`-`" or "`--`" and can be passed in any order.

Apart from graphic files there is also an auxiliary ⟨*main file*⟩`.cachepic` file created. It contains additional information used for graphics inclusion such as the page number (used only in PDF mode with multi-graphics file), margins and the depth of the graphics box. This file is not strictly required but without it some features are not available as explained below.

As mentioned in Section 2, the preview package is required for graphics generation but it should not be specified with options in the document preamble or this will likely lead to option clash. To use this package together with cachepic, pass `cachepic` option to the preview package and compile the document separately. Then run the `cachepic` tool with `-notex` option.

Once all the graphics are generated, only the style file `cachepic.sty` is needed to compile the document. If you want to avoid using even this file, you can add the following definitions the document preamble:

```
\newcommand\cachepic[2]{\includegraphics{#1}}
\newcommand\cacheinput[1]{\includegraphics{#1}}
\newenvironment{cachepicture}[1]{%
\includegraphics{#1}\comment}{\endcomment}
```

However, since the above definitions don't use the additional information in the ⟨*main file*⟩`.cachepic` file, this comes with some limitations. Firstly, information about the graphic's depth, i.e. how much it is lowered below the text baseline, is not preserved. Secondly, graphics stored in a single PDF file cannot be used, because the page number with the graphic is not known. Finally, the default small margin around the graphics is not corrected for, but you can generate the graphics without the margin (see the `-tight` option in Table 1) or use `\includegraphics[trim=0.5bp 0.5bp 0.5bp 0.5bp]{#1}` in the above definitions to correct for that.