

docshots: L^AT_EX Package that Renders T_EX Samples Next to Their PDF Snapshots*

Yegor Bugayenko
yegor256@gmail.com

2022-10-25, 0.0.5


NB! This package doesn't work on Windows! Also, you must run T_EX processor with `--shell-escape` option. Also, you must have `pdlafter`, [Perl](#), [Ghostscript](#), and [pdfcrop](#) installed.

1 Introduction

When you want to demonstrate to the readers of your documentation how to use certain T_EX commands, the best way would be to show exactly how the entire document will be rendered in PDF, using a subprocess that would render it (via `pdflatex`, for example). To [my best](#) knowledge, there were no packages that would allow you do exactly this. That's why I created this simple package. For example, this code:

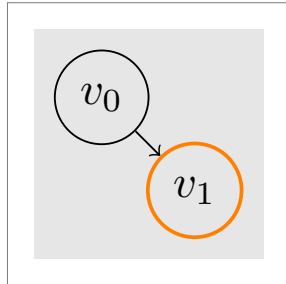
```
\begin{docshot}
\documentclass{article}
\usepackage{xcolor}
\pagestyle{empty}
\begin{document}
  Hello, {\color{orange}\LaTeX}!
\end{document}
\end{docshot}
```

is rendered as such:

	<pre>1 \documentclass{article} 2 \usepackage{xcolor} 3 \pagestyle{empty} 4 \begin{document} 5 Hello, {\color{orange}\LaTeX}! 6 \end{document}</pre>
---	---

Here is a more complex example:

*The sources are in GitHub at [yegor256/docshots](https://github.com/yegor256/docshots)



```

1 \documentclass{article}
2 \usepackage{tikz}
3 \usetikzlibrary{backgrounds}
4 \pagestyle{empty}
5 \begin{document}
6 \begin{tikzpicture}[
7   show background rectangle,
8   background rectangle/.style=
9     {draw=none,fill=gray!20}]
10 \node [circle,draw] (v0) {$v_0$};
11 \node [circle,draw=orange,thick,
12   below right of=v0] (v1) {$v_1$};
13 \draw [->] (v0) -- (v1);
14 \end{tikzpicture}
15 \end{document}

```

The picture you see on the left side is rendered by a subprocess executing `pdflatex` with the `.tex` content taken from the source file. After a successful processing of \TeX sources, we use [pdfcrop](#) to trim the document.

We execute `pdflatex` with `-interaction=batchmode` and `-halt-on-error` options. This means that \TeX processing will stop at the first error. Check your \TeX log to understand what went wrong.

When you render a text instead of a single picture, it's recommended to use [geometry](#) package to change the size of the page and then empty page style to remove page numbering:

“There is no sadder thing than a young pessimist, except an old pessimist” — *Mark Twain*

```

1 \documentclass{article}
2 \usepackage[paperwidth=2in]{geometry}
3 \pagestyle{empty}
4 \begin{document}
5 ‘‘There is no sadder thing than
6 a young pessimist, except an old
7 pessimist’’ --- \emph{Mark Twain}
8 \end{document}

```

2 Package Options

`pdflatex` The default command line tool for turning `.tex` into `.pdf` is `pdflatex`. However, you can change that by using `pdflatex` package option, for example:

```

\documentclass{article}
\usepackage[pdflatex=/usr/local/bin/pdflatex]{docshot}
\begin{document}
\begin{docshot}
Hello, world!
\end{docshot}
\end{document}

```

`gs` The default location of Ghostscript is just `gs`. You can change that by using `gs` package

option, for example:

```
\usepackage[gs=/usr/bin/ghostscript]{docshot}
```

pdfcrop The default location of pdfcrop is just pdfcrop. You can change that by using pdfcrop package option, for example:

```
\usepackage[pdfcrop=/bin/pdfcrop]{docshot}
```

margin When we crop the PDF rendered, we leave a margin around the content. The default value may be changed by the package option margin:

```
\usepackage[margin=10]{docshot}
```

hspace The horizontal distance between the image and its verbatim T_EX source may be configured via hspace package option:

```
\usepackage[hspace=1em]{docshot}
```

left The default width of the image may be changed by left option, while the width of
right the verbatim T_EX source may be modified by right option:

```
\usepackage[left=2in,right=.5\linewidth]{docshot}
```

dtx If you use this package inside .dtx documentation, add dtx package option. Thanks to this option all comment symbols will be removed from line starts:

```
\usepackage[dtx]{docshot}
```

tmpdir The default location of temp files is \docshots@tmpdir. You can change this using tmpdir option:

```
\usepackage[tmpdir=/tmp/foo]{docshot}
```

runs By default, we run pdf_lat_ex just once for each docshot. You can change this number using the package option runs. This may be helpful if you need BibT_EX processing, for example:

```
\usepackage[runs=3]{docshot}
```

small You don't have too much freedom in formatting of verbatim texts, but you can make
tiny their font a bit smaller using small package option. You can also make it very small using tiny option:

```
\usepackage[small]{docshot}
```

log With log option you can make us print all possible logs to the main T_EX log. By default, we don't do this and you won't see the output of pdf_lat_ex compilation, for example. Just use it like this:

```
\usepackage[log]{docshot}
```

inputminted By default, we render the verbatim text using \VerbatimInput command. You can change that and make us use \inputminted from [minted](#) package instead, for example:

```
\usepackage{minted}  
\setminted[java]{frame=lines,framesep=2mm}  
\usepackage[inputminted=java]{docshot}
```

lstinputlisting By default, we render the verbatim text using \VerbatimInput command. You can change that and make us use \lstinputlisting from [listings](#) package instead, for example:

```
\usepackage{listings}  
\lstset{basicstyle=\small}  
\usepackage[lstinputlisting]{docshot}
```

3 Prerequisites

`\docshotPrerequisite` If you need some files to be present next to the `.tex` snippet while it's rendered by `pdflatex`, you can use `\docshotPrerequisite` with a single mandatory argument. The argument is the name of a file you need to be copied from current directory to the temporary directory, where all snippets are rendered. The command can be used either in the body of the document or in the preamble — it doesn't matter, as long as it shows up before the docshot that needs the prerequisite. For example:

```
\documentclass{article}
\usepackage{docshot}
\docshotPrerequisite{duck.jpg}
\begin{document}
\begin{docshot}
  \documentclass{article}
  \usepackage{graphicx}
  \pagestyle{empty}
  \begin{document}
    This is my favorite picture of a duck:
    \includegraphics[width=2in]{duck.jpg}
  \end{document}
\end{docshot}
\end{document}
```

`\docshotAfter` If you need something to happen after each `pdflatex` run of a docshot, you may use `\docshotAfter` command right before docshot environment. For example, you have a bibliography file that you want to be visible for all snippets and you want all of them to run `biber` in order to process citations:

```
\documentclass{article}
\usepackage{docshot}
\docshotPrerequisite{main.bib}
\begin{document}
\docshotAfter{biber $2}
\begin{docshot}
  \documentclass{acmart}
  \usepackage[natbib=true]{biblatex}
  \addbibresource{main.bib}
  \pagestyle{empty}
  \begin{document}
    You must read the book of \citet{knuth1984}.
    \printbibliography
  \end{document}
\end{docshot}
\end{document}
```

The script you specify in the first argument of `\docshotAfter` will get three arguments when it runs:

\$1 the cycle of `pdflatex` processing (1, 2, ...),

\$2 the hash of the snippet,

\$3 the name of the `.tex` file.

\$2 is basically equals to \$1 with an attached `.tex` suffix. `\docshotAfter` applies only to the first docshot environment that goes after it! You must specify `\docshotAfter` before each docshot where you want such post-processing to happen.

4 Implementation

First, we include a few packages:

```
1 \RequirePackage{iexec}
2 \RequirePackage{fancyvrb}
3 \RequirePackage{xcolor}
4 \RequirePackage{graphicx}
5 \RequirePackage{tikz}
```

Then, we process package options:

```
6 \RequirePackage{pgfopts}
7 \def\docshots@log{}
8 \pgfkeys{
9   /docshots/.cd,
10  dtx/.store in=\docshots@dtx,
11  log/.code=\def\docshots@log{log},
12  lstinputlisting/.store in=\docshots@lstinputlisting,
13  inputminted/.store in=\docshots@inputminted,
14  tmpdir/.store in=\docshots@tmpdir,
15  tmpdir/.default=_docshots,
16  small/.store in=\docshots@small,
17  tiny/.store in=\docshots@tiny,
18  runs/.store in=\docshots@runs,
19  runs/.default=1,
20  pdflatex/.store in=\docshots@pdflatex,
21  pdflatex/.default=pdflatex,
22  gs/.store in=\docshots@gs,
23  gs/.default=gs,
24  pdfcrop/.store in=\docshots@pdfcrop,
25  pdfcrop/.default=pdfcrop,
26  margin/.store in=\docshots@margin,
27  margin/.default=5,
28  hspace/.store in=\docshots@hspace,
29  hspace/.default=.05\linewidth,
30  left/.store in=\docshots@left,
31  left/.default=.3\linewidth,
32  right/.store in=\docshots@right,
33  right/.default=.55\linewidth,
34  tmpdir,pdflatex,gs,pdfcrop,margin,hspace,left,right,runs
35 }
36 \ProcessPgfOptions{/docshots}
```

Then, we print the version of pdf \LaTeX to \LaTeX log:

```
37 \iexec[\docshots@log,quiet]{\docshots@pdflatex\space --version}%
```

Then, we print the version of pdfcrop to \LaTeX log:

```
38 \iexec[\docshots@log,quiet]{\docshots@pdfcrop\space --version}%
```

Then, we print the version of ghostscript to \LaTeX log:

```
39 \iexec[\docshots@log,quiet]{\docshots@gs\space --version}%
```

Then, we make a directory where all temporary files will be kept:

```
40 \iexec[null]{mkdir -p \docshots@tmpdir/\jobname}%
```

docshot Then, we define docshot environment:

```
41 \newenvironment{docshot}
```

```

42 {\VerbatimEnvironment\begin{VerbatimOut}
43   {\docshots@tmpdir/\jobname/verbatim.tex}}
44 {\end{VerbatimOut}}%

```

If we are in dtx mode, leading percent characters must be removed:

```

45   \ifdefined\docshots@dtx%
46     \iexec[null]{perl -i -0777pe "s/(\n|^)\x{25} /\n/g"
47       \docshots@tmpdir/\jobname/verbatim.tex}%
48   \fi%

```

We calculate MD5 hashsum of the file content:

```

49   \def\hash{\pdfmdfivesum file
50     {\docshots@tmpdir/\jobname/verbatim.tex}}%

```

If the PDF with the required name already exists, we ignore this step. Otherwise, we copy verbatim.tex into new file and run pdflatex:

```

51   \IfFileExists{\docshots@tmpdir/\jobname/\hash.pdf}
52     {\message{docshots: won't render, the PDF already exists
53       (\docshots@tmpdir/\jobname/\hash.pdf)^~J}}
54     {\iexec[\docshots@log,quiet]{cp \docshots@tmpdir/\jobname/verbatim.tex
55       \docshots@tmpdir/\jobname/\hash.tex}%
56     \foreach \n in {1,...,\docshots@runs}{
57       \iexec[\docshots@log,quiet]{cd \docshots@tmpdir/\jobname;
58         \docshots@pdflatex\space
59         -interaction=errorstopmode
60         -halt-on-error
61         -shell-escape
62         \hash.tex}
63       \message{docshots: pdflatex run no.\n^~J}
64       \IfFileExists{\docshots@tmpdir/\jobname/after.sh}
65         {\iexec[\docshots@log,quiet]{chmod a+x
66           \docshots@tmpdir/\jobname/after.sh}
67         \iexec[\docshots@log,quiet]{cd \docshots@tmpdir/\jobname;
68           ./after.sh \n\space \hash\space \hash.tex}}
69     {}}%

```

Here we delete after.sh which may exist here after the last compilation of pdflatex:

```

70   \iexec[\docshots@log,quiet]{rm -f \docshots@tmpdir/\jobname/after.sh}

```

If a cropped version of the PDF with the required name already exists, we ignore this step. Otherwise, we ask pdfcrop to crop the PDF:

```

71   \IfFileExists{\docshots@tmpdir/\jobname/\hash.crop.pdf}
72     {\message{docshots: won't pdfcrop, the PDF already exists
73       (\docshots@tmpdir/\jobname/\hash.crop.pdf)^~J}}
74     {\iexec[\docshots@log,quiet]{\docshots@pdfcrop\space
75       --margins \docshots@margin\space
76       \docshots@tmpdir/\jobname/\hash.pdf
77       \docshots@tmpdir/\jobname/\hash.crop.pdf}}%

```

We configure fancyvrb:

```

78   \fvset{numbers=left,numbersep=3pt}%
79   \fvset{frame=leftline,framerule=.4pt,rulecolor=\color{gray}}%
80   \fvset{samepage=true}%
81   \fvset{baselinestretch=1}%
82   \ifdefined\docshots@small%
83     \fvset{fontsize=\small}%

```

```

84 \fi%
85 \ifdefined\docshots@tiny%
86     \fvset{fontsize=\scriptsize}%
87 \fi%

We render the two column content:

88 \beginngroup%
89 \par%
90 \tikz[baseline=(a.north)]
91     \node (a) [draw=gray,inner sep=0]
92     {\includegraphics[width=\docshots@left]
93      {\docshots@tmpdir/\jobname/\hash.crop.pdf}};%
94 \hspace{\docshots@hspace}%
95 \begin{minipage}[t]{\docshots@right}%
96     \vspace{0pt}%
97     \ifdefined\docshots@lstinputlisting%
98         \lstinputlisting{\docshots@tmpdir/\jobname/\hash.tex}%
99     \else\ifdefined\docshots@inputminted%
100         \inputminted{\docshots@inputminted}
101         {\docshots@tmpdir/\jobname/\hash.tex}%
102     \else%
103         \VerbatimInput{\docshots@tmpdir/\jobname/\hash.tex}%
104     \fi\fi%
105     \vspace{0pt}%
106 \end{minipage}%
107 \par%
108 \endgroup%
109 }

```

`\docshotPrerequisite` Then, we define `\docshotPrerequisite` command:

```

110 \newcommand\docshotPrerequisite[1]{
111     \iexec[\docshots@log,quiet]{cp #1 \docshots@tmpdir/\jobname}%
112     \message{docshots: file #1 copied to
113             \docshots@tmpdir/\jobname/#1^^J}%
114 }

```

`\docshotAfter` Finally, we define `\docshotAfter` command:

```

115 \newcommand\docshotAfter[1]{
116     \iexec[\docshots@log,quiet]{/bin/echo -n '\detokenize{#1}'
117     > \docshots@tmpdir/\jobname/after.sh}%
118     \message{docshots: file
119             \docshots@tmpdir/\jobname/after.sh created^^J}%
120 }

```

Change History

0.0.1		
General: Initial version	5	"lstinputlisting" and "inputminted" introduced to enable printing of verbatim text either via listings or minted packages.
0.0.3		
General: The command is added to enable copying of supplementary files into docshots snippets processing directory.	4	0.0.5
0.0.4		General: Package option "log" added, which enables detailed logging via exec. By default, there is no logging at all.
General: Package options		5

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

B		
<code>\begin</code>	42, 95	53, 54, 55, 57, 64, 66, 67, 70, 71, 73, 76, 77, 93, 98, 101, 103, 111, 113, 117, 119
C		
<code>\color</code>	79	
D		
<code>\def</code>	7, 11, 49	
<code>\detokenize</code>	116	
<code>\docshot</code>	41	
<code>\docshotAfter</code>	115	
<code>\docshotPrerequisite</code>	110	
<code>\docshots@dtx</code>	10, 45	
<code>\docshots@gs</code>	22, 39	
<code>\docshots@hspace</code> .	28, 94	
<code>\docshots@inputminted</code>	13, 99, 100	
<code>\docshots@left</code> ...	30, 92	
<code>\docshots@log</code> ..	7, 11, 37, 38, 39, 54, 57, 65, 67, 70, 74, 111, 116	
<code>\docshots@lstinputlisting</code>	12, 97	
<code>\docshots@margin</code> .	26, 75	
<code>\docshots@pdfcrop</code> .	24, 38, 74	
<code>\docshots@pdflatex</code> .	20, 37, 58	
<code>\docshots@right</code> ..	32, 95	
<code>\docshots@runs</code> ...	18, 56	
<code>\docshots@small</code> ..	16, 82	
<code>\docshots@tiny</code> ...	17, 85	
<code>\docshots@tmpdir</code> 14,	40, 43, 47, 50, 51,	
E		
<code>\end</code>	44, 106	
F		
<code>\foreach</code>	56	
<code>\fvset</code> .	78, 79, 80, 81, 83, 86	
H		
<code>\hash</code>	49, 51, 53, 55, 62, 68, 71, 73, 76, 77, 93, 98, 101, 103	
<code>\hspace</code>	94	
I		
<code>\iexec</code>	37, 38, 39, 40, 46, 54, 57, 65, 67, 70, 74, 111, 116	
<code>\ifdefined</code>	45, 82, 85, 97, 99	
<code>\IfFileExists</code> ..	51, 64, 71	
<code>\includegraphics</code> ...	92	
<code>\inputminted</code>	100	
J		
<code>\jobname</code>	40, 43, 47, 50, 51, 53, 54, 55, 57, 64, 66, 67, 70, 71, 73, 76, 77, 93, 98, 101, 103, 111, 113, 117, 119	
L		
<code>\linewidth</code>	29, 31, 33	
M		
<code>\message</code>	52, 63, 72, 112, 118	
N		
<code>\newcommand</code>	110, 115	
<code>\newenvironment</code>	41	
<code>\node</code>	91	
P		
<code>\par</code>	89, 107	
<code>\pdfmdfivesum</code>	49	
<code>\pgfkeys</code>	8	
<code>\ProcessPgfoptions</code> ..	36	
R		
<code>\RequirePackage</code> ...	1, 2, 3, 4, 5, 6	
S		
<code>\scriptsize</code>	86	
<code>\small</code>	83	
<code>\space</code>	37, 38, 39, 58, 68, 74, 75	
T		
<code>\tikz</code>	90	
V		
<code>\VerbatimEnvironment</code>	42	
<code>\VerbatimInput</code>	103	
<code>\vspace</code>	96, 105	