

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

Yegor Bugayenko
yegor256@gmail.com

2024-01-13, 0.4.2

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:

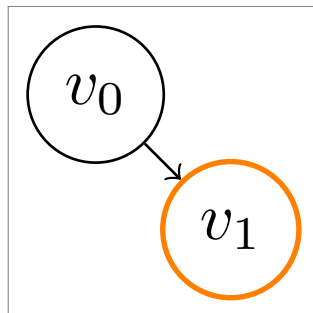
```
\begin{docshot}
\documentclass{article}
\usepackage{xcolor}
\pagestyle{empty}
\begin{document}
  Hello, {\color{orange}\LaTeX}!
\end{document}
\end{docshot}
```

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)



```

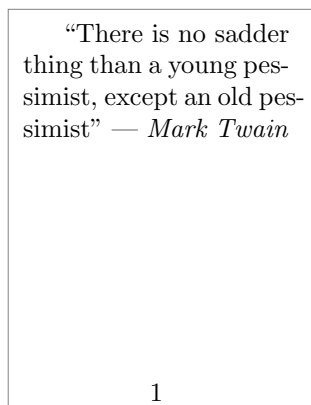
1 \documentclass{article}
2 \usepackage{tikz}
3 \pagestyle{empty}
4 \begin{document}
5 \begin{tikzpicture}
6 \node [circle,draw] (v0) {$v_0$};
7 \node [circle,draw=orange,thick,
8   below right of=v0] (v1) {$v_1$};
9 \draw [->] (v0) -- (v1);
10 \end{tikzpicture}
11 \end{document}

```

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:



```

1 \documentclass{article}
2 \usepackage[paperwidth=2in,
3   paperheight=2.3in]{geometry}
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}

```

You can also use `\pagestyle{empty}` as was done in previous docshots.

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}
```

nocrop You may disable cropping with the help of nocrop option:

```
\usepackage[nocrop]{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. You can change this using tmpdir option:

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

runs By default, we run pdf_latex 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_latex 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 Fine-tuning Options

`\docshotOptions` By default, we render the verbatim text using `\VerbatimInput` command with no options. You can add your options using `\docshotOptions` command:

```
\docshotOptions{firstline=4}
\begin{docshot}
...
\end{docshot}
```

The options will be cleaned up by the first render of a docshot.

4 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 pdf_latex 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.

5 Implementation

First, we include [iexec](#) in order to execute shell commands, such as pdf_latex and pdfcrop:

```
1 \RequirePackage{iexec}
```

Then, we include [fancyvrb](#) in order to render verbatim texts:

```
2 \RequirePackage{fancyvrb}
```

Then, we include [xcolor](#) in order to make borders gray:

```
3 \RequirePackage{xcolor}
```

Then, we include [graphicx](#) in order to be able to use `\includegraphics` command:

```
4 \RequirePackage{graphicx}
```

Then, we include [tikz](#) in order to be able to position elements:

```
5 \RequirePackage{tikz}
```

Then, we process package options with the help of [pgfopts](#):

```
6 \RequirePackage{pgfopts}
```

```
7 \RequirePackage{ifluatex}
```

```
8 \RequirePackage{ifxetex}
```

```
9 \def\docshots@log{}
```

```
10 \pgfkeys{
```

```
11   /docshots/.cd,
```

```
12   dtx/.store in=\docshots@dtx,
```

```
13   log/.code=\def\docshots@log{log},
```

```
14   nocrop/.code=\def\docshots@nocrop{},
```

```
15   lstinputlisting/.store in=\docshots@lstinputlisting,
```

```
16   inputminted/.store in=\docshots@inputminted,
```

```

17 tmpdir/.store in=\docshots@tmpdir,
18 tmpdir/.default=_docshots\ifxetex-xe\else\ifluatex-lua\fi\fi,
19 small/.store in=\docshots@small,
20 tiny/.store in=\docshots@tiny,
21 runs/.store in=\docshots@runs,
22 runs/.default=1,
23 pdflatex/.store in=\docshots@pdflatex,
24 pdflatex/.default=pdflatex,
25 gs/.store in=\docshots@gs,
26 gs/.default=gs,
27 pdfcrop/.store in=\docshots@pdfcrop,
28 pdfcrop/.default=pdfcrop,
29 margin/.store in=\docshots@margin,
30 margin/.default=5,
31 hspace/.store in=\docshots@hspace,
32 hspace/.default=.05\linewidth,
33 left/.store in=\docshots@left,
34 left/.default=.3\linewidth,
35 right/.store in=\docshots@right,
36 right/.default=.55\linewidth,
37 tmpdir,pdflatex,gs,pdfcrop,margin,hspace,left,right,runs
38 }
39 \ProcessPgfOptions{/docshots}

```

Then, we print the version of pdf_latex to T_EX log:

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

Then, we print the version of pdfcrop to T_EX log:

```

41 \ifdefined\docshots@nocrop\else%
42 \iexec[\docshots@log,quiet]{"\docshots@pdfcrop" --version}%
43 \fi%

```

Then, we print the version of ghostscript to T_EX log:

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

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

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

\docshots@mdfive Then, we define a command for MD5 hash calculating of a file, with the help of pdf_latex:

```

46 \RequirePackage{pdflatex}
47 \makeatletter
48 \newcommand\docshots@mdfive[1]{\pdf@filemdfivesum{#1}}
49 \makeatother

```

docshot Then, we define docshot environment:

```

50 \makeatletter
51 \newenvironment{docshot}
52 {\VerbatimEnvironment\begin{VerbatimOut}
53 {\docshots@tmpdir/\jobname/verbatim.tex}}
54 {\end{VerbatimOut}}%

```

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

```

55 \ifdefined\docshots@dtx%
56 \iexec[null]{perl -i -0777pe "s/(\n|^)\x{25}\\s?/\n/g"
57 \docshots@tmpdir/\jobname/verbatim.tex}%
58 \fi%

```

We calculate MD5 hashsum of the file content:

```
59 \def\hash{\docshots@mdfive{\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`:

```
60 \IfFileExists{\docshots@tmpdir/\jobname/\hash.pdf}
61 {\message{docshots: Won't render,
62   the PDF '\docshots@tmpdir/\jobname/\hash.pdf' already exists^^J}}
63 {\iexec[\docshots@log,quiet]{cp "\docshots@tmpdir/\jobname/verbatim.tex"
64   "\docshots@tmpdir/\jobname/\hash.tex"}%
65 \message{docshots: rendering at line no. \the\inputlineno^^J}%
66 \foreach \n in {1,...,\docshots@runs}{%
67   \iexec[\docshots@log,
68     stdout=\docshots@tmpdir/\jobname/\hash.stdout,
69     exit=\docshots@tmpdir/\jobname/\hash.ret,
70     quiet,trace]{cd "\docshots@tmpdir/\jobname" &&
71     "\docshots@pdflatex"
72     -interaction=errorstopmode
73     -halt-on-error
74     -shell-escape
75     \hash.tex}%
76   \ifnum\n=1\else%
77     \message{docshots: pdflatex run no.\n^^J}%
78   \fi%
79   \IfFileExists{\docshots@tmpdir/\jobname/after.sh}
80   {\iexec[\docshots@log,quiet]{chmod a+x
81     "\docshots@tmpdir/\jobname/after.sh"}
82   \iexec[\docshots@log,quiet]{cd "\docshots@tmpdir/\jobname" &&
83     ./after.sh \n\space \hash\space \hash.tex}}
84   {}}}%
```

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

```
85 \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:

```
86 \IfFileExists{\docshots@tmpdir/\jobname/\hash.crop.pdf}
87 {\message{docshots: Won't pdfcrop,
88   the PDF '\docshots@tmpdir/\jobname/\hash.crop.pdf'
89   already exists^^J}}
90 {\ifdefined\docshots@nocrop
91   \iexec[\docshots@log,quiet]{cp
92     "\docshots@tmpdir/\jobname/\hash.pdf"
93     "\docshots@tmpdir/\jobname/\hash.crop.pdf"}%
94   \else%
95     \iexec[\docshots@log,quiet]{"\docshots@pdfcrop"
96       --margins 0
97       "\docshots@tmpdir/\jobname/\hash.pdf"
98       "\docshots@tmpdir/\jobname/\hash.crop.pdf"}%
99     \fi}%
100 \message{docshots: the PDF is ready from line no. \the\inputlineno^^J}%
```

Finally, we render the two-column content:

```
101 \begingroup%
102 \par%
```

```

103 \tikz[baseline=(a.north)]
104 \node (a) [draw=gray,inner sep=\docshots@margin,
105 line width=.2pt]
106 {\includegraphics[width=\docshots@left]
107 {\docshots@tmpdir/\jobname/\hash.crop.pdf}};%
108 \hspace{\docshots@hspace}%
109 \begin{minipage}[t]{\docshots@right}%
110 \vspace{0pt}%
111 \ifdefined\docshots@lstinputlisting%
112 \expandafter\lstinputlisting\expandafter[\docshots@opts]
113 {\docshots@tmpdir/\jobname/\hash.tex}%
114 \else\ifdefined\docshots@inputminted%
115 \expandafter\inputminted\expandafter[\docshots@opts]
116 {\docshots@inputminted}
117 {\docshots@tmpdir/\jobname/\hash.tex}%
118 \else%
119 \fvset{numbers=left,numbersep=3pt}%
120 \fvset{frame=leftline,framerule=.2pt,rulecolor=\color{gray}}%
121 \fvset{samepage=true}%
122 \fvset{commandchars=none}%
123 \fvset{baselinestretch=1}%
124 \ifdefined\docshots@small%
125 \fvset{fontsize=\small}%
126 \fi%
127 \ifdefined\docshots@tiny%
128 \fvset{fontsize=\scriptsize}%
129 \fi%
130 \ifdefined\docshots@opts%
131 \fvset{\docshots@opts}%
132 \fi%
133 \VerbatimInput{\docshots@tmpdir/\jobname/\hash.tex}%
134 \fi\fi%
135 \vspace{0pt}%
136 \end{minipage}%
137 \par%
138 \endgroup%
139 \docshotOptions{}%
140 }
141 \makeatother

```

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

```

142 \makeatletter
143 \newcommand\docshotPrerequisite[1]{
144 \iexec[\docshots@log,quiet]{cp #1 "\docshots@tmpdir/\jobname"}%
145 \message{docshots: File '#1' copied to
146 '\docshots@tmpdir/\jobname/#1'^^J}%
147 }
148 \makeatother

```

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

```

149 \makeatletter
150 \newcommand\docshotAfter[1]{
151 \iexec[\docshots@log,quiet]{/bin/echo -n '\detokenize{#1}'
152 > "\docshots@tmpdir/\jobname/after.sh"}%

```



```

153 \message{docshots: File
154   '\docshots@tmpdir/\jobname/after.sh' created^^J}%
155 }
156 \makeatother

```

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

```

157 \makeatletter
158 \gdef\docshots@opts{}
159 \newcommand\docshotOptions[1]{%
160   \gdef\docshots@opts{#1}%
161 }
162 \makeatother

```

Change History

0.0.1	General: Initial version 5	0.1.1	<code>\docshots@mdfive</code> : New supplementary command added to calculate MD5 sum of a file. 6
0.0.3	General: The command is added to enable copying of supplementary files into docshots snippets processing directory. 4	0.2.0	General: New package option "nocrop" added to allow disabling of "pdfcrop" execution. 5
0.0.4	General: Package options "lstinputlisting" and "inputminted" introduced to enable printing of verbatim text either via listings or minted packages. 5	0.3.0	<code>\docshotOptions</code> : New command introduced to help specify custom options for verbatim environments. 9
0.0.5	General: Package option "log" added, which enables detailed logging via exec. By default, there is no logging at all. 5	0.3.1	<code>docshot</code> : A bug fixed, now handling of leading percentage symbols is done right. 6
0.1.0	<code>docshot</code> : The output is saved to a hash-named file, for better uniqueness of temporary files. . . . 7	0.4.0	<code>docshot</code> : Now, <code>\TeXoutput</code> has a log message with the number of the line in the source file, which we render. 7

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		
\begin	86, 88, 92, 93, 97, 98, 107, 113, 117, 133, 144, 146, 152, 154	98, 107, 113, 117, 133, 144, 146, 152, 154
C		
\color	120	
D		
\def	9, 13, 14, 59	
\detokenize	151	
\docshot	50	
\docshotAfter	149	
\docshotOptions	139, 157	
\docshotPrerequisite	142	
\docshots@dtx	12, 55	
\docshots@gs	25, 44	
\docshots@hspace	31, 108	
\docshots@inputminted	16, 114, 116	
\docshots@left	33, 106	
\docshots@log	9, 13, 40, 42, 44, 63, 67, 80, 82, 85, 91, 95, 144, 151	
\docshots@lstinputlisting	15, 111	
\docshots@margin	29, 104	
\docshots@mdfive	46, 59	
\docshots@nocrop	14, 41, 90	
\docshots@opts	112, 115, 130, 131, 158, 160	
\docshots@pdfcrop	27, 42, 95	
\docshots@pdflatex	23, 40, 71	
\docshots@right	35, 109	
\docshots@runs	21, 66	
\docshots@small	19, 124	
\docshots@tiny	20, 127	
\docshots@tmpdir	17, 45, 53, 57, 59, 60, 62, 63, 64, 68, 69, 70, 79, 81, 82, 85, 86, 88, 92, 93, 97,	
E		
\end	54, 136	
\expandafter	112, 115	
F		
\foreach	66	
\fvset	119, 120, 121, 122, 123, 125, 128, 131	
G		
\gdef	158, 160	
H		
\hash	59, 60, 62, 64, 68, 69, 75, 83, 86, 88, 92, 93, 97, 98, 107, 113, 117, 133	
\hspace	108	
I		
\iexec	40, 42, 44, 45, 56, 63, 67, 80, 82, 85, 91, 95, 144, 151	
\ifdefined	41, 55, 90, 111, 114, 124, 127, 130	
\IfFileExists	60, 79, 86	
\ifluatex	18	
\ifnum	76	
\ifxetex	18	
\includegraphics	106	
\inputlineno	65, 100	
\inputminted	115	
J		
\jobname	45, 53, 57, 59, 60, 62, 63, 64, 68, 69, 70, 79, 81, 82, 85, 86, 88, 92, 93, 97,	
L		
\linewidth	32, 34, 36	
\lstinputlisting	112	
M		
\makeatletter	47, 50, 142, 149, 157	
\makeatother	49, 141, 148, 156, 162	
\message	61, 65, 77, 87, 100, 145, 153	
N		
\newcommand	48, 143, 150, 159	
\newenvironment	51	
\node	104	
P		
\par	102, 137	
\pdf@filemdfivesum	48	
\pgfkeys	10	
\ProcessPgfOptions	39	
R		
\RequirePackage	1, 2, 3, 4, 5, 6, 7, 8, 46	
S		
\scriptsize	128	
\small	125	
\space	83	
T		
\the	65, 100	
\tikz	103	
V		
\VerbatimEnvironment	52	
\VerbatimInput	133	
\vspace	110, 135	