

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

Yegor Bugayenko
yegor256@gmail.com

2026-10-15, 0.5.2

The T_EX processor must be invoked with the `--shell-escape` option, and the following must be installed on the host system: `pdflatex`, [Perl](#), [Ghostscript](#), and [pdfcrop](#). The `--shell-escape` option is required because the package launches these external programs from within the T_EX run in order to compile and crop each snippet. The package operates on Windows as well as on Unix, provided these tools are on the PATH; on Windows, version 0.16.1 or newer of [iexec](#) is required as well.

1 Introduction

When demonstrating to the readers of a document how certain T_EX commands are to be employed, the most cogent approach is to display precisely how the entire document will be rendered as a PDF, by means of a subprocess (such as `pdflatex`) that performs the rendering. To the [author's best](#) knowledge, no existing package addresses this need; hence the present, modest package was devised.

For instance, the following code:

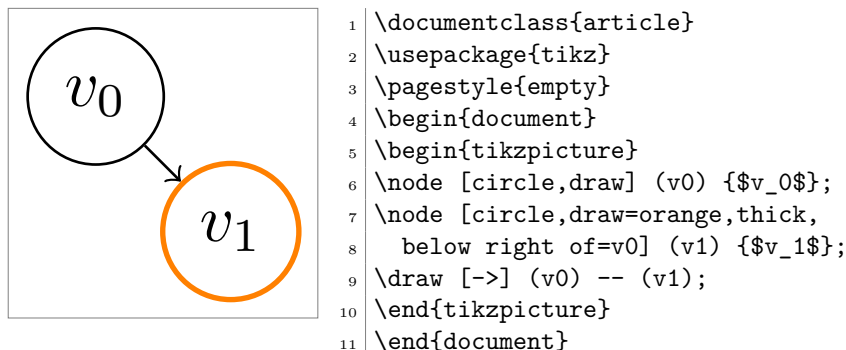
```
\begin{docshot}  
\documentclass{article}  
\usepackage{xcolor}  
\pagestyle{empty}  
\begin{document}  
  Hello, {\color{orange}\LaTeX}!  
\end{document}  
\end{docshot}
```

is rendered thus:

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

*The sources are on GitHub at [yegor256/docshots](#)

A more elaborate example follows:

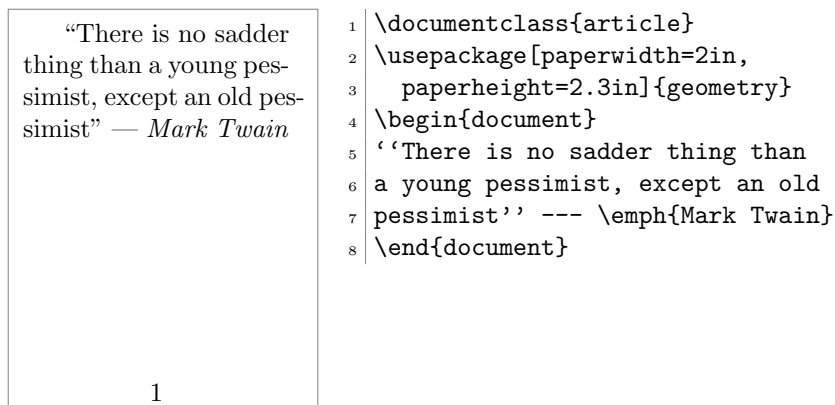


The body of every `docshot` environment must be a complete, self-contained $\text{\LaTeX}$ document: it begins with `\documentclass`, loads whatever packages it requires, and wraps its content in `\begin{document}` and `\end{document}`. The body is written verbatim to a separate `.tex` file and compiled on its own, exactly as the two examples above demonstrate.

The figure on the left is produced by a subprocess that executes `pdflatex` on the `.tex` content extracted from the source file. Once the $\text{\TeX}$ sources have been successfully processed, `pdfcrop` is invoked to trim the document.

`pdflatex` is run with the `-interaction=errorstopmode` and `-halt-on-error` options. Consequently, $\text{\TeX}$ processing halts at the first error encountered. The $\text{\TeX}$ log should be consulted to ascertain the cause of any failure; the `log` option, described below, copies the full output of `pdflatex` into that log.

Whenever text rather than a single figure is to be rendered, the `geometry` package is recommended for adjusting the page dimensions:



The use of `\pagestyle{empty}`, as in the preceding docshots, is likewise possible.

2 Package Options

All options listed below are supplied to the package in the usual way, within the optional argument of `\usepackage[...]{docshots}`.

`pdflatex` The default command-line tool for converting `.tex` into `.pdf` is `pdflatex`. This

may, however, be altered by means of the `pdflatex` package option, for example:

```
\documentclass{article}
\usepackage[pdflatex=/usr/local/bin/pdflatex]{docshots}
\begin{document}
\begin{docshot}
...
\end{docshot}
\end{document}
```

gs The default location of Ghostscript is simply `gs` on Unix and `gswin64c` on Windows. This may be altered by means of the `gs` package option, for example:

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

pdfcrop The default location of `pdfcrop` is simply `pdfcrop`. This may be altered by means of the `pdfcrop` package option, for example:

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

margin A margin is kept between the snapshot and the gray frame drawn around it. The default value of 5 points may be modified by means of the `margin` package option:

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

nocrop Cropping may be disabled by means of the `nocrop` option:

```
\usepackage[nocrop]{docshots}
```

hspace The horizontal distance between the image and its verbatim `TEX` source may be configured by means of the `hspace` package option:

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

left The default width of the image may be altered by the `left` option, while the **right** width of the verbatim `TEX` source may be adjusted by the `right` option:

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

snippet By default, the verbatim `TEX` source is placed to the right of the snapshot. The `snippet` option accepts one of three values—`left`, `right`, or `none`—to place the source on either side of the snapshot or to suppress it entirely, leaving the snapshot alone:

```
\usepackage[snippet=left]{docshots}
```

dtx Whenever this package is employed within `.dtx` documentation, the `dtx` package option must be added. Under this option, all leading comment symbols (the `%` characters) are removed from the start of each line:

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

tmpdir The default location for temporary files is `_docshots` (`_docshots-xe` under `XeTeX` and `_docshots-lua` under `LuaTeX`). This may be altered by means of the `tmpdir` option:

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

runs By default, `pdflatex` is invoked but once per docshot. This number may be modified by means of the **runs** package option, which proves serviceable when Bib_TE_X processing is required, for example:

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

small The latitude available for formatting verbatim texts is limited; nevertheless, the **tiny** font may be reduced somewhat by means of the **small** package option, or made considerably smaller still by the **tiny** option:

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

log By means of the **log** option, the package writes all available logs into the main T_EX log. By default, this is not done, and the output of `pdflatex` compilation, for instance, remains hidden. The option is employed thus:

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

inputminted By default, the verbatim text is rendered by means of the `\VerbatimInput` command. This may be substituted with `\inputminted` from the [minted](#) package, for example:

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

lstinputlisting By default, the verbatim text is rendered by means of the `\VerbatimInput` command. This may be substituted with `\lstinputlisting` from the [listings](#) package, for example:

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

3 Fine-tuning Options

docshot (*env.*) Every **docshot** environment accepts an optional argument, in which per-docshot settings are supplied as a comma-separated list of **key=value** pairs. Four keys are recognised: **snippet**, **options**, **after**, and **prerequisite**. Each applies solely to the environment that carries it, leaving the defaults intact for the docshots that follow:

```
\begin{docshot}[snippet=left, options={firstline=4}]
...
\end{docshot}
```

snippet The **snippet** key overrides the package-wide **snippet** option for a single docshot, accepting **left**, **right**, or **none**:

```
\begin{docshot}[snippet=none]
...
\end{docshot}
```

options By default, the verbatim text is rendered by means of the `\VerbatimInput` command with no options. Additional options may be supplied through the `options` key; they are passed to `\lstset` or to `\inputminted` instead, when the `lstinputlisting` or `inputminted` package option is in effect:

```
\begin{docshot}[options={firstline=4}]
...
\end{docshot}
```

prerequisite Whenever a file is required to be present alongside the `.tex` snippet during rendering by `pdflatex`, the `prerequisite` key names a file to be copied from the current directory to the temporary directory in which all snippets are rendered. The key may be repeated for several files:

```
\begin{docshot}[prerequisite=duck.jpg]
\documentclass{article}
\usepackage{graphicx}
\pagestyle{empty}
\begin{document}
  A favoured photograph of a duck:
  \includegraphics[width=2in]{duck.jpg}
\end{document}
\end{docshot}
```

after Should some action be required after each `pdflatex` run of a `docshot`, the `after` key supplies the command to run. For example, when a bibliography is to remain available to a snippet and `biber` is to be invoked on it in order to process citations:

```
\begin{docshot}[prerequisite=main.bib, after={biber $2}]
\documentclass{article}
\usepackage[natbib=true]{biblatex}
\addbibresource{main.bib}
\pagestyle{empty}
\begin{document}
  The book of \citet{knuth1984} is essential reading.
  \printbibliography
\end{document}
\end{docshot}
```

The command supplied to the `after` key receives three arguments at runtime:

- \$1 the cycle of `pdflatex` processing (1, 2, ...),
 - \$2 the hash of the snippet,
 - \$3 the name of the `.tex` file.
- \$3 is, in essence, \$2 with an appended `.tex` suffix.

4 Implementation

First, we include `iexec` in order to execute shell commands, such as `pdflatex` 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 the `\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](#): We also detect Windows, in order to pick the right default for Ghostscript, the right file separator for shell commands (a backslash, since `cmd.exe` rejects forward slashes in redirection targets), and an expanding variant of the string-replacement command for later use by the `after` key:

```
6 \RequirePackage{pgfopts}
7 \RequirePackage{ifluatex}
8 \RequirePackage{ifxetex}
9 \RequirePackage{expl3}
10 \newif\ifdocshots@windows
11 \ExplSyntaxOn
12 \sys_if_platform_windows:TF{\docshots@windowstrue}{\docshots@windowsfalse}
13 \cs_generate_variant:Nn \str_replace_all:Nnn { Nne }
14 \newcommand\docshots@aftercmd[3]{%
15   \str_set:Nx \l_tmpa_str { \docshots@after }%
16   \str_replace_all:Nne \l_tmpa_str { $1 } { #1 }%
17   \str_replace_all:Nne \l_tmpa_str { $2 } { #2 }%
18   \str_replace_all:Nne \l_tmpa_str { $3 } { #3 }%
19   \cs_set:Npx \docshots@aftersh { \str_use:N \l_tmpa_str }%
20 }
21 \ExplSyntaxOff
22 \ifdocshots@windows
23   \def\docshots@gsdefault{gswin64c}%
24   \edef\docshots@sep{\backslashchar}%
25 \else
26   \def\docshots@gsdefault{gs}%
27   \def\docshots@sep{/}%
28 \fi
```

We render inside the temporary directory, so that `pdflatex` finds the prerequisites there and leaves its output next to the snippet. On Windows this relies on `iexec` 0.16.1 or newer, which runs the command in a `cmd /c` subshell; with an older `iexec` the `cd` would outlive the parentheses, the exit-code file would land in the wrong directory, and the run would stall, so we stop early:

```
29 \ifdocshots@windows
30   \@ifpackagelater{iexec}{2026/07/13}{}{%
31     \PackageError{docshots}{iexec 0.16.1 or newer is required on Windows}
32     {Older versions of iexec misplace the exit-code file after a cd}}%
33 \fi
34 \def\docshots@render{cd "\docshots@tmpdir\docshots@sep\jobname" &&
35   "\docshots@pdflatex"
36   -interaction=errorstopmode -halt-on-error -shell-escape
```

```

37 \hash.tex}%
38 \def\docshots@log{}
39 \pgfkeys{
40   /docshots/.cd,
41   dtx/.store in=\docshots@dtx,
42   log/.code=\def\docshots@log{log},
43   nocrop/.code=\def\docshots@nocrop{},
44   lstinputlisting/.store in=\docshots@lstinputlisting,
45   inputminted/.store in=\docshots@inputminted,
46   tmpdir/.store in=\docshots@tmpdir,
47   tmpdir/.default=_docshots\ifxetex-xe\else\ifluatex-lua\fi\fi,
48   small/.store in=\docshots@small,
49   tiny/.store in=\docshots@tiny,
50   runs/.store in=\docshots@runs,
51   runs/.default=1,
52   pdflatex/.store in=\docshots@pdflatex,
53   pdflatex/.default=pdflatex,
54   gs/.store in=\docshots@gs,
55   gs/.default=\docshots@gsdefault,
56   pdfcrop/.store in=\docshots@pdfcrop,
57   pdfcrop/.default=pdfcrop,
58   margin/.store in=\docshots@margin,
59   margin/.default=5,
60   hspace/.store in=\docshots@hspace,
61   hspace/.default=.05\linewidth,
62   left/.store in=\docshots@left,
63   left/.default=.3\linewidth,
64   right/.store in=\docshots@right,
65   right/.default=.55\linewidth,
66   snippet/.store in=\docshots@snippet,
67   snippet/.default=right,
68   tmpdir,pdflatex,gs,pdfcrop,margin,hspace,left,right,snippet,runs
69 }
70 \ProcessPgfOptions{/docshots}

```

We remember the three admissible values of the `snippet` option, so that a docshot may later compare its own placement against them:

```

71 \def\docshots@snippetleft{left}
72 \def\docshots@snippetright{right}
73 \def\docshots@snippetnone{none}

```

Then, we print the version of `pdflatex` to the $\TeX$ log:

```

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

```

Then, we print the version of `pdfcrop` to the $\TeX$ log:

```

75 \ifdefined\docshots@nocrop\else%
76   \iexec[\docshots@log,quiet]{"\docshots@pdfcrop" --version}%
77 \fi%

```

Then, we print the version of `ghostscript` to the $\TeX$ log, ignoring a failure, since Ghostscript is not strictly required for rendering:

```

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

```

Then, we make a directory where all temporary files will be kept. We use Perl instead of `mkdir`, so that the code works on Windows too:

```

79 \iexec[null]{perl -MFile::Path=make_path -e "make_path(shift)"

```

```
80 "\docshots@tmpdir/\jobname"}%
```

`\docshots@mdfive` Then, we define a command for MD5 hash calculating of a file, with the help of [pdftexcmds](#):

```
81 \RequirePackage{pdftexcmds}
82 \newcommand\docshots@mdfive[1]{\pdf@filemdfivesum{#1}}
```

`\docshots@prereq` Then, we define an internal command that copies a prerequisite file from the current directory into the temporary one, so that `pdflatex` finds it during rendering:

```
83 \newcommand\docshots@prereq[1]{%
84   \iexec[\docshots@log,quiet]{perl -MFile::Copy -e "copy(shift,shift) or die q(cannot copy)"
85     "#1" "\docshots@tmpdir/\jobname/#1"}%
86   \message{docshots: File '#1' copied to
87     '\docshots@tmpdir/\jobname/#1'^^J}%
88 }
```

Then, we register the keys accepted by the optional argument of the `docshot` environment. Each stores into the same internal macro that the environment reads while rendering:

```
89 \pgfkeys{
90   /docshots/shot/.cd,
91   snippet/.store in=\docshots@thissnippet,
92   options/.store in=\docshots@opts,
93   after/.code=\edef\docshots@after{\detokenize{#1}},
94   prerequisite/.code=\docshots@prereq{#1},
95 }
```

Then, we define the reader of that optional argument. Before peeking for the `[`, we make the end-of-line active, exactly as `fancyvrb` does: an absent optional argument then leaves the raw end-of-line in place, which `fancyvrb` needs in order to find where the verbatim body begins. A plain `\@ifnextchar` would skip that end-of-line and mislead the verbatim scanner:

```
96 \def\docshots@shot{%
97   \catcode'\^^M=\active%
98   \@ifnextchar[%
99     {\catcode'\^^M=5 \docshots@grab}%
100    {\catcode'\^^M=5 \docshots@begin}}
101 \def\docshots@grab[#1]{\pgfkeys{/docshots/shot/.cd,#1}\docshots@begin}
102 \def\docshots@begin{\VerbatimEnvironment\begin{\VerbatimOut}
103   {\docshots@tmpdir/\jobname/docshots-temp.tex}}
```

`docshot` Then, we define the `docshot` environment. Its optional argument carries the per-`docshot` settings (`snippet`, `options`, `after`, and `prerequisite`), which we parse under normal catcodes, before the body is captured verbatim. The defaults are set locally, so that each `docshot` starts afresh and no setting leaks into the next:

```
104 \newenvironment{docshot}
105 {\let\docshots@thissnippet\docshots@snippet%
106   \def\docshots@opts{}%
107   \docshots@shot}
108 {\end{\VerbatimOut}}%
```

If we are in `dtx` mode, leading percent characters must be removed. The substitution stays line-oriented and anchored with `\A`, since the whole-file form needed the

caret and vertical-bar metacharacters that `cmd.exe` reads as an escape and a pipe, mangling the command on Windows:

```

109 \ifdefined\docshots@dtx%
110 \iexec[null]{perl -i -pe "s/\\A\\x{25} ?//"}
111 \docshots@tmpdir/\jobname/docshots-temp.tex}%
112 \fi%
```

We calculate MD5 hashsum of the file content:

```

113 \def\hash{\docshots@mdfive{\docshots@tmpdir/\jobname/docshots-temp.tex}}%
```

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

```

114 \IfFileExists{\docshots@tmpdir/\jobname/\hash.pdf}
115 {\message{docshots: Won't render,
116 the PDF '\docshots@tmpdir/\jobname/\hash.pdf' already exists^^J}}
117 {\iexec[\docshots@log,quiet]{perl -MFile::Copy -e "copy(shift,shift) or die q(cannot copy
118 "\docshots@tmpdir/\jobname/docshots-temp.tex"
119 "\docshots@tmpdir/\jobname/\hash.tex")}%
120 \message{docshots: rendering at line no. \the\inputlineno^^J}%
121 \foreach \n in {1,...,\docshots@runs}{%
122 \iexec[\docshots@log,
123 stdout=\docshots@tmpdir\docshots@sep\jobname\docshots@sep\hash.stdout,
124 exit=\docshots@tmpdir\docshots@sep\jobname\docshots@sep\hash.ret,
125 quiet,trace]{\docshots@render}%
126 \ifnum\n=1\else%
127 \message{docshots: pdflatex run no.\n^^J}%
128 \fi%
```

If a post-processing command was registered by the `after` key, we substitute its `$1`, `$2`, and `$3` placeholders and execute it. There is no shell script and no `chmod`, so the mechanism works on Windows too:

```

129 \ifdefined\docshots@after%
130 \begingroup%
131 \docshots@aftercmd{\n}{\hash}{\hash.tex}%
132 \iexec[\docshots@log,quiet]{cd "\docshots@tmpdir\docshots@sep\jobname" &&
133 \docshots@aftersh}%
134 \endgroup%
135 \fi}}%
```

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:

```

136 \IfFileExists{\docshots@tmpdir/\jobname/\hash.crop.pdf}
137 {\message{docshots: Won't pdfcrop,
138 the PDF '\docshots@tmpdir/\jobname/\hash.crop.pdf'
139 already exists^^J}}
140 {\ifdefined\docshots@nocrop
141 \iexec[\docshots@log,quiet]{perl -MFile::Copy -e "copy(shift,shift) or die q(cannot copy
142 "\docshots@tmpdir/\jobname/\hash.pdf"
143 "\docshots@tmpdir/\jobname/\hash.crop.pdf")}%
144 \else%
145 \iexec[\docshots@log,quiet]{"\docshots@pdfcrop"
146 --gscmd "\docshots@gs"
147 --margins 0
148 "\docshots@tmpdir/\jobname/\hash.pdf"
149 "\docshots@tmpdir/\jobname/\hash.crop.pdf"}%
```

```

150     \fi}%
151   \message{docshots: the PDF is ready from line no. \the\inputlineno^^J}%

```

Finally, we render the content. The placement was resolved by the optional argument, defaulting to the package-wide `snippet` setting. The snapshot is a framed image, and the source is a top-aligned minipage; we keep each in a macro of its own, so that either may precede the other:

```

152   \begingroup%
153   \par%
154   \def\docshots@image{\tikz[baseline=(a.north)]
155     \node (a) [draw=gray,inner sep=\docshots@margin,
156       line width=.2pt]
157     {\includegraphics[width=\docshots@left]
158       {\docshots@tmpdir/\jobname/\hash.crop.pdf}};}%
159   \def\docshots@code{\begin{minipage}[t]{\docshots@right}%
160     \vspace{0pt}%
161     \ifdefined\docshots@lstinputlisting%
162       \ifdefined\docshots@opts%
163         \expandafter\lstset\expandafter{\docshots@opts}%
164       \fi%
165       \lstinputlisting{\docshots@tmpdir/\jobname/\hash.tex}%
166     \else\ifdefined\docshots@inputminted%
167       \expandafter\inputminted\expandafter[\docshots@opts]
168       {\docshots@inputminted}
169       {\docshots@tmpdir/\jobname/\hash.tex}%
170     \else%
171       \fvset{numbers=left,numbersep=3pt}%
172       \fvset{frame=leftline,framerule=.2pt,rulecolor=\color{gray}}%
173       \fvset{samepage=true}%
174       \fvset{commandchars=none}%
175       \fvset{baselinestretch=1}%
176       \ifdefined\docshots@small%
177         \fvset{fontsize=\small}%
178       \fi%
179       \ifdefined\docshots@tiny%
180         \fvset{fontsize=\scriptsize}%
181       \fi%
182       \ifdefined\docshots@opts%
183         \expandafter\fvset\expandafter{\docshots@opts}%
184       \fi%
185       \edef\FancyVerbHFuzz{\number\maxdimen sp}%
186       \VerbatimInput{\docshots@tmpdir/\jobname/\hash.tex}%
187     \fi\fi%
188     \vspace{0pt}%
189   \end{minipage}}%

```

According to the `snippet` setting, the source stands to the left of the snapshot, to its right, or is omitted altogether:

```

190   \ifx\docshots@thissnippet\docshots@snippetnone%
191     \docshots@image%
192   \else\ifx\docshots@thissnippet\docshots@snippetleft%
193     \docshots@code\hspace{\docshots@hspace}\docshots@image%
194   \else\ifx\docshots@thissnippet\docshots@snippetright%
195     \docshots@image\hspace{\docshots@hspace}\docshots@code%
196   \else%

```

```

197 \PackageError{docshots}{Unknown snippet
198 '\docshots@thissnippet', use left, right, or none}{}%
199 \fi\fi\fi%
200 \par%
201 \endgroup%
202 }

203 \endinput

```

Change History

0.0.1	General: Initial version	5	0.4.0	docshot : Now, $\TeX$ output has a log message with the number of the line in the source file, which we render.	9
0.0.3	\docshots@prereq : New supplementary command added to copy prerequisite files into the rendering directory.	8	0.5.0	General: Windows compatibility: file operations go through Perl, and the "gs" default becomes "gswin64c" on Windows.	6
0.0.4	General: Package options "lstinutlisting" and "inputminted" introduced to enable printing of verbatim text via either the listings or the minted package.	6	0.6.0	General: Package option "snippet" added, with values "left", "right", and "none", to place the verbatim source on either side of the snapshot or to omit it entirely.	6
0.0.5	General: Package option "log" added, which enables detailed logging via iexec. By default, there is no logging at all.	6		The environment gains an optional argument that carries the per-docshot settings, replacing the docshotSnippet, docshotOptions, docshotAfter, and docshotPrerequisite commands.	4
0.1.0	docshot : The output is saved to a hash-named file, for better uniqueness of temporary files.	9		docshot : The environment gains an optional argument carrying per-docshot settings; the commands docshotSnippet, docshotOptions, docshotAfter, and docshotPrerequisite are removed in its favour.	8
0.1.1	\docshots@mdfive : New supplementary command added to calculate MD5 sum of a file.	8	0.6.1	General: Windows renders inside the temporary directory again, exactly like Unix, which requires iexec 0.16.1 or newer there.	6
0.2.0	General: New package option "nocrop" added to allow disabling of "pdfcrop" execution.	6			
0.3.1	docshot : A bug fixed, now handling of leading percentage symbols is done right.	9			

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.

Symbols		
\@backslashchar ... <i>24</i>	\docshots@log <i>38, 42, 74,</i> <i>76, 78, 84, 117,</i> <i>122, 132, 141, 145</i>	\docshots@windowstrue <i>12</i>
\@ifnextchar <i>98</i>		\dtx <i>3</i>
\@ifpackagelater .. <i>30</i>		
\^ <i>97, 99, 100</i>	\docshots@lstinputlisting <i>44, 161</i>	E
A	\docshots@margin <i>58, 155</i>	\edef <i>24, 93, 185</i>
\active <i>97</i>	\docshots@mdfive <i>81, 113</i>	\end <i>108, 189</i>
\after <i>5</i>	\docshots@nocrop <i>43, 75, 140</i>	\endinput <i>203</i>
B	\docshots@opts <i>92, 106, 162,</i> <i>163, 167, 182, 183</i>	environments:
\begin <i>102, 159</i>		docshot <i>4</i>
C	\docshots@pdfcrop <i>56, 76, 145</i>	\expandafter <i>163, 167, 183</i>
\catcode ... <i>97, 99, 100</i>	\docshots@pdflatex <i>35, 52, 74</i>	\ExplSyntaxOff <i>21</i>
\color <i>172</i>	\docshots@prereq <i>83, 94</i>	\ExplSyntaxOn <i>11</i>
\cs <i>13, 19</i>	\docshots@render <i>34, 125</i>	F
D	\docshots@right <i>64, 159</i>	\FancyVerbHFuzz ... <i>185</i>
\def .. <i>23, 26, 27, 34,</i> <i>38, 42, 43, 71, 72,</i> <i>73, 96, 101, 102,</i> <i>106, 113, 154, 159</i>	\docshots@runs . <i>50, 121</i>	\foreach <i>121</i>
\detokenize <i>93</i>	\docshots@sep .. <i>24,</i> <i>27, 34, 123, 124, 132</i>	\fvset <i>171,</i> <i>172, 173, 174,</i> <i>175, 177, 180, 183</i>
\docshot <i>104</i>	\docshots@shot . <i>96, 107</i>	G
docshot (env.) <i>4</i>	\docshots@small <i>48, 176</i>	\gs <i>3</i>
\docshots@after <i>15, 93, 129</i>	\docshots@snippet <i>66, 105</i>	H
\docshots@aftercmd <i>14, 131</i>	\docshots@snippetleft <i>71, 192</i>	\hash <i>37, 113,</i> <i>114, 116, 119,</i> <i>123, 124, 131,</i> <i>136, 138, 142,</i> <i>143, 148, 149,</i> <i>158, 165, 169, 186</i>
\docshots@aftersh <i>19, 133</i>	\docshots@snippetnone <i>73, 190</i>	\hspace <i>3, 193, 195</i>
\docshots@begin <i>100, 101, 102</i>	\docshots@snippetright <i>72, 194</i>	I
\docshots@code <i>159, 193, 195</i>	\docshots@thissnippet <i>91, 105,</i> <i>190, 192, 194, 198</i>	\iexec ... <i>74, 76, 78,</i> <i>79, 84, 110, 117,</i> <i>122, 132, 141, 145</i>
\docshots@dtx .. <i>41, 109</i>	\docshots@tiny . <i>49, 179</i>	\ifdefined <i>75, 109, 129,</i> <i>140, 161, 162,</i> <i>166, 176, 179, 182</i>
\docshots@grab . <i>99, 101</i>	\docshots@tmpdir <i>34, 46,</i> <i>80, 85, 87, 103,</i> <i>111, 113, 114,</i> <i>116, 118, 119,</i> <i>123, 124, 132,</i> <i>136, 138, 142,</i> <i>143, 148, 149,</i> <i>158, 165, 169, 186</i>	\ifdocshots@windows <i>10, 22, 29</i>
\docshots@gs <i>54, 78, 146</i>		\IfFileExists . <i>114, 136</i>
\docshots@gsdefault <i>23, 26, 55</i>	\docshots@windowsfalse <i>12</i>	\ifluatex <i>47</i>
\docshots@hspace <i>60, 193, 195</i>		\ifnum <i>126</i>
\docshots@image <i>154, 191, 193, 195</i>		\ifx <i>190, 192, 194</i>
\docshots@inputminted <i>45, 166, 168</i>		\ifxetex <i>47</i>
\docshots@left . <i>62, 157</i>		

<code>\includegraphics</code> .. 157	<code>\maxdimen</code> 185	R
<code>\inputlineno</code> .. 120, 151	<code>\message</code> ... 86, 115,	<code>\RequirePackage</code> 1, 2,
<code>\inputminted</code> 4, 167	120, 127, 137, 151	3, 4, 5, 6, 7, 8, 9, 81
J	N	<code>\right</code> 3
<code>\jobname</code> 34,	<code>\newcommand</code> .. 14, 82, 83	<code>\runs</code> 4
80, 85, 87, 103,	<code>\newenvironment</code> ... 104	S
111, 113, 114,	<code>\newif</code> 10	<code>\scriptsize</code> 180
116, 118, 119,	<code>\nocrop</code> 3	<code>\small</code> 4, 177
123, 124, 132,	<code>\node</code> 155	<code>\snippet</code> 3, 4
136, 138, 142,	<code>\number</code> 185	<code>\str</code> 13, 15, 16, 17, 18, 19
143, 148, 149,	O	<code>\sys</code> 12
158, 165, 169, 186	<code>\options</code> 5	T
L	P	<code>\the</code> 120, 151
<code>\l</code> 15, 16, 17, 18, 19	<code>\PackageError</code> .. 31, 197	<code>\tikz</code> 154
<code>\left</code> 3	<code>\par</code> 153, 200	<code>\tiny</code> 4
<code>\linewidth</code> ... 61, 63, 65	<code>\pdf@filemdfivesum</code> . 82	<code>\tmpdir</code> 3
<code>\log</code> 4	<code>\pdfcrop</code> 3	V
<code>\lstinputlisting</code> 4, 165	<code>\pdflatex</code> 2	<code>\VerbatimEnvironment</code>
<code>\lstset</code> 163	<code>\pgfkeys</code> ... 39, 89, 101	 102
M	<code>\prerequisite</code> 5	<code>\VerbatimInput</code> 186
<code>\margin</code> 3	<code>\ProcessPgfOptions</code> . 70	<code>\vspace</code> 160, 188