

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

Yegor Bugayenko
yegor256@gmail.com

2026-10-08, 0.5.1

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`.

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:

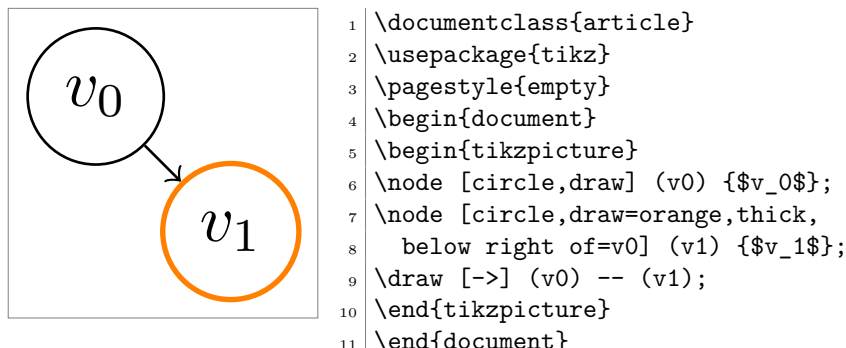
```
\begin{docshot}  
\documentclass{article}  
\usepackage{xcolor}  
\pagestyle{empty}  
\begin{document}  
  Hello, {\color{orange}\LaTeX}!  
\end{document}  
\end{docshot}
```

is rendered thus:

Hello, L^AT_EX!	<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 in GitHub at [yegor256/docshots](#)

A more elaborate example follows:

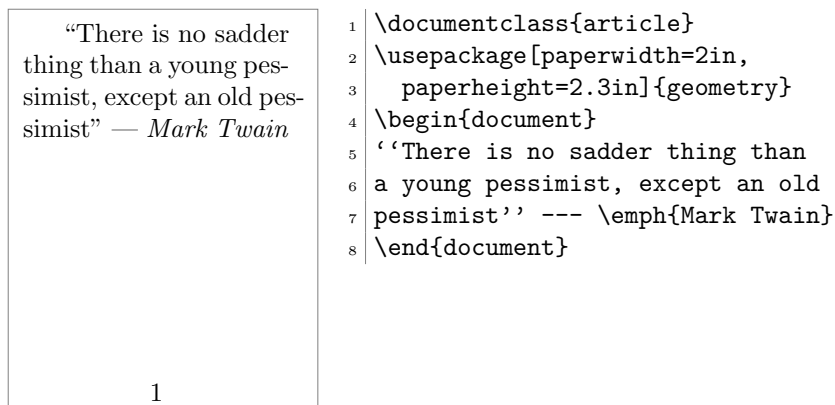


The body of every `docshot` environment must be a complete, self-contained $\text{\LaTeX}$ document: it begins with `\documentclass`, loads whatever packages it requires, and wraps its content in `\begin{document}` and `\end{document}`. The body is written verbatim to a separate `.tex` file and compiled 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}
Hello, world!
\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 When the rendered PDF is cropped, a margin is preserved around the content. The default value may be modified by 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 T_EX 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 T_EX source may be adjusted by the `right` option:

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

snippet By default, the verbatim T_EX 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`. 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:

```
\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{acmart}
\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
```

On Unix we render inside the temporary directory. On Windows we cannot, since `cmd.exe` keeps a `cd` in effect after the parentheses close, which would send `iexec`'s exit-code file to the wrong place; instead we stay put and point `pdflatex` at the directory with `-output-directory`. The `pdflatex` arguments keep forward slashes, since $\TeX$ reads a backslash as an escape character:

```
29 \ifdocshots@windows
30   \def\docshots@render{"\docshots@pdflatex"
31     -interaction=errorstopmode -halt-on-error -shell-escape
32     -output-directory="\docshots@tmpdir/\jobname"
33     "\docshots@tmpdir/\jobname/\hash.tex"}%
34 \else
35   \def\docshots@render{cd "\docshots@tmpdir/\jobname" &&
36     "\docshots@pdflatex"
```

```

37     -interaction=errorstopmode -halt-on-error -shell-escape
38     \hash.tex}%
39 \fi
40 \def\docshots@log{}
41 \pgfkeys{
42   /docshots/.cd,
43   dtx/.store in=\docshots@dtx,
44   log/.code=\def\docshots@log{log},
45   nocrop/.code=\def\docshots@nocrop{},
46   lstinputlisting/.store in=\docshots@lstinputlisting,
47   inputminted/.store in=\docshots@inputminted,
48   tmpdir/.store in=\docshots@tmpdir,
49   tmpdir/.default=_docshots\ifxetex-xe\else\ifluatex-lua\fi\fi,
50   small/.store in=\docshots@small,
51   tiny/.store in=\docshots@tiny,
52   runs/.store in=\docshots@runs,
53   runs/.default=1,
54   pdflatex/.store in=\docshots@pdflatex,
55   pdflatex/.default=pdflatex,
56   gs/.store in=\docshots@gs,
57   gs/.default=\docshots@gsdefault,
58   pdfcrop/.store in=\docshots@pdfcrop,
59   pdfcrop/.default=pdfcrop,
60   margin/.store in=\docshots@margin,
61   margin/.default=5,
62   hspace/.store in=\docshots@hspace,
63   hspace/.default=.05\linewidth,
64   left/.store in=\docshots@left,
65   left/.default=.3\linewidth,
66   right/.store in=\docshots@right,
67   right/.default=.55\linewidth,
68   snippet/.store in=\docshots@snippet,
69   snippet/.default=right,
70   tmpdir,pdflatex,gs,pdfcrop,margin,hspace,left,right,snippet,runs
71 }
72 \ProcessPgfOptions{/docshots}

```

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

```

73 \def\docshots@snippetleft{left}
74 \def\docshots@snippetright{right}
75 \def\docshots@snippetnone{none}

```

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

```

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

```

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

```

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

```

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

```

80 \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:

```
81 \iexec[null]{perl -MFile::Path=make_path -e "make_path(shift)"
82 "\docshots@tmpdir/\jobname"}%
```

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

```
83 \RequirePackage{pdftexcmds}
84 \makeatletter
85 \newcommand\docshots@mdfive[1]{\pdf@filemdfivesum{#1}}
86 \makeatother
```

`\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:

```
87 \makeatletter
88 \newcommand\docshots@prereq[1]{%
89 \iexec[\docshots@log,quiet]{perl -MFile::Copy -e "copy(shift,shift) or die q(cannot copy)"
90 "#1" "\docshots@tmpdir/\jobname/#1"}%
91 \message{docshots: File '#1' copied to
92 '\docshots@tmpdir/\jobname/#1'^^J}%
93 }
```

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:

```
94 \pgfkeys{
95 /docshots/shot/.cd,
96 snippet/.store in=\docshots@thissnippet,
97 options/.store in=\docshots@opts,
98 after/.code=\edef\docshots@after{\detokenize{#1}},
99 prerequisite/.code=\docshots@prereq{#1},
100 }
```

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:

```
101 \def\docshots@shot{%
102 \catcode'\^M=\active%
103 \@ifnextchar[%
104 {\catcode'\^M=5 \docshots@grab}%
105 {\catcode'\^M=5 \docshots@begin}}
106 \def\docshots@grab[#1]{\pgfkeys{/docshots/shot/.cd,#1}\docshots@begin}
107 \def\docshots@begin{\VerbatimEnvironment\begin\VerbatimOut}
108 {\docshots@tmpdir/\jobname/docshots-temp.tex}}
109 \makeatother
```

`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:

```
110 \makeatletter
```

```

111 \newenvironment{docshot}
112 {\let\docshots@thissnippet\docshots@snippet%
113  \def\docshots@opts{}%
114  \docshots@shot}
115 {\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:

```

116  \ifdefined\docshots@dtx%
117    \iexec[null]{perl -i -pe "s/\\A\\x{25} ?//"}
118    \docshots@tmpdir/\jobname/docshots-temp.tex}%
119  \fi%

```

We calculate MD5 hashsum of the file content:

```

120  \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`:

```

121  \IfFileExists{\docshots@tmpdir/\jobname/\hash.pdf}
122    {\message{docshots: Won't render,
123     the PDF '\docshots@tmpdir/\jobname/\hash.pdf' already exists^^J}}
124    {\iexec[\docshots@log,quiet]{perl -MFile::Copy -e "copy(shift,shift) or die q(cannot copy
125     "\docshots@tmpdir/\jobname/docshots-temp.tex"
126     "\docshots@tmpdir/\jobname/\hash.tex")"}%
127    \message{docshots: rendering at line no. \the\inputlineno^^J}%
128    \foreach \n in {1,...,\docshots@runs}{%
129      \iexec[\docshots@log,
130        stdout=\docshots@tmpdir\docshots@sep\jobname\docshots@sep\hash.stdout,
131        exit=\docshots@tmpdir\docshots@sep\jobname\docshots@sep\hash.ret,
132        quiet,trace]{\docshots@render}%
133      \ifnum\n=1\else%
134        \message{docshots: pdflatex run no.\n^^J}%
135      \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:

```

136    \ifdefined\docshots@after%
137      \begingroup%
138        \docshots@aftercmd{\n}{\hash}{\hash.tex}%
139        \iexec[\docshots@log,quiet]{cd "\docshots@tmpdir\docshots@sep\jobname" &&
140        \docshots@aftersh}%
141      \endgroup%
142    \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:

```

143  \IfFileExists{\docshots@tmpdir/\jobname/\hash.crop.pdf}
144    {\message{docshots: Won't pdfcrop,
145     the PDF '\docshots@tmpdir/\jobname/\hash.crop.pdf'
146     already exists^^J}}
147    {\ifdefined\docshots@nocrop
148      \iexec[\docshots@log,quiet]{perl -MFile::Copy -e "copy(shift,shift) or die q(cannot copy

```

```

149     "\docshots@tmpdir/\jobname/\hash.pdf"
150     "\docshots@tmpdir/\jobname/\hash.crop.pdf"%
151   \else%
152   \iexec[\docshots@log,quiet]{"\docshots@pdfcrop"
153     --gscmd "\docshots@gs"
154     --margins 0
155     "\docshots@tmpdir/\jobname/\hash.pdf"
156     "\docshots@tmpdir/\jobname/\hash.crop.pdf"%
157   \fi}%
158 \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:

```

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

```

According to the `snippet` setting, the source stands to the left of the snapshot, to

its right, or is omitted altogether:

```
197 \ifx\docshots@thissnippet\docshots@snippetnone%
198   \docshots@image%
199 \else\ifx\docshots@thissnippet\docshots@snippetleft%
200   \docshots@code\hspace{\docshots@hspace}\docshots@image%
201 \else\ifx\docshots@thissnippet\docshots@snippetright%
202   \docshots@image\hspace{\docshots@hspace}\docshots@code%
203 \else%
204   \PackageError{docshots}{Unknown snippet
205     '\docshots@thissnippet', use left, right, or none}{}%
206 \fi\fi\fi%
207 \par%
208 \endgroup%
209 }
210 \makeatother
211 \endinput
```

Change History

0.0.1		handling of leading percentage symbols is done right.	9
General: Initial version	5		
0.0.3			
\docshots@prereq: New supplementary command added to copy prerequisite files into the rendering directory.	8	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
0.0.4		0.5.0	
General: Package options "lstinutlisting" and "inputminted" introduced to enable printing of verbatim text via either the listings or the minted package.	6	General: Windows compatibility: file operations go through Perl, and the "gs" default becomes "gswin64c" on Windows.	6
0.0.5		0.6.0	
General: Package option "log" added, which enables detailed logging via exec. By default, there is no logging at all.	6	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
0.1.0			
docshot: The output is saved to a hash-named file, for better uniqueness of temporary files.	9	The environment gains an optional argument that carries the per-docshot settings, replacing the docshotSnippet, docshotOptions, docshotAfter, and docshotPrerequisite commands.	4
0.1.1			
\docshots@mdfive: New supplementary command added to calculate MD5 sum of a file.	8	docshot: The environment gains an optional argument carrying per-docshot settings; the commands docshotSnippet, docshotOptions, docshotAfter, and docshotPrerequisite are removed in its favour.	8
0.2.0			
General: New package option "nocrop" added to allow disabling of "pdfcrop" execution.	6		
0.3.1			
docshot: A bug fixed, now			

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	78, 80, 89, 124,	<code>\dtx</code> 3
<code>\@backslashchar</code> ... 24	129, 139, 148, 152	
<code>\@ifnextchar</code> 103		E
<code>\^</code> 102, 104, 105		<code>\edef</code> 24, 98, 192
A		<code>\end</code> 115, 196
<code>\active</code> 102	<code>\docshots@lstinputlisting</code> 46, 168	<code>\endinput</code> 211
<code>\after</code> 5	<code>\docshots@margin</code> 60, 162	environments:
	<code>\docshots@mdfive</code> 83, 120	<code>docshot</code> 4
	<code>\docshots@nocrop</code> ..	<code>\expandafter</code> 170, 174, 190
	 45, 77, 147	<code>\ExplSyntaxOff</code> 21
B	<code>\docshots@opts</code>	<code>\ExplSyntaxOn</code> 11
<code>\begin</code> 107, 166	.. 97, 113, 169,	
	170, 174, 189, 190	
	<code>\docshots@pdfcrop</code> ..	
	 58, 78, 152	F
C	<code>\docshots@pdflatex</code> .	<code>\FancyVerbHFFuzz</code> ... 192
<code>\catcode</code> .. 102, 104, 105	 30, 36, 54, 76	<code>\foreach</code> 128
<code>\color</code> 179	<code>\docshots@prereq</code> 87, 99	<code>\fvset</code> 178,
<code>\cs</code> 13, 19	<code>\docshots@render</code> ..	179, 180, 181,
	 30, 35, 132	182, 184, 187, 190
D	<code>\docshots@right</code> 66, 166	
<code>\def</code> 23, 26, 27, 30, 35,	<code>\docshots@runs</code> . 52, 128	G
40, 44, 45, 73, 74,	<code>\docshots@sep</code>	<code>\gs</code> 3
75, 101, 106, 107,	24, 27, 130, 131, 139	
113, 120, 161, 166	<code>\docshots@shot</code> 101, 114	H
<code>\detokenize</code> 98	<code>\docshots@small</code> 50, 183	<code>\hash</code> ... 33, 38, 120,
<code>\docshot</code> 110	<code>\docshots@snippet</code> 68, 112	121, 123, 126,
<code>docshot (env.)</code> 4	<code>\docshots@snippetleft</code>	130, 131, 138,
<code>\docshots@after</code> ...	 73, 199	143, 145, 149,
..... 15, 98, 136	<code>\docshots@snippetnone</code>	150, 155, 156,
<code>\docshots@aftercmd</code> .	 75, 197	165, 172, 176, 193
..... 14, 138	<code>\docshots@snippetright</code>	<code>\hspace</code> 3, 200, 202
<code>\docshots@aftersh</code> 19, 140	 74, 201	
<code>\docshots@begin</code> ...	<code>\docshots@thissnippet</code>	I
.... 105, 106, 107	 96, 112,	<code>\iexec</code> ... 76, 78, 80,
<code>\docshots@code</code>	197, 199, 201, 205	81, 89, 117, 124,
.... 166, 200, 202	<code>\docshots@tiny</code> . 51, 186	129, 139, 148, 152
<code>\docshots@dtx</code> .. 43, 116	<code>\docshots@tmpdir</code> ..	<code>\ifdefined</code>
<code>\docshots@grab</code> 104, 106	. 32, 33, 35, 48,	.. 77, 116, 136,
<code>\docshots@gs</code> 56, 80, 153	82, 90, 92, 108,	147, 168, 169,
<code>\docshots@gsdefault</code>	118, 120, 121,	173, 183, 186, 189
..... 23, 26, 57	123, 125, 126,	<code>\ifdocshots@windows</code>
<code>\docshots@hspace</code> ..	130, 131, 139,	 10, 22, 29
..... 62, 200, 202	143, 145, 149,	<code>\IfFileExists</code> . 121, 143
<code>\docshots@image</code> ...	150, 155, 156,	<code>\ifluatex</code> 49
. 161, 198, 200, 202	165, 172, 176, 193	<code>\ifnum</code> 133
<code>\docshots@inputminted</code>	<code>\docshots@windowsfalse</code>	<code>\ifx</code> 197, 199, 201
..... 47, 173, 175	 12	<code>\ifxetex</code> 49
<code>\docshots@left</code> . 64, 164	<code>\docshots@windowstrue</code>	<code>\includegraphics</code> .. 164
<code>\docshots@log</code>	 12	<code>\inputlineno</code> .. 127, 158
.... 40, 44, 76,		

<code>\inputminted</code> 4, 174	<code>\maxdimen</code> 192	R
	<code>\message</code> . . . 91, 122, 127, 134, 144, 158	<code>\RequirePackage</code> 1, 2, 3, 4, 5, 6, 7, 8, 9, 83
J		<code>\right</code> 3
<code>\jobname</code> . 32, 33, 35, 82, 90, 92, 108, 118, 120, 121, 123, 125, 126, 130, 131, 139, 143, 145, 149, 150, 155, 156, 165, 172, 176, 193	N	<code>\runs</code> 4
	<code>\newcommand</code> . . 14, 85, 88	S
	<code>\newenvironment</code> . . . 111	<code>\scriptsize</code> 187
	<code>\newif</code> 10	<code>\small</code> 4, 184
	<code>\nocrop</code> 3	<code>\snippet</code> 3, 4
	<code>\node</code> 162	<code>\str</code> 13, 15, 16, 17, 18, 19
	<code>\number</code> 192	<code>\sys</code> 12
L	O	T
<code>\l</code> 15, 16, 17, 18, 19	<code>\options</code> 5	<code>\the</code> 127, 158
<code>\left</code> 3	P	<code>\tikz</code> 161
<code>\linewidth</code> . . . 63, 65, 67	<code>\PackageError</code> 204	<code>\tiny</code> 4
<code>\log</code> 4	<code>\par</code> 160, 207	<code>\tmpdir</code> 3
<code>\lstinputlisting</code> 4, 172	<code>\pdf@filemdfivesum</code> . 85	V
<code>\lstset</code> 170	<code>\pdfcrop</code> 3	<code>\VerbatimEnvironment</code>
M	<code>\pdflatex</code> 2	 107
<code>\makeatletter</code> 84, 87, 110	<code>\pgfkeys</code> . . . 41, 94, 106	<code>\VerbatimInput</code> 193
<code>\makeatother</code> 86, 109, 210	<code>\prerequisite</code> 5	<code>\vspace</code> 167, 195
<code>\margin</code> 3	<code>\ProcessPgfOptions</code> . 72	