

# iexec: L<sup>A</sup>T<sub>E</sub>X Package for Inputable Shell Executions\*

Yegor Bugayenko  
yegor256@gmail.com

2022-10-21, 0.10.1

## 1 Introduction

This package helps you execute shell commands right from the document and then put their output to the document:

Today is **21-Oct-2022**

```
1 \documentclass{article}
2 \usepackage{iexec}
3 \usepackage[paperwidth=3in]{geometry}
4 \pagestyle{empty}
5 \begin{document}
6 Today is \textbf{%
7   \iexec{date +%e-%b-%Y}}
8 \end{document}
```

**\iexec** The only command provided by this package is `\iexec [options] {cmd}`. Its only mandatory argument *cmd* is the command to be executed through the terminal shell (bash, or whatever is set as the default one on your user console).

You have to run `pdflatex` (or just `latex`) with the `--shell-escape` flag in order to let `shellesc` (the package we use) to execute your shell command.

## 2 Options

**quiet** If you don't want the output to be visible, use `\phantom{\iexec{...}}`. Otherwise, you can use `quiet` option:

```
I just want to delete some file:
\iexec[quiet]{rm -f foo.txt}
```

In this case, whatever the shell command produces will not be included into the document.

**stdout** The output of your code is saved into the file provided as an optional argument of `\iexec` (the default value is `iexec.tmp`):

```
Today is \iexec[stdout=date.txt]{date +%e-%b-%Y | tr -d '\n'}.
```

The tailing part of the command here removes all ends-of-line.

**stderr**     The error output of the code is saved into the file provided as an optional argument of `\iexec` (by default the error output is streamed into `stdout`):

```
Today is \iexec[stderr=my.txt]{broken-command}.
```

**trace**     The file specified will be deleted right after its usage. If you don't want this to happen, use `trace` package option: all files will remain in the directory where they were created. It's possible to turn tracing on globally, for the entire document, using `trace` option of the package:

```
\documentclass{article}
\usepackage[trace]{iexec}
\begin{document}
This file won't be deleted: \iexec[stdout=me.txt]{whoami}.
\end{document}
```

**append**    The `stdout` produced will be appended to the file specified:

```
\documentclass{article}
\usepackage[trace]{iexec}
\begin{document}
\iexec[append,stdout=foo.txt,quiet]{echo 'Hello, '}
\iexec[append,stdout=foo.txt,quiet]{echo 'Jeffrey!'}
\input{foo.txt}
\end{document}
```

**log**       The `stdout` produced will be printed in  $\TeX$  log:

```
\iexec[log]{echo 'Hello, \LaTeX!'}
```

**null**       The `stdout` of the command will be sent to `/dev/null`:

```
\iexec[null]{rm some-file.txt}
```

**ignore**     By default, we report an error if exit code is not equal to zero. You can suppress this with `ignore` option:

```
\iexec[ignore]{broken-command}
```

### 3 Implementation

First, we include [shellesc](#) package, which we use to execute shell commands:

```
1 \RequirePackage{shellesc}
```

Then, we parse package options:

```
2 \RequirePackage{xkeyval}
3 \makeatletter
4 \newif\ifiexec@trace
5 \DeclareOptionX{trace}{\ifiexec@tracetrue}
6 \ProcessOptionsX\relax
7 \makeatother
```

---

\*The sources are in GitHub at [yegor256/iexec](#)

Then, we prepare to parse the options of `\iexec` command:

```

8 \RequirePackage{pgfkeys}
9 \makeatletter\pgfkeys{
10 /iexec/.is family,
11 /iexec,
12 stdout/.store in = \iexec@stdout,
13 stdout/.default = iexec.tmp,
14 stderr/.store in = \iexec@stderr,
15 trace/.store in = \iexec@traceit,
16 append/.store in = \iexec@append,
17 log/.store in = \iexec@log,
18 null/.store in = \iexec@null,
19 quiet/.store in = \iexec@quiet,
20 ignore/.store in = \iexec@ignore,
21 stdout
22 }\makeatother

```

`\iexec@typeout` Then, we define an internal command `\iexec@typeout` for printing the content of a file, as suggested [here](#):

```

23 \RequirePackage{expl3}
24 \makeatletter\ExplSyntaxOn
25 \NewDocumentCommand{\iexec@typeout}{m}{
26 \iexec_typeout_file:n { #1 }}
27 \ior_new:N \g_iexec_typeout_ior
28 \cs_new_protected:Nn \iexec_typeout_file:n
29 {
30 \ior_open:Nn \g_iexec_typeout_ior { #1 }
31 \ior_str_map_inline:Nn \g_iexec_typeout_ior
32 {\iow_term:n { ##1 }}
33 \ior_close:N \g_iexec_typeout_ior
34 }
35 \ExplSyntaxOff\makeatother

```

`\iexec` Then, we define `\iexec` command. It is implemented with the help of `\ShellEscape` from `shellesc` package:

```

36 \makeatletter
37 \newread\iexec@retfile
38 \newcommand\iexec[2] [] {%
39 \begingroup%
40 \pgfqkeys{/iexec}{#1}%

```

First, we verify that latex is running with `--shell-escape` option, since without it nothing will work; so, it's better to throw an error earlier than later:

```

41 \ifnum\ShellEscapeStatus=1\else%
42 \PackageError{iexec}{You must run TeX processor with
43 --shell-escape option}{}%
44 \fi%
45 \begingroup%

```

Then, start the log from a clean line:

```

46 \ifdefined\iexec@log%
47 \message{^^J}%
48 \fi%

```

Then, we define a few special chars in order to escape them in the shell (the full list of them is in [macros2e](#)):

```
49      \let%\@percentchar%
50      \let\\ \@backslashchar%
51      \let\{\@charlb%
52      \let\}\@charrb%
```

Then, we execute it:

```
53      \def\iexec@cmd{(#2)
54      \ifdefined\iexec@append>\fi>
55      \ifdefined\iexec@null/dev/null\else\iexec@stdout\fi
56      \space\ifdefined\iexec@stderr2>\iexec@stderr\else2>&1\fi
57      \ifdefined\iexec@ignore\else; echo $? >iexec.ret\fi}
58      \ShellEscape{\iexec@cmd}%
```

Then, a message is printed to TeX log:

```
59      \ifdefined\iexec@log%
60      \message{iexec: [\iexec@cmd]^^J}%
61      \fi%
62      \endgroup%
```

Then, if required, the content of the stdout file will be printed to the log:

```
63      \ifdefined\iexec@null\else%
64      \ifdefined\iexec@log%
65      \message{iexec: This is the content of \iexec@stdout:^^J}%
66      \iexec@typeout{\iexec@stdout}%
67      \message{<EOF>^^J}%
68      \fi\fi%
```

Then, we check exit code:

```
69      \ifdefined\iexec@log\else%
70      \immediate\openin\iexec@retfile=iexec.ret%
71      \read\iexec@retfile to \iexec@ret%
72      \immediate\closein\iexec@retfile%
73      \ifnum\iexec@ret=0\else%
74      \PackageError{iexec}{Exit code is \iexec@ret}{}%
75      \fi%
76      \fi%
```

Then, include the produced output into the current document:

```
77      \ifdefined\iexec@null\else%
78      \ifdefined\iexec@quiet%
79      \ifdefined\iexec@log%
80      \message{iexec: Due to 'quiet' option we didn't read
81      the content of '\iexec@stdout'
82      \ifdefined\pdffilesize(\pdffilesize{\iexec@stdout}
83      bytes)\fi^^J}%
84      \fi%
85      \else%
86      \ifdefined\iexec@log%
87      \message{iexec: We include the content of
88      '\iexec@stdout'\ifdefined\pdffilesize(\pdffilesize
89      {\iexec@stdout} bytes)\fi...^^J}%
90      \fi%
91      \input{\iexec@stdout}%
92      \message{iexec: The content of '\iexec@stdout'}
```

```

93     was included into the document^^J}%
94 \fi\fi%

```

Finally, delete the file or leave it untouched:

```

95 \ifdefined\iexec@null\else%
96 \ifiexec@trace%
97 \ifdefined\iexec@log%
98 \message{iexec: Due to package option 'trace',
99 the file '\iexec@stdout' was not deleted^^J}%
100 \fi%
101 \else%
102 \ifdefined\iexec@traceit%
103 \ifdefined\iexec@log%
104 \message{iexec: Due to 'trace' option,
105 the file '\iexec@stdout' was not deleted^^J}%
106 \fi%
107 \else%
108 \ShellEscape{rm \iexec@stdout}%
109 \ifdefined\iexec@log%
110 \message{iexec: The file '\iexec@stdout' was deleted^^J}%
111 \fi%
112 \fi%
113 \fi\fi%
114 \endgroup
115 }\makeatother

```

# Change History

v0.10.0	General: The option "ignore" suppresses the checking of "iexec.ret" value. . . . . 3	turn on log/debug messages in TeX log (they were all visible always, which was sometimes annoying. Also, this option enables printing of the entire content of stdout to the log too (this may be pretty convenient for debugging). . . . . 3
	\iexec: The ability to track exit code was added. Now, the code is saved into "iexec.ret" file, which is then read and checked for zero value. . . 4	
	The file "iexec.ret" is reused for all scripts. . . . . 3	v0.8.0
v0.7.0	General: The option "append" was introduced — if it's turned on, stdout will be appended to the file, instead of rewriting it (this is how it was before). . . . . 3	\iexec: The option "null" was introduced, allowing redirection of stdout to "/dev/null". Doesn't work on Windows, though. . . . . 4
	The option "log" was introduced, to	v0.9.0
		\iexec: The option "stderr" was introduced, allowing redirection of stderr to a file. Without this option specified, stderr will go to stdout. . 4

# 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>Symbols</b>		<b>\iexec@ignore</b> . . . . 20, 57	<b>\makeatother</b> 7, 22, 35, 115
<b>\%</b> . . . . .	49	<b>\iexec@log</b> 17, 46, 59, 64,	<b>\message</b> 47, 60, 65, 67,
<b>\@backslashchar</b> . . . .	50	69, 79, 86, 97, 103, 109	80, 87, 92, 98, 104, 110
<b>\@charlb</b> . . . . .	51	<b>\iexec@null</b> . . . . .	
<b>\@charrb</b> . . . . .	52	. . . 18, 55, 63, 77, 95	<b>N</b>
<b>\@percentchar</b> . . . . .	49	<b>\iexec@quiet</b> . . . . 19, 78	<b>\NewDocumentCommand</b> . 25
<b>\\</b> . . . . .	50	<b>\iexec@ret</b> . . . . 71, 73, 74	<b>\newif</b> . . . . . 4
<b>\{</b> . . . . .	51	<b>\iexec@retfile</b> . . . .	<b>\newread</b> . . . . . 37
<b>\}</b> . . . . .	52	. . . . 37, 70, 71, 72	
		<b>\iexec@stderr</b> . . . . 14, 56	<b>O</b>
<b>C</b>		<b>\iexec@stdout</b> 12, 55, 65,	<b>\openin</b> . . . . . 70
<b>\closein</b> . . . . .	72	66, 81, 82, 88, 89,	
<b>\cs</b> . . . . .	28	91, 92, 99, 105, 108, 110	<b>P</b>
		<b>\iexec@traceit</b> . . 15, 102	<b>\PackageError</b> . . . . 42, 74
<b>D</b>		<b>\iexec@tracetrue</b> . . . . 5	<b>\pdffilesize</b> . . . . 82, 88
<b>\DeclareOptionX</b> . . . . .	5	<b>\iexec@typeout</b> . . . 23, 66	<b>\pgfkeys</b> . . . . . 9
<b>\def</b> . . . . .	53	<b>\ifdefined</b> 46, 54, 55, 56,	<b>\pgfqkeys</b> . . . . . 40
		57, 59, 63, 64, 69,	<b>\ProcessOptionsX</b> . . . . 6
<b>E</b>		77, 78, 79, 82, 86,	
<b>\ExplSyntaxOff</b> . . . . .	35	88, 95, 97, 102, 103, 109	<b>R</b>
<b>\ExplSyntaxOn</b> . . . . .	24	<b>\ifiexec@trace</b> . . . . 4, 96	<b>\read</b> . . . . . 71
		<b>\ifnum</b> . . . . . 41, 73	<b>\relax</b> . . . . . 6
<b>G</b>		<b>\immediate</b> . . . . . 70, 72	<b>\RequirePackage</b> 1, 2, 8, 23
<b>\g</b> . . . . .	27, 30, 31, 33	<b>\input</b> . . . . . 91	
		<b>\ior</b> . . . . . 27, 30, 31, 33	<b>S</b>
<b>I</b>		<b>\iow</b> . . . . . 32	<b>\ShellEscape</b> . . . . 58, 108
<b>\iexec</b> . . . . .	26, 28, 36		<b>\ShellEscapeStatus</b> . . 41
<b>\iexec@append</b> . . . . 16, 54		<b>M</b>	<b>\space</b> . . . . . 56
<b>\iexec@cmd</b> . . . . 53, 58, 60		<b>\makeatletter</b> . 3, 9, 24, 36	