

iexec: L^AT_EX Package for Inputable Shell Executions*

Yegor Bugayenko
yegor256@gmail.com

2022-10-12, 0.9.0

1 Introduction

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

```
\documentclass{article}
\usepackage{iexec}
\begin{document}
Today is
  \iexec{date +%e-%b-%Y}
\end{document}
```

Today is 12-Oct-2022

`\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:

```
\documentclass{article}
\usepackage{iexec}
\begin{document}
I just want to delete some file:
\iexec[quiet]{rm -f foo.txt}
\end{document}
```

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`):

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

```

\documentclass{article}
\usepackage{iexec}
\begin{document}
Today is \iexec[stdout=date.txt]{date +%e-%b-%Y | tr -d '\n'}.
\end{document}

```

The trailing 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`):

```

\documentclass{article}
\usepackage{iexec}
\begin{document}
Today is \iexec[stderr=my.txt]{broken-command}.
\end{document}

```

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:

```

\documentclass{article}
\usepackage{iexec}
\begin{document}
\iexec[log]{echo 'Hello, \LaTeX!'}
\end{document}

```

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

```

\documentclass{article}
\usepackage{iexec}
\begin{document}
\iexec[null]{rm some-file.txt}
\end{document}

```

3 Implementation

First, we include a few packages:

```
1 \RequirePackage{shellesc}
2 \RequirePackage{pgfkeys}
3 \RequirePackage{expl3}
```

Then, we parse package options:

```
4 \RequirePackage{xkeyval}
5 \makeatletter\newif\ifiexec@trace
6 \DeclareOptionX{trace}{\ifiexec@tracetrue}
7 \ProcessOptionsX\relax\makeatother
8 \makeatletter\pgfkeys{
9   /iexec/.is family,
10  /iexec,
11   stdout/.estore in = \ifiexec@stdout,
12   stdout/.default = iexec.tmp,
13   stderr/.estore in = \ifiexec@stderr,
14   trace/.estore in = \ifiexec@traceit,
15   append/.estore in = \ifiexec@append,
16   log/.estore in = \ifiexec@log,
17   null/.estore in = \ifiexec@null,
18   quiet/.estore in = \ifiexec@quiet,
19   stdout
20 }\makeatother
```

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

```
21 \makeatletter\ExplSyntaxOn
22 \NewDocumentCommand{\iexec@typeout}{m}{
23   \iexec_typeout_file:n { #1 }
24   \ior_new:N \g_iexec_typeout_ior
25   \cs_new_protected:Nn \iexec_typeout_file:n
26   {
27     \ior_open:Nn \g_iexec_typeout_ior { #1 }
28     \ior_str_map_inline:Nn \g_iexec_typeout_ior
29     { \ior_term:n { ##1 } }
30     \ior_close:N \g_iexec_typeout_ior
31   }
32 \ExplSyntaxOff\makeatother
```

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

```
33 \makeatletter\newcommand\iexec[2] [] {%
34   \begingroup
35     \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:

```
36   \ifdefined\pdfshellescape\ifnum\pdfshellescape=1\else%
37     \PackageError{iexec}{You must run TeX processor with
38       --shell-escape option}{}%
39   \fi\fi%
40   \begingroup%
```

Then, start the log from a clean line:

```
41 \message{^^J}%
```

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

```
42 \let%\@percentchar%
43 \let\\ \@backslashchar%
44 \let\{\@charlb%
45 \let\}\@charrb%
```

Then, we execute it:

```
46 \def\iexec@cmd{(#2)
47 \ifdefined\iexec@append>\fi>
48 \ifdefined\iexec@null/dev/null\else\iexec@stdout\fi
49 \space\ifdefined\iexec@stderr2>\iexec@stderr\else2>&1\fi}
50 \ShellEscape{\iexec@cmd}%
```

Then, a message is printed to TeX log:

```
51 \message{iexec: [\iexec@cmd]^^J}%
52 \endgroup%
```

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

```
53 \ifdefined\iexec@null\else
54 \ifdefined\iexec@log
55 \message{iexec: this is the content of \iexec@stdout:^^J}%
56 \iexec@typeout{\iexec@stdout}
57 \message{<EOF>^^J}%
58 \fi\fi%
```

Then, include the produced output into the current document:

```
59 \ifdefined\iexec@null\else
60 \ifdefined\iexec@quiet
61 \message{iexec: Due to 'quiet' option we didn't read
62 the content of '\iexec@stdout' (\pdffilesize{} bytes)^^J}%
63 \else%
64 \message{iexec: We include the content of
65 '\iexec@stdout' (\pdffilesize{} bytes)...^^J}%
66 \input{\iexec@stdout}%
67 \message{iexec: The content of '\iexec@stdout'
68 was included into the document^^J}%
69 \fi\fi%
```

Finally, delete the file or leave it untouched:

```
70 \ifdefined\iexec@null\else
71 \ifiexec@trace%
72 \message{iexec: Due to package option 'trace',
73 the file '\iexec@stdout' was not deleted^^J}%
74 \else%
75 \ifdefined\iexec@traceit%
76 \message{iexec: Due to 'trace' option,
77 the file '\iexec@stdout' was not deleted^^J}%
78 \else%
79 \ShellEscape{rm \iexec@stdout}%
80 \message{iexec: The file '\iexec@stdout' was deleted^^J}%
81 \fi%
82 \fi\fi%
```

```
83 \endgroup
84 }\makeatother
```

Change History

v0.2.0		v0.6.0	
General: Initial version	3	General: Additional options 'log' and 'append'	3
v0.4.0		v0.7.0	
General: Runtime verification for -shell-escape option	3	General: Additional option 'null'	3
v0.4.1		v0.8.0	
General: Options trace, stdout, and quiet added	3	General: Bug fixes	3
v0.5.0		v0.9.0	
General: Major bug fixes	3	General: Additional option 'stderr' . . .	3

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			
\%	42	\iexec@append	15, 47
\@backslashchar	43	\iexec@cmd	46, 50, 51
\@charlb	44	\iexec@log	16, 54
\@charrb	45	\iexec@null	
\@percentchar	42		17, 48, 53, 59, 70
\\	43	\iexec@quiet	18, 60
\{	44	\iexec@stderr	13, 49
\}	45	\iexec@stdout	11,
			48, 55, 56, 62, 65,
			66, 67, 73, 77, 79, 80
C		\iexec@traceit	14, 75
\cs	25	\iexec@tracetrue	6
		\iexec@typeout	21, 56
D		\ifdefined	36, 47, 48, 49,
\DeclareOptionX	6		53, 54, 59, 60, 70, 75
\def	46	\ifiexec@trace	5, 71
		\ifnum	36
E		\input	66
\ExplSyntaxOff	32	\ior	24, 27, 28, 30
\ExplSyntaxOn	21	\iow	29
G			
\g	24, 27, 28, 30		
		M	
I		\makeatletter	5, 8, 21, 33
\iexec	23, 25, <u>33</u>	\makeatother	7, 20, 32, 84
		S	
		\message	41, 51, 55, 57,
			61, 64, 67, 72, 76, 80
		N	
		\NewDocumentCommand	22
		\newif	5
		P	
		\PackageError	37
		\pdffilesize	62, 65
		\pdfshellescape	36
		\pgfkeys	8
		\pgfqkeys	35
		\ProcessOptionsX	7
		R	
		\relax	7
		\RequirePackage	1, 2, 3, 4