

iexec: L^AT_EX Package for Inputable Shell Executions*

Yegor Bugayenko
yegor256@gmail.com

2022-10-15, 0.9.1

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 15-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 [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
```

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 stdout
21 }\makeatother
```

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

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

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

```
35 \makeatletter\newcommand\iexec[2] [] {%
36 \begingroup
37 \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:

```
38 \ifnum\ShellEscapeStatus=1\else%
```

```

39     \PackageError{iexec}{You must run TeX processor with
40     --shell-escape option}{}%
41 \fi%
42 \begingroup%

```

Then, start the log from a clean line:

```

43     \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](#)):

```

44     \let\%\@percentchar%
45     \let\\\@backslashchar%
46     \let\{\@charlb%
47     \let\}\@charrb%

```

Then, we execute it:

```

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

```

Then, a message is printed to TeX log:

```

53     \message{iexec: [\iexec@cmd]^^J}%
54 \endgroup%

```

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

```

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

```

Then, include the produced output into the current document:

```

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

```

Finally, delete the file or leave it untouched:

```

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

```

```
81         \ShellEscape{rm \iexec@stdout}%  
82         \message{iexec: The file '\iexec@stdout' was deleted^^J}%  
83         \fi%  
84     \fi\fi%  
85 \endgroup  
86 }\makeatother
```

Change History

v0.2.0		v0.6.0	
General: Initial version	3	General: Additional options 'log' and	
v0.4.0		'append'	3
General: Runtime verification for		v0.7.0	
-shell-escape option	3	General: Additional option 'null'	3
v0.4.1		v0.8.0	
General: Options trace, stdout, and		General: Bug fixes	3
quiet added	3	v0.9.0	
v0.5.0		General: Additional option 'stderr' . . .	3
General: Major bug fixes	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			
\%	44	\iexec@append	16, 49
\@backslashchar	45	\iexec@cmd	48, 52, 53
\@charlb	46	\iexec@log	17, 56
\@charrb	47	\iexec@null	
\@percentchar	44		18, 50, 55, 61, 72
\\	45	\iexec@quiet	19, 62
\{	46	\iexec@stderr	14, 51
\}	47	\iexec@stdout . . .	12,
			50, 57, 58, 64, 67,
			68, 69, 75, 79, 81, 82
C		\iexec@traceit . . .	15, 77
\cs	27	\iexec@tracetrue	5
		\iexec@typeout . . .	22, 58
D		\ifdefined . .	49, 50, 51,
\DeclareOptionX	5		55, 56, 61, 62, 72, 77
\def	48	\ifiexec@trace	4, 73
		\ifnum	38
E		\input	68
\ExplSyntaxOff	34	\ior	26, 29, 30, 32
\ExplSyntaxOn	23	\iow	31
G			
\g	26, 29, 30, 32		
		M	
I		\makeatletter .	3, 9, 23, 35
\iexec	25, 27, <u>35</u>	\makeatother .	7, 21, 34, 86
		\message	43, 53, 57, 59,
			63, 66, 69, 74, 78, 82
		N	
		\NewDocumentCommand .	24
		\newif	4
		P	
		\PackageError	39
		\pdffilesize	64, 67
		\pgfkeys	9
		\pgfqkeys	37
		\ProcessOptionsX	6
		R	
		\relax	6
		\RequirePackage	1, 2, 8, 22
		S	
		\ShellEscape	52, 81
		\ShellEscapeStatus . .	38
		\space	51