

iexec: L^AT_EX Package for Inputable Shell Executions*

Yegor Bugayenko
yegor256@gmail.com

2022-10-10, 0.8.1

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 10-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.

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

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

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

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

`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 = \iexec@stdout,
12  stdout/.default = iexec.tmp,
13  trace/.estore in = \iexec@traceit,
14  append/.estore in = \iexec@append,
15  log/.estore in = \iexec@log,
16  null/.estore in = \iexec@null,
17  quiet/.estore in = \iexec@quiet,
18  stdout
19 }\makeatother

```

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

```

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

```

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

```

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

```

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

```

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

```

40   \let\%\@percentchar%
41   \let\\\@backslashchar%
42   \let\{\@charlb%
43   \let\}\@charrb%

```

Then, we execute it:

```
44 \ShellEscape{#2 \ifdefined\iexec@append>\fi>\ifdefined\iexec@null/dev/null\else\iexec@st
```

Then, a message is printed to TeX log:

```
45 \message{^^Jiexec: [#2 \ifdefined\iexec@append>\fi>\ifdefined\iexec@null/dev/null\else\i
46 \endgroup%
```

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

```
47 \ifdefined\iexec@null\else
48 \ifdefined\iexec@log
49 \message{^^Jiexec: this is the content of \iexec@stdout:^^J}%
50 \iexec@typeout{\iexec@stdout}
51 \message{^^J<EOF>^^J}%
52 \fi\fi%
```

Then, include the produced output into the current document:

```
53 \ifdefined\iexec@null\else
54 \ifdefined\iexec@quiet
55 \message{^^Jiexec: Due to 'quiet' option we didn't read
56 the content of '\iexec@stdout' (\pdffilesize{} bytes)^^J}%
57 \else%
58 \message{^^Jiexec: We include the content of
59 '\iexec@stdout' (\pdffilesize{} bytes)...^^J}%
60 \input{\iexec@stdout}%
61 \message{^^Jiexec: The content of '\iexec@stdout'
62 was included into the document^^J}%
63 \fi\fi%
```

Finally, delete the file or leave it untouched:

```
64 \ifdefined\iexec@null\else
65 \ifiexec@trace%
66 \message{^^Jiexec: Due to package option 'trace',
67 the file '\iexec@stdout' was not deleted^^J}%
68 \else%
69 \ifdefined\iexec@traceit%
70 \message{^^Jiexec: Due to 'trace' option,
71 the file '\iexec@stdout' was not deleted^^J}%
72 \else%
73 \ShellEscape{rm \iexec@stdout}%
74 \message{^^Jiexec: The file '\iexec@stdout' was deleted^^J}%
75 \fi%
76 \fi\fi%
77 \endgroup
78 }\makeatother
```

Change History

v0.2.0		v0.5.0
General: Initial version	2	General: Major bug fixes 2
v0.4.0		v0.6.0
General: Runtime verification for		General: Additional options 'log' and
-shell-escape option	2	'append' 2
v0.4.1		v0.7.0
General: Options trace, stdout, and		General: Additional option 'null' 2
quiet added	2	

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			
\%	40	\iexec@append	14, 44, 45
\@backslashchar	41	\iexec@log	15, 48
\@charlb	42	\iexec@null	16, 44, 45, 47, 53, 64
\@charrb	43	\iexec@quiet	17, 54
\@percentchar	40	\iexec@stdout	11, 44,
\\	41		45, 49, 50, 56, 59,
\{	42		60, 61, 67, 71, 73, 74
\}	43	\iexec@traceit	13, 69
		\iexec@tracetrue	6
A		\iexec@typeout	20, 50
\append	1	\ifdefined	35, 44, 45,
			47, 48, 53, 54, 64, 69
C		\ifiexec@trace	5, 65
\cs	24	\ifnum	35
		\input	60
D		\ior	23, 26, 27, 29
\DeclareOptionX	6	\iow	28
E		L	
\ExplSyntaxOff	31	\log	1
\ExplSyntaxOn	20		
		M	
G		\makeatletter	5, 8, 20, 32
\g	23, 26, 27, 29	\makeatother	7, 19, 31, 78
		\message	45, 49, 51,
I			55, 58, 61, 66, 70, 74
\iexec	1, 22, 24, 32		
		N	
		\NewDocumentCommand	21
		\newif	5
		\null	1
		P	
		\PackageError	36
		\pdffilesize	56, 59
		\pdfshellescape	35
		\pgfkeys	8
		\pgfqkeys	34
		\ProcessOptionsX	7
		Q	
		\quiet	1
		R	
		\relax	7
		\RequirePackage	1, 2, 3, 4
		S	
		\ShellEscape	44, 73
		\stdout	1
		T	
		\trace	1