

# The shellesc Package\*

LaTeX3 project

2016/06/07

## 1 Introduction

For many years web2c based  $\TeX$  implementations have used the syntax of the `\write` command to access system commands by using a special stream 18 (streams above 15 can not be allocated to files in classical  $\TeX$  so stream 18 would otherwise just print to the terminal).

This is a useful extension that did not break the strict rules on extensions in classical  $\TeX$ . This package provides a simple macro level interface hiding the `write18` implementation so a command to remove a file on a unix-like system could be specified using `\ShellEscape{rm file.txt}` (or `del` in windows). Note that by default system access is not allowed and latex will typically need to be called with the `--shell-escape` command line option.

The package may be used with standard `latex` or `pdflatex` or `xetex`, however it is mostly motivated by `lualatex` as from Lua $\TeX$  0.87 onwards Lua $\TeX$  does *not* support the `\write18` syntax to access system commands: it has 256 write streams and stream 18 can be associated to a file and (without this package) has no special significance. This package defines the same `\ShellEscape` syntax in Lua $\TeX$ , but the implementation is via Lua and the `os.execute` function.

`\ShellEscape` in fact corresponds to `\immediate\write18` (or `\directlua`). Very rarely you may need to delay a system command until the current page is output (when page numbers are known) for this classically you could use `\write18` (or `\latelua`). This package provides `\DelayedShellEscape` as a common syntax for this use.

To aid porting existing documents to Lua $\TeX$  0.87 this package does overload the `\write` command so that `\write18{rm file.txt}` will work with Lua $\TeX$ . Note that the redefinition of `\write` can not detect whether `\immediate` has been used, `\immediate` will work as normal when writing to file streams or the terminal but the special case of stream 18 which is defined to use `os.execute` always uses `\directlua` (so corresponds to `\immediate\write18`). In the rare situations that you need non-immediate `\write18` in a document being ported to current Lua $\TeX$ , you will need to change to use the `\DelayedShellEscape` command.

## 2 Implementation

1 `\package`

---

\*This file has version number v0.02b, last revised 2016/06/07.

## 2.1 Status Check

```
2 \ifcase
3   \ifx\pdfshellescape\@undefined
4     \ifx\shellescape\@undefined
5       \ifx\directlua\@undefined
6         \z@
7       \else
8         \directlua{%
9           tex.sprint((status.shell_escape or os.execute()) .. " ")}
10      \fi
11     \else
12       \shellescape
13     \fi
14   \else
15     \pdfshellescape
16   \fi
17 \PackageWarning{shellesc}{Shell escape disabled}
18 \or
19 \PackageInfo {shellesc}{Unrestricted shell escape enabled}
20 \else
21 \PackageInfo {shellesc}{Restricted shell escape enabled}
22 \fi
```

## 2.2 The shellesc package interface

`\ShellEscape` Execute the supplied tokens as a system dependent command, assuming such execution is allowed.

```
23 \ifx\lastsavedimageresourcepages\@undefined
24   \protected\def\ShellEscape{\immediate\write18 }
25 \else
26   \protected\def\ShellEscape#1{%
27     \directlua{os.execute("\luaescapestring{#1}")}}
28 \fi
```

`\DelayedShellEscape` Execute the supplied tokens as a system dependent command, when this node is shipped out with the completed page, assuming such execution is allowed.

```
29 \ifx\lastsavedimageresourcepages\@undefined
30   \protected\def\DelayedShellEscape{\relax\write18 }
31 \else
32   \protected\def\DelayedShellEscape#1{%
33     \lattelua{os.execute("\luaescapestring{#1}")}}
34 \fi
```

## 2.3 The write18 package interface

In web2c based engines other than LuaTeX, `\write18` may be used directly. The same was true in older LuaTeX, but from version 0.85 onwards that is not available.

The above `shellesc` package interface is recommended for new code, however for ease of porting existing documents and packages to newer LuaTeX releases, a `\write18` interface is provided here via a call to Lua's `os.execute`.

Note that as currently written this always does an *immediate* call to the system.

`\immediate` is supported but ignored, `\immediate\write18` and `\write18` both execute immediately. To use a delayed execution at the next shipout, use the `\DelayedShellEscape` command defined above.

Note that it would be easy to make `\wriete18` defined here use delayed execution, just use `\DelayedShellEscape` instead of `ShellEscape` in the definition below. However detecting `\immediate` is tricky so the choice here is to always use the immediate form, which is overwhelmingly more commonly used with `\write18`.

Stop at this point if not a recent Lua<sub>TEX</sub>.

```
35 \ifx\lastsavedimageresourcepages\@undefined\expandafter\endinput\fi
36 \directlua{%
```

```
37 shellesc = shellesc or {}
```

Lua function to use the token scanner to grab the following <sub>TEX</sub> number, and then test if stream 18 is being used, and then insert an appropriate <sub>TEX</sub> command to handle the following brace group in each case.

```
38 local function write_or_execute()
39   local s = token.scan_int()
40   if (s==18) then
41     tex.sprint(\the\numexpr\catcodetable@atletter\relax,
42               "\string\ShellEscape ")
43   else
44     tex.sprint(\the\numexpr\catcodetable@atletter\relax,
45               "\string\shellesc@write " .. s)
46   end
47 end

48 shellesc.write_or_execute=write_or_execute
49 }

50 \let\shellesc@write\write
51 \protected\def\write{\directlua{shellesc.write_or_execute()}}
52 \endpackage
```