# The shellesc Package*

LaTeX project

2023/07/08

## 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 LuaTeX 0.87 onwards LuaTeX 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 packge defines the same `\ShellEscape` syntax in LuaLaTeX, 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 you could classically use `\write18` (or (`\latelua`). This package provides `\DelayedShellEscape` as a common syntax for this use.

The shell escape status may be queried by checking the integer (chardef) command `\ShellEscapeStatus`, 0 (disabled) 1 (enabled) 2 (restricted).

To aid porting existing documents to LuaTeX 0.87 this package does overload the `\write` command so that `\write18{rm file.txt}` will work with LuaTeX. 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 LuaTeX, you will need to change to use the `\DelayedShellEscape` command.

## 2 Implementation

1 ⟨*package⟩

*This file has version number v1.0d, last revised 2023/07/08.

```
\chardef\shellesc@quotecat\catcode`\"
\chardef\shellesc@underscorecat\catcode`\_
\@makeother\"
\@makeother\_
```

## 2.1 Status Check

## 2.2 The shellesc package interface

`\ShellEscapeStatus` Integer value with meanings 0 (shell escape disabled), 1 (shell escape allowed), 2 (restricted shell escape).

```
\chardef\ShellEscapeStatus
  \ifx\pdfshellescape\@undefined
    \ifx\shellescape\@undefined
      \ifx\directlua\@undefined
        \z@
      \else
        \directlua{%
          tex.sprint((status.shell_escape or os.execute()) .. " ")}
      \fi
    \else
      \shellescape
    \fi
  \else
    \pdfshellescape
  \fi

\ifcase\ShellEscapeStatus
  \PackageWarning{shellesc}{Shell escape disabled}
\or
  \PackageInfo   {shellesc}{Unrestricted shell escape enabled}
\else
  \PackageInfo   {shellesc}{Restricted shell escape enabled}
\fi
```

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

```
\ifx\lastsavedimageresourcepages\@undefined
  \protected\def\ShellEscape{\immediate\write18 }

\else
  \protected\def\ShellEscape{\directlua\ShellEscape@Lua}
\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.

```
\ifx\lastsavedimageresourcepages\@undefined
  \protected\def\DelayedShellEscape{\relax\write18 }

\else
  \protected\def\DelayedShellEscape{\latelua\ShellEscape@Lua}
\fi
```

`\ShellEscape@Lua` Shared Lua code for `\DelayedShellEscape` and `\ShellEscape`.

```
\ifx\directlua\@undefined\else
```

```
\protected\def\ShellEscape@Lua#1{{%
local status, msg = os.execute("\luaescapestring{#1}")%
if status == nil then
    texio.write_nl("log",%
      "runsystem(" .. "\luaescapestring{#1}"%
       .. ")...(" .. msg .. ").\string\n")
  elseif status == 0 then
    texio.write_nl("log",%
      "runsystem(" .. "\luaescapestring{#1}"%
      .. ")...executed.\string\n")
  else
    texio.write_nl("log",%
      "runsystem(" .. "\luaescapestring{#1}"%
       .. ")...failed. " .. (msg or "") .. "\string\n")
  end
}}
\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 LuaTEX.

```
\ifx\lastsavedimageresourcepages\@undefined
 \catcode`\"\shellesc@quotecat
 \catcode`\_\shellesc@underscorecat
 \expandafter\endinput
\fi
\directlua{%
shellesc = shellesc or {}
```

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

```
local function write_or_execute()
  local s = token.scan_int()
  if (s==18) then
     tex.sprint(\the\numexpr\catcodetable@atletter\relax,
                "\string\\ShellEscape ")
  else
     tex.sprint(\the\numexpr\catcodetable@atletter\relax,
```

```
                "\string\\shellesc@write " .. s)
  end
end
shellesc.write_or_execute=write_or_execute
}
\let\shellesc@write\write
\protected\def\write{\directlua{shellesc.write_or_execute()}}
\catcode`\"\shellesc@quotecat
\catcode`\_\shellesc@underscorecat
```
⟨/package⟩