

Package ‘scriptexec’

November 25, 2017

Title Execute Native Scripts

Version 0.2.1

Description Run complex native scripts with a single command, similar to system commands.

License Apache License 2.0

URL <https://github.com/sagiegurari/scriptexec>

BugReports <https://github.com/sagiegurari/scriptexec/issues>

Depends R (>= 3.2.3)

Encoding UTF-8

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Sagie Gur-Ari [aut, cre]

Maintainer Sagie Gur-Ari <sagiegurari@gmail.com>

Repository CRAN

Date/Publication 2017-11-25 17:02:49 UTC

R topics documented:

create_script_file	2
execute	2
generate_env_setup_script	3
get_command	4
get_platform_value	5
is_windows	5
modify_script	6
scriptexec	6

Index	7
--------------	----------

`create_script_file` *Creates a temporary file, writes the provided script content into it and returns the file name.*

Description

Creates a temporary file, writes the provided script content into it and returns the file name.

Usage

```
create_script_file(script = "")
```

Arguments

`script` The script text

Value

The temporary file name

Examples

```
filename <- create_script_file('echo test')
```

`execute` *Executes a script and returns the output. The stdout and stderr are captured and returned. In case of errors, the exit code will return in the status field.*

Description

Executes a script and returns the output. The stdout and stderr are captured and returned. In case of errors, the exit code will return in the status field.

Usage

```
execute(script = "", args = c(), env = character(), wait = TRUE,  
        runner = NULL, print_commands = FALSE)
```

Arguments

script	The script text
args	Optional script command line arguments (arguments are added as variables in the script named ARG1, ARG2, ...)
env	Optional character vector of name=value strings to set environment variables
wait	A TRUE/FALSE parameter, indicating whether the function should wait for the command to finish, or run it asynchronously (output status will be -1)
runner	The executable used to invoke the script (by default cmd.exe for windows, sh for other platforms)
print_commands	True if to print each command before invocation (not available for windows)

Value

The process output and status code (in case wait=TRUE) in the form of list(status = status, output = output)

Examples

```
# execute script text
output <- execute('echo Current Directory:\ndir')
cat(sprintf('Exit Status: %s Output: %s\n', output$status, output$output))

# execute multiple commands as a script
output <- execute(c('cd', 'echo User Home:', 'dir'))
cat(sprintf('Exit Status: %s Output: %s\n', output$status, output$output))

# pass arguments (later defined as ARG1, ARG2, ...) and env vars
output <- execute('echo $ARG1 $ARG2', args = c('TEST1', 'TEST2'), env = c('MYENV=TEST3'))
cat(sprintf('%s\n', output))

# non zero status code is returned in case of errors
output <- execute('exit 1')
cat(sprintf('Status: %s\n', output$status))
cat(sprintf('%s\n', output))

#do not wait for command to finish
execute('echo my really long task', wait = FALSE)
```

generate_env_setup_script

Generates and returns a script which sets up the env vars for the script execution.

Description

Generates and returns a script which sets up the env vars for the script execution.

Usage

```
generate_env_setup_script(env = character())
```

Arguments

env Optional character vector of name=value strings to set environment variables

Value

The script text which sets up the env

Examples

```
script <- generate_env_setup_script(c('ENV_TEST=MYENV'))
```

get_command	<i>Returns the command and arguments needed to execute the provided script file on the current platform.</i>
-------------	--

Description

Returns the command and arguments needed to execute the provided script file on the current platform.

Usage

```
get_command(filename, runner = NULL)
```

Arguments

filename The script file to execute

runner The executable used to invoke the script (by default cmd.exe for windows, sh for other platforms)

Value

A list holding the command and arguments

Examples

```
command_struct <- get_command('myfile.sh')
command <- command_struct$command
cli_args <- command_struct$args
```

`get_platform_value` *Returns the value based on the current platform.*

Description

Returns the value based on the current platform.

Usage

```
get_platform_value(unix_value, windows_value)
```

Arguments

`unix_value` The unix platform value
`windows_value` The windows platform value

Value

`unix_value` in case of unix system, else the `windows_value`

Examples

```
platform_value <- get_platform_value('.sh', '.bat')
```

`is_windows` *Returns true if windows, else false.*

Description

Returns true if windows, else false.

Usage

```
is_windows()
```

Value

True if windows, else false.

Examples

```
windows <- is_windows()
```

modify_script	<i>Modifies the provided script text and ensures the script content is executed in the correct location.</i>
---------------	--

Description

Modifies the provided script text and ensures the script content is executed in the correct location.

Usage

```
modify_script(script, args = c(), env = character(),  
             print_commands = FALSE)
```

Arguments

script	The script text
args	Optional script command line arguments
env	Optional character vector of name=value strings to set environment variables
print_commands	True if to print each command before invocation (not available for windows)

Value

The modified script text

Examples

```
script <- modify_script(script = 'echo test', args = c('first', 'second'), env = c('MYENV=MYENV'))
```

scriptexec	<i>scriptexec: Execute native scripts</i>
------------	---

Description

This package provides one main function: execute which executes the provided script and returns its output.

Index

`create_script_file`, 2

`execute`, 2

`generate_env_setup_script`, 3

`get_command`, 4

`get_platform_value`, 5

`is_windows`, 5

`modify_script`, 6

`scriptexec`, 6

`scriptexec-package (scriptexec)`, 6