

Package ‘locatexec’

July 22, 2025

Type Package

Title Detection and Localization of Executable Files

Version 0.1.1

Description A set of functions to locate some programs available on the user machine. The package provides functions to locate 'Node.js', 'npm', 'LibreOffice', 'Microsoft Word', 'Microsoft PowerPoint', 'Microsoft Excel', 'Python', 'pip', 'Mozilla Firefox' and 'Google Chrome'. User can test the availability of a program with eventually a version and call it with function `system2()` or `system()`. This allows the use of a single function to retrieve the path to a program regardless of the operating system and its configuration.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

Imports utils

BugReports <https://github.com/ardata-fr/locatexec/issues>

NeedsCompilation no

Author David Gohel [aut, cre],
ArData [cph]

Maintainer David Gohel <david.gohel@ardata.fr>

Repository CRAN

Date/Publication 2021-05-19 09:00:02 UTC

Contents

chrome_exec	2
excel_exec	3
exec_available	3
exec_locate	5
exec_version	6
firefox_exec	7

is_osx	8
is_unix	8
is_windows	9
libreoffice_exec	9
node_exec	10
npm_exec	10
pip_exec	11
powerpoint_exec	11
python_exec	12
word_exec	12
Index	13

chrome_exec	<i>Path to 'Google Chrome' executable</i>
-------------	---

Description

return the full path of 'Google Chrome' executable if found.

Usage

```
chrome_exec()
```

Value

chrome executable full path in a character vector of length 1.

See Also

Other executable full path: [excel_exec\(\)](#), [firefox_exec\(\)](#), [libreoffice_exec\(\)](#), [node_exec\(\)](#), [npm_exec\(\)](#), [pip_exec\(\)](#), [powerpoint_exec\(\)](#), [python_exec\(\)](#), [word_exec\(\)](#)

Examples

```
if(exec_available("chrome"))  
  message(chrome_exec())
```

excel_exec	<i>Path to 'Microsoft Excel' executable</i>
------------	---

Description

return the full path of 'Microsoft Excel' executable if found.

Usage

excel_exec()

Value

executable full path in a character vector of length 1.

See Also

Other executable full path: [chrome_exec\(\)](#), [firefox_exec\(\)](#), [libreoffice_exec\(\)](#), [node_exec\(\)](#), [npm_exec\(\)](#), [pip_exec\(\)](#), [powerpoint_exec\(\)](#), [python_exec\(\)](#), [word_exec\(\)](#)

Examples

```
if(exec_available("excel"))
  message(excel_exec())
```

exec_available	<i>Check executable availability and version</i>
----------------	--

Description

determine whether an executable is currently available on the user machine (optionally checking for a specific or greater version).

Usage

exec_available(exec, version = NULL, error = FALSE)

Arguments

- | | |
|------|---|
| exec | executable identifier, a single character. Use one of these values: <ul style="list-style-type: none">• chrome: 'Google Chrome' executable• firefox: 'Mozilla Firefox' executable• libreoffice: 'LibreOffice' executable• node: 'node.js' executable• npm: 'npm' executable |
|------|---|

	• python: 'python' executable • pip: 'pip' executable • excel: 'Microsoft Excel' executable • word: 'Microsoft Word' executable • powerpoint: 'Microsoft PowerPoint' executable
version	required version of executable
error	if TRUE, executes an error action if executable with the required version is not found.

Value

a logical indicating whether the executable is available.

Examples

```
if (exec_available("python")) {
  cat("python", as.character(exec_version("python")), "is available!\n")
}
if(exec_available("python", "3.9")) {
  cat("required version of python is available!\n")
}
if (exec_available("pip")) {
  cat("pip", as.character(exec_version("pip")), "is available!\n")
}
if (exec_available("pip", "19.3")) {
  cat("pip", as.character(exec_version("pip")), "is available!\n")
}
if (exec_available("chrome")) {
  cat("chrome", as.character(exec_version("chrome")), "is available!\n")
}
if (exec_available("firefox")) {
  cat("firefox", as.character(exec_version("firefox")), "is available!\n")
}
if (exec_available("node")) {
  cat("node.js", as.character(exec_version("node")), "is available!\n")
}
if (exec_available("npm")) {
  cat("npm", as.character(exec_version("npm")), "is available!\n")
}
if (exec_available("npm", version = "10.13.0")) {
  cat("npm", as.character(exec_version("npm")), "is available!\n")
}
if (exec_available("excel")) {
  cat("excel", as.character(exec_version("excel")), "is available!\n")
}
if (exec_available("word")) {
  cat("word", as.character(exec_version("word")), "is available!\n")
}
if (exec_available("powerpoint")) {
  cat("powerpoint", as.character(exec_version("powerpoint")), "is available!\n")
}
```

exec_locate

*Find an executable***Description**

Searches for an executable in a some places and use the highest version found (unless a specific version is requested).

The function mimic the behavior of the `rmarkdown::find_pandoc()` function in order to locate programs.

Some methods work differently depending on the OS or the program:

- Under Windows, the search for 'Google Chrome', 'Mozilla Firefox', 'Microsoft Word', 'Microsoft PowerPoint' and 'Microsoft Excel' is done in the registry, which means that only one version can be chosen, the one referenced in the registry. (you still can force another path with argument `dir`).
- 'Microsoft Word', 'Microsoft PowerPoint' and 'Microsoft Excel' can not be located on linux OS.
- the search for 'pip' is using the result of the search of 'python' to find the corresponding 'pip' executable.

Usage

```
exec_locate(exec, cache = TRUE, dir = NULL, version = NULL)
```

Arguments

exec	executable identifier, a single character. Use one of these values: • chrome: 'Google Chrome' executable • firefox: 'Mozilla Firefox' executable • libreoffice: 'LibreOffice' executable • node: 'node.js' executable • npm: 'npm' executable • python: 'python' executable • pip: 'pip' executable • excel: 'Microsoft Excel' executable • word: 'Microsoft Word' executable • powerpoint: 'Microsoft PowerPoint' executable
cache	if FALSE, search for the executable again even if the executable has been found previously.
dir	A character vector of directory paths under which the executable may be found.
version	The version of the executable to look for (e.g., "14.15.4"). If NULL (the default), it searches for the highest version.

Value

A list containing the path of the executable and its version if found. If not found, the version will be 0 and the exec_file will be NULL.

libreoffice

On some Ubuntu platforms, 'LibreOffice' require to add in the environment variable LD_LIBRARY_PATH the following path: /usr/lib/libreoffice/program (you should see the message "libreglo.so cannot open shared object file" if it is the case). This can be done with R command Sys.setenv(LD_LIBRARY_PATH = "/usr/lib/libreoffice/program/")

See Also

exec_available() will check if an executable is available and exec_version() will return the version of a located executable.

Examples

```
exec_locate("firefox")
exec_locate("chrome")
exec_locate("chrome", version = "88.0.4324.150")
exec_locate("libreoffice")
exec_locate("node")
exec_locate("npm")
exec_locate("python")
exec_locate("pip")
exec_locate("excel")
exec_locate("word")
exec_locate("powerpoint")
```

exec_version	Get executable version
--------------	------------------------

Description

get the version of an executable.

Usage

```
exec_version(exec)
```

Arguments

- | | |
|------|---|
| exec | executable identifier, a single character. Use one of these values: |
|------|---|
- chrome: 'Google Chrome' executable
 - firefox: 'Mozilla Firefox' executable
 - libreoffice: 'LibreOffice' executable

- node: 'node.js' executable
- npm: 'npm' executable
- python: 'python' executable
- pip: 'pip' executable
- excel: 'Microsoft Excel' executable
- word: 'Microsoft Word' executable
- powerpoint: 'Microsoft PowerPoint' executable

Value

a `numeric_version()` with the version of the executable found.

Examples

```
if(exec_available("chrome"))
    message(exec_version("chrome"))
if(exec_available("node"))
    message(exec_version("node"))
if(exec_available("python"))
    message(exec_version("python"))
if(exec_available("libreoffice"))
    message(exec_version("libreoffice"))
if(exec_available("word"))
    message(exec_version("word"))
if(exec_available("powerpoint"))
    message(exec_version("powerpoint"))
if(exec_available("firefox"))
    message(exec_version("firefox"))
```

firefox_exec

Path to 'Mozilla Firefox' executable

Description

return the full path of 'Mozilla Firefox' executable if found.

Usage

```
firefox_exec()
```

Value

Firefox executable full path in a character vector of length 1.

See Also

Other executable full path: `chrome_exec()`, `excel_exec()`, `libreoffice_exec()`, `node_exec()`, `npm_exec()`, `pip_exec()`, `powerpoint_exec()`, `python_exec()`, `word_exec()`

Examples

```
if(exec_available("firefox"))
    message(firefox_exec())
```

is_osx	<i>Test if the system is "macOS"</i>
--------	--------------------------------------

Description

Test if the operating system is "macOS"

Usage

```
is_osx()
```

Value

TRUE or FALSE

See Also

Other operating system testing: [is_unix\(\)](#), [is_windows\(\)](#)

Examples

```
is_osx()
```

is_unix	<i>Test if the system is "unix"</i>
---------	-------------------------------------

Description

Test if the operating system is "unix"

Usage

```
is_unix()
```

Value

TRUE or FALSE

See Also

Other operating system testing: [is_osx\(\)](#), [is_windows\(\)](#)

Examples

```
is_unix()
```

is_windows	<i>Test if the system is "Windows"</i>
------------	--

Description

Test if the operating system is "Windows"

Usage

```
is_windows()
```

Value

TRUE or FALSE

See Also

Other operating system testing: [is_osx\(\)](#), [is_unix\(\)](#)

Examples

```
is_windows()
```

libreoffice_exec	<i>Path to 'LibreOffice' executable</i>
------------------	---

Description

return the full path of 'LibreOffice' executable if found.

Usage

```
libreoffice_exec()
```

Value

executable full path in a character vector of length 1.

See Also

Other executable full path: [chrome_exec\(\)](#), [excel_exec\(\)](#), [firefox_exec\(\)](#), [node_exec\(\)](#), [npm_exec\(\)](#), [pip_exec\(\)](#), [powerpoint_exec\(\)](#), [python_exec\(\)](#), [word_exec\(\)](#)

Examples

```
if(exec_available("libreoffice")) {  
  message(libreoffice_exec())  
}
```

node_exec	<i>Path to 'node.js' executable</i>
-----------	-------------------------------------

Description

return the full path of 'node.js' executable if found.

Usage

```
node_exec()
```

Value

executable full path in a character vector of length 1.

See Also

Other executable full path: [chrome_exec\(\)](#), [excel_exec\(\)](#), [firefox_exec\(\)](#), [libreoffice_exec\(\)](#), [npm_exec\(\)](#), [pip_exec\(\)](#), [powerpoint_exec\(\)](#), [python_exec\(\)](#), [word_exec\(\)](#)

Examples

```
if(exec_available("node"))  
  message(node_exec())
```

npm_exec	<i>Path to 'npm' executable</i>
----------	---------------------------------

Description

return the full path of 'npm' executable if found.

Usage

```
npm_exec()
```

Value

executable full path in a character vector of length 1.

See Also

Other executable full path: [chrome_exec\(\)](#), [excel_exec\(\)](#), [firefox_exec\(\)](#), [libreoffice_exec\(\)](#), [node_exec\(\)](#), [pip_exec\(\)](#), [powerpoint_exec\(\)](#), [python_exec\(\)](#), [word_exec\(\)](#)

Examples

```
if(exec_available("npm"))  
  message(npm_exec())
```

pip_exec	<i>Path to 'pip' executable</i>
----------	---------------------------------

Description

return the full path of 'pip' executable if found.

Usage

```
pip_exec()
```

Value

executable full path in a character vector of length 1.

See Also

Other executable full path: [chrome_exec\(\)](#), [excel_exec\(\)](#), [firefox_exec\(\)](#), [libreoffice_exec\(\)](#), [node_exec\(\)](#), [npm_exec\(\)](#), [powerpoint_exec\(\)](#), [python_exec\(\)](#), [word_exec\(\)](#)

Examples

```
if(exec_available("pip"))  
  message(pip_exec())
```

powerpoint_exec	<i>Path to 'Microsoft PowerPoint' executable</i>
-----------------	--

Description

return the full path of 'Microsoft PowerPoint' executable if found.

Usage

```
powerpoint_exec()
```

Value

executable full path in a character vector of length 1.

See Also

Other executable full path: [chrome_exec\(\)](#), [excel_exec\(\)](#), [firefox_exec\(\)](#), [libreoffice_exec\(\)](#), [node_exec\(\)](#), [npm_exec\(\)](#), [pip_exec\(\)](#), [python_exec\(\)](#), [word_exec\(\)](#)

Examples

```
if(exec_available("powerpoint"))  
  message(powerpoint_exec())
```

python_exec	<i>Path to 'python' executable</i>
-------------	------------------------------------

Description

return the full path of 'python' executable if found.

Usage

```
python_exec()
```

Value

executable full path in a character vector of length 1.

See Also

Other executable full path: [chrome_exec\(\)](#), [excel_exec\(\)](#), [firefox_exec\(\)](#), [libreoffice_exec\(\)](#), [node_exec\(\)](#), [npm_exec\(\)](#), [pip_exec\(\)](#), [powerpoint_exec\(\)](#), [word_exec\(\)](#)

Examples

```
if(exec_available("python"))  
  message(python_exec())
```

word_exec	<i>Path to 'Microsoft Word' executable</i>
-----------	--

Description

return the full path of 'Microsoft Word' executable if found.

Usage

```
word_exec()
```

Value

executable full path in a character vector of length 1.

See Also

Other executable full path: [chrome_exec\(\)](#), [excel_exec\(\)](#), [firefox_exec\(\)](#), [libreoffice_exec\(\)](#), [node_exec\(\)](#), [npm_exec\(\)](#), [pip_exec\(\)](#), [powerpoint_exec\(\)](#), [python_exec\(\)](#)

Examples

```
if(exec_available("word"))  
  message(word_exec())
```

Index

* executable full path

- chrome_exec, [2](#)
- excel_exec, [3](#)
- firefox_exec, [7](#)
- libreoffice_exec, [9](#)
- node_exec, [10](#)
- npm_exec, [10](#)
- pip_exec, [11](#)
- powerpoint_exec, [11](#)
- python_exec, [12](#)
- word_exec, [12](#)

* operating system testing

- is_osx, [8](#)
- is_unix, [8](#)
- is_windows, [9](#)

chrome_exec, [2](#), [3](#), [7](#), [9–12](#)

excel_exec, [2](#), [3](#), [7](#), [9–12](#)

exec_available, [3](#)

exec_available(), [6](#)

exec_locate, [5](#)

exec_version, [6](#)

exec_version(), [6](#)

firefox_exec, [2](#), [3](#), [7](#), [9–12](#)

is_osx, [8](#), [8](#), [9](#)

is_unix, [8](#), [8](#), [9](#)

is_windows, [8](#), [9](#)

libreoffice_exec, [2](#), [3](#), [7](#), [9](#), [10–12](#)

node_exec, [2](#), [3](#), [7](#), [9](#), [10](#), [10](#), [11](#), [12](#)

npm_exec, [2](#), [3](#), [7](#), [9](#), [10](#), [10](#), [11](#), [12](#)

numeric_version(), [7](#)

pip_exec, [2](#), [3](#), [7](#), [9–11](#), [11](#), [12](#)

powerpoint_exec, [2](#), [3](#), [7](#), [9–11](#), [11](#), [12](#)

python_exec, [2](#), [3](#), [7](#), [9–12](#), [12](#)

word_exec, [2](#), [3](#), [7](#), [9–12](#), [12](#)