

# Package: fileaccess (via r-universe)

June 17, 2026

**Type** Package

**Title** File and Directory Access Validation

**Version** 0.1.1

**Description** Provides utility functions to validate read, write, execute, network, directory, and file access for local and Universal Naming Convention (UNC) network paths. Useful for pre-flight checks before file operations in data pipelines.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Imports** tools

**Suggests** spelling, testthat (>= 3.0.0)

**Language** en-US

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Vaishali JS [aut, cre]

**Maintainer** Vaishali JS <vaishali123.js@gmail.com>

**Repository** <https://cran.r-universe.dev>

**Date/Publication** 2026-06-17 19:33:31 UTC

**RemoteUrl** <https://github.com/cran/fileaccess>

**RemoteRef** HEAD

**RemoteSha** 992d6d578f4694f3977ae3d976bb05200f2691b1

## Contents

directory.access . . . . .	2
execute.access . . . . .	3
file.access . . . . .	4
fileaccess . . . . .	5
network.access . . . . .	5
read.access . . . . .	6
write.access . . . . .	7

**Index****9**

---

directory.access	<i>Check Whether Path is a Directory</i>
------------------	--

---

**Description**

Validates whether a directory exists and checks its read/write access along with content counts. Supports both local and UNC network paths.

**Usage**

```
directory.access(path)
```

**Arguments**

path                    Character string. Directory path to validate.

**Value**

A list of class `directory_access_result` containing:

- path - Normalized directory path
- exists - TRUE/FALSE
- readable - TRUE/FALSE
- writable - TRUE/FALSE
- file\_count - Number of files in directory
- folder\_count - Number of subdirectories
- empty - TRUE/FALSE
- status - Overall status message

**Examples**

```
# Check a temporary directory
result <- directory.access(tempdir())
print(result)

# Check a path that does not exist
result <- directory.access(
  file.path(tempdir(), "no_such_folder")
)
print(result)

# UNC network path (requires network access)
directory.access("//server/share/project")
```

---

execute.access	<i>Check Whether File is Executable</i>
----------------	---

---

**Description**

Validates whether a file or command can be executed. Supports both local paths and executables available in system PATH.

**Usage**

```
execute.access(path)
```

**Arguments**

path                    Character string. Executable name or full path.

**Value**

A list of class `execute_access_result` containing:

- path - Resolved executable path
- exists - TRUE/FALSE
- executable - TRUE/FALSE
- in\_path - TRUE/FALSE
- status - Overall status message

**Examples**

```
# Check if Rscript is available (available on all R machines)
result <- execute.access("Rscript")
print(result)
```

```
# Check a full path to an executable
result <- execute.access(file.path(R.home("bin"), "Rscript"))
print(result)
```

---

`file.access`*Check Whether Path is a File*

---

**Description**

Validates whether a file exists and returns detailed file information including permissions, size, type, and timestamps. Supports both local and UNC network paths.

**Usage**

```
file.access(path)
```

**Arguments**

`path`                    Character string. File path to validate.

**Value**

A list of class `file_access_result` containing:

- `path` - Normalized file path
- `exists` - TRUE/FALSE
- `can_read` - TRUE/FALSE
- `can_write` - TRUE/FALSE
- `can_execute` - TRUE/FALSE
- `locked` - TRUE/FALSE
- `size_mb` - File size formatted
- `type` - Detected file type description
- `modified` - Last modified timestamp
- `status` - Overall status message

**Examples**

```
# Create a temp file and check it
tmp <- tempfile(fileext = ".csv")
writeLines("col1,col2\n1,2", tmp)
result <- file.access(tmp)
print(result)
unlink(tmp)

# Check a file that does not exist
result <- file.access(
  file.path(tempdir(), "missing.csv")
)
print(result)
```

```
# UNC network path (requires network access)
file.access("//server/share/data.csv")
```

---

fileaccess	<i>fileaccess: File and Directory Access Validation</i>
------------	---

---

### Description

Provides utility functions to validate read, write, execute, network, directory, and file access for local and UNC network paths. Useful for pre-flight checks before file operations in data pipelines.

### Main functions

- [read.access](#) - Check if a path is readable
- [write.access](#) - Check if a path is writable
- [execute.access](#) - Check if a file/command is executable
- [network.access](#) - Check if a UNC network path is accessible
- [directory.access](#) - Check directory existence and permissions
- [file.access](#) - Check file existence, permissions and metadata

### Author(s)

**Maintainer:** Vaishali JS <vaishali123.js@gmail.com>

---

network.access	<i>Check Whether Path is a Network Path</i>
----------------	---

---

### Description

Validates whether a path is a network (UNC) path and whether it is accessible.

### Usage

```
network.access(path)
```

### Arguments

path	Character string. Network path to validate.
------	---

**Value**

A list of class `network_access_result` containing:

- `path` - Input path
- `network_path` - TRUE/FALSE
- `server` - Server name extracted from UNC path
- `share` - Share name extracted from UNC path
- `accessible` - TRUE/FALSE
- `status` - Overall status message

**Examples**

```
# A local path is not a network path
result <- network.access(tempdir())
print(result)

# Parse a UNC path structure (no network needed – just parsing)
result <- network.access("//myserver/myshare/data")
print(result)

# Real UNC network path (requires actual network access)
network.access("//server/share/data")
```

---

`read.access`

*Check Whether File or Directory is Readable*

---

**Description**

Validates whether a file or directory exists and can be accessed for reading. Supports both local and UNC network paths.

**Usage**

```
read.access(path)
```

**Arguments**

`path` Character string. Full path to file or directory.

**Value**

A list of class `read_access_result` containing:

- path - Normalized input path
- exists - TRUE/FALSE
- type - "File" or "Directory"
- readable - TRUE/FALSE
- status - Overall status message

**Examples**

```
# Check a temporary directory
result <- read.access(tempdir())
print(result)

# Check an existing temporary file
tmp <- tempfile()
writeLines("hello", tmp)
result <- read.access(tmp)
print(result)
unlink(tmp)

# Check a path that does not exist
result <- read.access(file.path(tempdir(), "no_such_file.csv"))
print(result)

# UNC network path (requires network access)
read.access("//server/share/data/file.csv")
```

---

write.access

*Check Whether File or Directory is Writable*

---

**Description**

Validates whether a file or directory can be accessed for writing. Supports both local and UNC network paths.

**Usage**

```
write.access(path)
```

**Arguments**

path                    Character string. Full path to file or directory.

**Value**

A list of class `write_access_result` containing:

- `path` - Normalized input path
- `exists` - TRUE/FALSE
- `type` - "File", "Directory", or "New File/Directory"
- `writable` - TRUE/FALSE
- `status` - Overall status message

**Examples**

```
# Check a temporary directory
result <- write.access(tempdir())
print(result)

# Check an existing temporary file
tmp <- tempfile()
writeLines("hello", tmp)
result <- write.access(tmp)
print(result)
unlink(tmp)

# Check a new file path that does not exist yet
result <- write.access(file.path(tempdir(), "newfile.csv"))
print(result)

# UNC network path (requires network access)
write.access("//server/share/output/results.csv")
```

# Index

`directory.access`, [2](#), [5](#)

`execute.access`, [3](#), [5](#)

`file.access`, [4](#), [5](#)

`fileaccess`, [5](#)

`fileaccess-package (fileaccess)`, [5](#)

`network.access`, [5](#), [5](#)

`read.access`, [5](#), [6](#)

`write.access`, [5](#), [7](#)