

Package: fluxnet (via r-universe)

October 17, 2026

Title Download, Read, and Visualize FLUXNET Data

Version 0.6.0.9000

Description Utility functions to help download, read in, and work with data from FLUXNET.

License MIT + file LICENSE

URL <https://github.com/EcosystemEcologyLab/fluxnet-package>,
<https://ecosystemecologylab.github.io/fluxnet-package/>

BugReports <https://github.com/EcosystemEcologyLab/fluxnet-package/issues>

Imports cli, dplyr, fs, ggplot2 (>= 4.0), lifecycle, lubridate, magrittr, processx, purrr, rappdirs, readr, reticulate (>= 1.41), rlang, stringr, tidyr

Suggests DBI, duckdb, glue, knitr, maps, quarto, rmarkdown, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/pak/sysreqs cmake make libicu-dev libpng-dev libuv1-dev python3 libx11-dev

Repository <https://cct-datascience.r-universe.dev>

Date/Publication 2026-06-26 21:40:04 UTC

RemoteUrl <https://github.com/EcosystemEcologyLab/fluxnet-package>

RemoteRef HEAD

RemoteSha 3a9758aab6a01cdb902890dcb4eb97cc57361bdc

Contents

flux_amf_credentials	2
flux_badm	3
flux_citations	4
flux_db_build	5
flux_db_connect	6
flux_db_update	7
flux_discover_files	8
flux_download	9
flux_extract	10
flux_install_shuttle	11
flux_listall	12
flux_map_sites	14
flux_qc	14
flux_read	16
flux_varinfo	17
Index	19

flux_amf_credentials	<i>Generate list of user info to pass to flux_download()</i>
----------------------	--

Description

A helper to generate a list of AmeriFlux user info to pass along to `flux_download()`. Because these elements are by default pulled from environment variables, it is recommended that you set them in a project-level `.Renviro`n file like `AMERIFLUX_USER_NAME=myusername`, etc.

Usage

```
flux_amf_credentials(  
  user_name = Sys.getenv("AMERIFLUX_USER_NAME", unset = NA_character_),  
  user_email = Sys.getenv("AMERIFLUX_USER_EMAIL", unset = NA_character_),  
  intended_use = Sys.getenv("AMERIFLUX_INTENDED_USE", unset = 6),  
  description = Sys.getenv("AMERIFLUX_DESCRIPTION", unset = NA_character_)  
)
```

Arguments

- | | |
|--------------|--|
| user_name | Your AmeriFlux username. |
| user_email | The email address associated with your AmeriFlux profile. |
| intended_use | An integer 1–6 as follows: <ul style="list-style-type: none">• 1 = Synthesis / network synthesis analysis• 2 = Land model/Earth system model• 3 = Remote sensing research• 4 = Other research |

- 5 = Education (Teacher or Student)
 - 6 = Other
- description An optional description of the project.

Value

A list with elements user_name, user_email, intended_use, and description.

flux_badm	<i>Read and tidy BADM subsets</i>
-----------	-----------------------------------

Description

Reads in BADM data from "BIF" csv files, subsets to a single VARIABLE_GROUP and returns a wide data frame.

Usage

```
flux_badm(
  manifest,
  variable_group,
  site_ids = NULL,
  networks = c("AMF", "CNF", "EUF", "FLX", "ICOS", "JPF", "KOF", "SAEON", "TERN")
)
```

Arguments

- manifest A manifest data frame produced by `flux_discover_files()`.
- variable_group A single VARIABLE_GROUP without a GRP_ prefix. Options can be viewed at <https://ameriflux.lbl.gov/data/badm/badm-standards/>
- site_ids A vector of site IDs to filter the manifest by. If NULL (the default), the manifest isn't filtered by site ID.
- networks A character vector indicating which networks to extract files from. Multiple values may be provided. Defaults to all networks.

Value

A tibble.

Examples

```
## Not run:
manifest <- flux_discover_files()
flux_badm(manifest, "SOIL_CHEM")
flux_badm(manifest, "LAI")

## End(Not run)
```

flux_citations

*Output required per-dataset citations for FLUXNET data***Description**

Given a vector of site IDs, this either returns a dataframe with or BibTeX citations for each site.

Usage

```
flux_citations(
  site_ids,
  output = c("data.frame", "bibtex"),
  bibtex_path = NULL,
  ...
)
```

Arguments

site_ids	Character vector of site IDs, e.g. <code>c("AR-Bal", "DE-Gwg")</code> .
output	Either <code>"data.frame"</code> to return a tibble or <code>"bibtex"</code> to return (or write) BibTeX entries. See ' Value ' for more details.
bibtex_path	Path to a .bib file to write BibTeX to, passed to the con argument of <code>writelines()</code> . If NULL (default), BibTeX will be returned as character. Has no effect if output = <code>'data.frame'</code> .
...	Additional arguments passed to <code>flux_listall()</code> .

Value

If output = `'data.frame'`, a tibble is returned with a bibentry list-column with elements of class `bibentry` and a citation column with citations formatted in the default style (see the `format()` method for `bibentry()` for more details). If output = `'bibtex'`, BibTeX entries are either returned as an atomic character vector (if bibtex_path is NULL) or written to a file.

Examples

```
# Return dataframe with bibentries and formatted citations
flux_citations(c("AR-Bal", "DE-Gwg"))

# Return BibTeX
flux_citations(c("AR-Bal", "DE-Gwg"), output = "bibtex")

# Append BibTeX entries to a file
## Not run:
flux_citations(
  c("AR-Bal", "DE-Gwg"),
  output = "bibtex",
  bibtex_path = "references.bib"
```

```
)

## End(Not run)
```

flux_db_build

Ingest FLUXNET data into a DuckDB database

Description

[Experimental] As an alternative to reading FLUXNET data into memory with `flux_read()`, this ingests all FLUXNET data into a DuckDB database that can be queried in R with dplyr via dbplyr. This is especially useful for daily and hourly data which are likely to overflow memory when trying to read them in with `flux_read()`.

Usage

```
flux_db_build(
  manifest,
  duckdb_path = Sys.getenv("FLUXNET_DB_PATH", unset = "fluxnet/fluxnet.duckdb")
)
```

Arguments

manifest	A file manifest tibble created by <code>flux_discover_files()</code> .
duckdb_path	File path to a .duckdb file used to store FLUXNET data. Can be set with the environment variable FLUXNET_DB_PATH.

Value

Returns nothing; called for side-effects only.

Note

Currently "BIF" and "BIFVARINFO" CSV files are *not* included in the DuckDB database.

See Also

`flux_db_update()`, `flux_db_connect()`

Examples

```
## Not run:

# Ingest all available data into a DuckDB database
manifest <- flux_discover_files()
flux_db_build(manifest)

## End(Not run)
```

flux_db_connect	<i>Connect to (or create) DuckDB database for FLUXNET data</i>
-----------------	--

Description

[Experimental] A convenience wrapper around [DBI::dbConnect\(\)](#).

Usage

```
flux_db_connect(  
  duckdb_path = Sys.getenv("FLUXNET_DB_PATH", unset = "fluxnet/fluxnet.duckdb")  
)
```

Arguments

duckdb_path File path to a .duckdb file used to store FLUXNET data. Can be set with the environment variable FLUXNET_DB_PATH.

Value

A database connection object of class duckdb_connection.

See Also

[flux_db_build\(\)](#)

Examples

```
## Not run:  
  
con <- flux_db_connect()  
DBI::dbListTables(con)  
# [1] "annual" "daily" "hourly" "manifest" "monthly" "weekly"  
  
# close connection  
DBI::dbDisconnect(con)  
  
## End(Not run)
```

flux_db_updateUpdate existing DuckDB database with new FLUXNET data

Description

[Experimental] Updates an existing DuckDB database created by `flux_db_build()`. When you've downloaded and extracted new data since building the database, this may be faster than rebuilding the database by deleting the `.duckdb` file and re-running `flux_db_build()`. If run successfully, it will update the 'manifest' table as well.

Usage

```
flux_db_update(  
  manifest,  
  duckdb_path = Sys.getenv("FLUXNET_DB_PATH", unset = "fluxnet/fluxnet.duckdb")  
)
```

Arguments

manifest	A file manifest tibble created by <code>flux_discover_files()</code> .
duckdb_path	File path to a <code>.duckdb</code> file used to store FLUXNET data. Can be set with the environment variable <code>FLUXNET_DB_PATH</code> .

Value

Returns nothing; called for side-effects only.

Note

This will error when new CSVs contain columns not already in the database. In this case, you'll have to delete the `.duckdb` file and run `flux_db_build()` again to regenerate the database.

See Also

`flux_db_build()`, `flux_db_connect()`

Examples

```
## Not run:  
  
# Build database  
manifest <- flux_discover_files()  
flux_db_build(manifest)  
  
# Download and extract additional sites  
flux_download()  
flux_extract()
```

```
# Update database
manifest <- flux_discover_files()
flux_db_update(manifest)
```

```
## End(Not run)
```

flux_discover_files *Create a "manifest" of downloaded and unzipped FLUXNET data files*

Description

Create a "manifest" of downloaded and unzipped FLUXNET data files

Usage

```
flux_discover_files(data_dir = "fluxnet/unzipped", ...)
```

Arguments

data_dir	The directory to look for FLUXNET CSV files in, typically the same as the output_dir used for flux_extract() .
...	Arguments passed to flux_listall() .

Value

Prints a summary of discovered available data and returns (invisibly) a dataframe with file paths and metadata extracted from file names and merged in from [flux_listall\(\)](#).

Examples

```
## Not run:
# Download data
flux_download(site_ids = c("AU-Boy", "BR-CST"))

# Extract annual and monthly data
flux_extract(resolutions = c("y", "m"))

# Create a manifest of extracted files
manifest <- flux_discover_files()

## End(Not run)
```

flux_download	<i>Download FLUXNET zip files</i>
---------------	-----------------------------------

Description

Downloads zip files (one per site) for available FLUXNET sites.

Usage

```
flux_download(
  file_list_df = NULL,
  site_ids = NULL,
  download_dir = "fluxnet",
  overwrite = FALSE,
  user_info = list(ameriflux = flux_amf_credentials()),
  ...
)
```

Arguments

file_list_df	When NULL (default), <code>flux_listall()</code> is used to determine the sites with data available to download (specific sites can be selected with <code>site_ids</code>). If <code>file_list_df</code> is supplied, it should be a data frame generated by <code>flux_listall()</code> , but potentially filtered to exclude some rows. This provides an alternative way of downloading only specific sites. See the examples for a possible use case. If <code>file_list_df</code> is not NULL, <code>cache_dir</code> , <code>use_cache</code> , and <code>cache_age</code> will be ignored but <code>site_ids</code> will still be used.
site_ids	If NULL (default) all available sites will be downloaded. Alternatively, supply a character vector of site IDs. For example, <code>c("UK-GaB", "CA-Ca2")</code> .
download_dir	The directory to download zip files to.
overwrite	Logical; overwrite already downloaded .zip files? If FALSE it will skip downloading existing files, unless they are invalid .zip files (e.g. due to partial download or corruption).
user_info	An optional list with data-hub-specific user information. Only AmeriFlux uses this currently. By default, these are retrieved from environment variables by <code>flux_amf_credentials()</code> .
...	Arguments passed to <code>flux_listall()</code> .

Value

Invisibly returns a tibble with the download URL, path on disk, HTTP status code, and whether or not the download was successful.

Examples

```
## Not run:
# Download data for all available sites
flux_download()

# Download data for just select site IDs
flux_download(site_ids = c("UK-GaB", "CA-Ca2"))

# Download specific sites filtered by site metadata (IGPB and data hub for example)
available_sites <- flux_listall()
to_get <-
  available_sites[available_sites$igbp == "CRO" & available_sites$data_hub == "AmeriFlux", ]
flux_download(file_list_df = to_get)

# Get a fresh list of available data and overwrite any existing downloads
flux_download(use_cache = FALSE, overwrite = FALSE)

## End(Not run)
```

flux_extract

Extract FLUXNET data from downloaded zip files

Description

Extracts data from zip files downloaded by `flux_download()` with options to extract only subsets of the files they contain.

Usage

```
flux_extract(
  zip_dir = "fluxnet",
  output_dir = fs::path(zip_dir, "unzipped"),
  site_ids = NULL,
  networks = c("AMF", "CNF", "EUF", "FLX", "ICOS", "JPF", "KOF", "SAEON", "TERN"),
  resolutions = c("y", "m", "w", "d", "h"),
  extract_varinfo = TRUE,
  extract_txt = FALSE,
  overwrite = FALSE
)
```

Arguments

zip_dir	The directory with the zip files
output_dir	The directory to unzip files to. Within this directory, data files will be nested by site.

site_ids	A character vector of site IDs (e.g. <code>c("AR-TF2", "CA-Ca2")</code>) can be supplied to only unzip data for certain sites. If <code>NULL</code> (default), all zip files found in <code>zip_dir</code> will be unzipped.
networks	A character vector indicating which networks to extract files from. Multiple values may be provided. Defaults to all networks.
resolutions	A character vector indicating which time resolutions to extract. Options are yearly ("y"), monthly ("m"), daily ("d"), and hourly/half-hourly ("h"). Multiple options may be passed with all of them as default.
extract_varinfo	Logical; extract the BIF and BIFVARINFO files containing variable information in the BADM Interchange Format? Defaults to <code>TRUE</code> .
extract_txt	Logical; extract the README.txt and DATA_POLICY_LICENSE_AND_INSTRUCTIONS.txt files? Defaults to <code>FALSE</code> .
overwrite	Logical; should existing extracted files be overwritten (<code>TRUE</code>) or ignored (<code>FALSE</code>)?

Value

Invisibly returns a dataframe of file paths and error messages if there are any errors generated by unzipping.

flux_install_shuttle *Install the fluxnet-shuttle CLI*

Description

Uses reticulate to install the **fluxnet-shuttle** Python library and command line interface into a virtual environment. Required for `flux_listall()` and `flux_download()`.

Usage

```
flux_install_shuttle(
  venv = Sys.getenv("FLUXNET_VENV", unset = "fluxnet"),
  shuttle_version = Sys.getenv("FLUXNET_SHUTTLE_VERSION", unset = "main"),
  from = c("github", "pypi"),
  reinitialize = FALSE
)
```

Arguments

venv	A name to use for creating a virtual environment. Defaults to "fluxnet", but we recommend using a project-specific virtual environment. You can set this in a project-level <code>.Renv</code> file as <code>FLUXNET_VENV=myproject</code> and it will be pulled from there.
shuttle_version	A version tag (e.g. "0.3.7") to install. Defaults to GitHub development version (for now). Can also be set as an environment variable in <code>.Renv</code> , E.g. <code>FLUXNET_SHUTTLE_VERSION=0.3.6</code>

from	Where to install from? Currently only "github" is available, but eventually "pypi" will be an option to install from PyPI .
reinitialize	Logical; if TRUE, the virtual environment specified in venv will be removed and re-initialized. Useful if you'd like to update the version of fluxnet-shuttle.

Value

The path to the fluxnet-shuttle CLI executable, silently.

Note

This will be run automatically the first time you run `flux_listall()` or `flux_download()`, so it is not necessary to run this function separately first. Big thanks to Andrew Heiss for helping me figure this all out!

Examples

```
## Not run:

# Standard install of development version
flux_install_shuttle()

# Specify a version
flux_install_shuttle(shuttle_version = "0.3.5")

# When run a second time, even after restarting the R session, it skips
# installation as long as the `venv` exists unless `reinitialize = TRUE`
flux_install_shuttle(shuttle_version = "0.3.6")

# If you want to update the version, set `reinitilaize = TRUE`
flux_install_shuttle(shuttle_version = "0.3.6", reinitialize = TRUE)

## End(Not run)
```

flux_listall

List available FLUXNET zip files for download

Description

This provides a wrapper around the **fluxnet-shuttle** command-line utility's `listall` command, which downloads a data frame of available .zip files. By default, the downloaded CSV is stored in `rappdirs::user_cache_dir("fluxnet")`. If there is already a FLUXNET shanpshot CSV file downloaded and it is more recent than `cache_age`, it will be read in instead of downloading a new snapshot.

Usage

```
flux_listall(
  cache_dir = rappdirs::user_cache_dir("fluxnet"),
  cache_age = as.difftime(1, units = "days"),
  clean_cache = 10L,
  log_file = NULL,
  echo_cmd = FALSE,
  ...
)
```

Arguments

cache_dir	The directory to store the list of available FLUXNET data in.
cache_age	A difftime object of length 1. If there are no cached snapshots more recent than cache_age, a new one will be downloaded and stored. You can force the cache to be invalidated with cache_age = -Inf.
clean_cache	A number of files ≥ 1 to keep in cache_dir. Defaults to 10, which keeps only the 10 most recent snapshots.
log_file	An optional file path (e.g. "log.txt") to direct the fluxnet-shuttle log to. Useful for debugging.
echo_cmd	Set to TRUE to print the shell command in the console. Passed to <code>processx::run()</code> .
...	Arguments passed to <code>flux_install_shuttle()</code> .

Value

A data frame of stations with available data and their metadata.

Note

To force the fluxnet R package to re-install the fluxnet-shuttle utility, remove the Python virtualenv it is installed in by running `reticulate::virtualenv_remove("fluxnet")`. Then, when you run `flux_listall()` next, the virtualenv will be re-created and fluxnet-shuttle will be re-installed.

Examples

```
## Not run:
fluxnet_files <- flux_listall()

# Invalidate cache and update it
fluxnet_files <- flux_listall(cache_age = -Inf)

## End(Not run)
```

flux_map_sites	<i>Map sites with downloaded data</i>
----------------	---------------------------------------

Description

Plot locations of sites in a file manifest on a world map.

Usage

```
flux_map_sites(
  manifest,
  color_var = c("data_hub", "igbp", "network", "first_year", "last_year")
)
```

Arguments

manifest	A data manifest created by <code>flux_discover_files()</code> .
color_var	A variable to use to color-code points.

Value

A ggplot2 object

Examples

```
## Not run:
manifest <- flux_discover_files()
flux_map_sites(manifest)

## End(Not run)
```

flux_qc	<i>Flag overly-gapfilled data</i>
---------	-----------------------------------

Description

Flags rows of data based on variables with associated _QC columns.

Usage

```
flux_qc(
  data,
  qc_vars,
  threshold = 0.5,
  operator = c("any", "all"),
  max_gapfilled = deprecated()
)
```

Arguments

data	A data frame created by <code>flux_read()</code> .
qc_vars	A character vector of column names with associated *_QC columns to use for flagging.
threshold	For aggregated data (weekly, daily, monthly, annual), a number between 0 and 1. For hourly and half-hourly data, either 0, 1, 2, or 3 corresponding to 0 = measured, 1 = good quality gap-fill (MDS), 2 = medium quality gap-fill, 3 = poor quality gap-fill. Rows with _QC values greater than the number provided will be flagged with qc_flagged = TRUE. Input can be length 1 or the same length as qc_vars to supply a different threshold for each variable.
operator	How to flag data when multiple qc_vars are supplied? If "any", the row will be marked as bad if <i>any</i> of the QC vars indicate gap-filling above their max_gapfill threshold. If "all" then the row will be flagged only if <i>all</i> of the QC vars are above their max_gapfill.
max_gapfilled	[Deprecated] Deprecated; use threshold instead.

Value

A tibble with the added columns p_gapfilled and qc_flagged. If operator = "any", qc_flagged = TRUE indicates that at least one of the supplied QC variables was more gapfilled than threshold and p_gapfilled will be the maximum proportion gapfilled across the QC vars for each row. If operator = "all", then qc_flagged = TRUE indicates that *all* of the supplied QC variables were more gapfilled than the thresholds supplies and p_gapfilled will be the minimum proportion gap-filled across all QC variables for each row.

Note

Most stations use Variable Ustar Threshold (VUT) method for NEE, but some use Constant Ustar Threshold (CUT), so when using NEE variables for QC flagging, it is a good idea to include both the VUT and CUT variant. E.g. qc_vars = c("NEE_VUT_REF", "NEE_CUT_REF").

Examples

```
## Not run:

# Flag rows where NEE_VUT_REF is more than 50% gapfilled
manifest <- flux_discover_files()
annual <- flux_read(manifest, resolution = "y")
annual_flagged <- flux_qc(
  annual,
  qc_vars = "NEE_VUT_REF",
  threshold = 0.5
)

# Use multiple variables each with a different threshold for QC
annual_flagged2 <- flux_qc(
  annual,
  qc_vars = c("NEE_VUT_REF", "TA_F"),
  threshold = c(0.4, 0.6)
```

```

)

# Same as above, but require *both* variables to be above their thresholds
# to consider that row a problem
annual_flagged2 <- flux_qc(
  annual,
  qc_vars = c("NEE_VUT_REF", "TA_F"),
  threshold = c(0.4, 0.6),
  operator = "all"
)

## End(Not run)

```

flux_read	<i>Read in FLUXNET data</i>
-----------	-----------------------------

Description

Reads and minimally cleans FLUXNET data found by [flux_discover_files\(\)](#).

Usage

```

flux_read(
  manifest,
  resolution = c("y", "m", "w", "d", "h"),
  datasets = c("ERA5", "FLUXMET"),
  networks = c("AMF", "CNF", "EUF", "FLX", "ICOS", "JPF", "KOF", "SAEON", "TERN"),
  site_ids = NULL
)

```

Arguments

manifest	A manifest data frame produced by flux_discover_files() .
resolution	The time resolution to read in. Must be one of "y" (annual), "m" (monthly), "w" (weekly), "d" (daily), or "h" (hourly/half-hourly).
datasets	Character vector of one or both of "FLUXMET" or "ERA5". Defaults to both.
networks	A character vector indicating which networks to extract files from. Multiple values may be provided. Defaults to all networks.
site_ids	A vector of site IDs to filter the manifest by. If NULL (the default), the manifest isn't filtered by site ID.

Examples

```
## Not run:
manifest <- flux_discover_files()
daily <- flux_read(manifest, resolution = "d")
annual <- flux_read(manifest, resolution = "y")

# Filter manifest by metadata first
metadata <- flux_listall()

library(dplyr)
manifest_enriched <- left_join(manifest, metadata, by = join_by(site_id))
manifest_WET <- manifest_enriched %>% filter(igbp == "WET")
annual_wet <- flux_read(manifest_WET, resolution = "y")

## End(Not run)
```

flux_varinfo

Read variable info from "BIFVARINFO" files

Description

Extracts and tidies variable information from "BIFVARINFO" csv files.

Usage

```
flux_varinfo(
  manifest,
  resolution = c("y", "m", "w", "d", "h"),
  networks = c("AMF", "CNF", "EUF", "FLX", "ICOS", "JPF", "KOF", "SAEON", "TERN"),
  site_ids = NULL
)
```

Arguments

manifest	A manifest data frame produced by flux_discover_files() .
resolution	The time resolution to read in. Must be one of "y" (annual), "m" (monthly), "w" (weekly), "d" (daily), or "h" (hourly/half-hourly).
networks	A character vector indicating which networks to extract files from. Multiple values may be provided. Defaults to all networks.
site_ids	A vector of site IDs to filter the manifest by. If NULL (the default), the manifest isn't filtered by site ID.

Value

A tibble

Note

This only returns variable info (VARIABLE_GROUP == GRP_VAR_INFO) from the "BIFVARINFO" files as much of the other metadata they contain can be found in the results of `flux_listall()`.

HEIGHT is returned as a character value because some heights are reported as ranges and cannot be parsed as a single numeric value.

Examples

```
## Not run:  
manifest <- flux_discover_files()  
flux_varinfo(manifest)  
  
## End(Not run)
```

Index

bibentry, [4](#)
bibentry(), [4](#)

DBI::dbConnect(), [6](#)

flux_amf_credentials, [2](#)
flux_amf_credentials(), [9](#)
flux_badm, [3](#)
flux_citations, [4](#)
flux_db_build, [5](#)
flux_db_build(), [6](#), [7](#)
flux_db_connect, [6](#)
flux_db_connect(), [5](#), [7](#)
flux_db_update, [7](#)
flux_db_update(), [5](#)
flux_discover_files, [8](#)
flux_discover_files(), [3](#), [5](#), [7](#), [14](#), [16](#), [17](#)
flux_download, [9](#)
flux_download(), [2](#), [10–12](#)
flux_extract, [10](#)
flux_extract(), [8](#)
flux_install_shuttle, [11](#)
flux_install_shuttle(), [13](#)
flux_listall, [12](#)
flux_listall(), [4](#), [8](#), [9](#), [11](#), [12](#)
flux_map_sites, [14](#)
flux_qc, [14](#)
flux_read, [16](#)
flux_read(), [5](#), [15](#)
flux_varinfo, [17](#)

processx::run(), [13](#)

writeLines(), [4](#)