

Package ‘daisytools’

October 4, 2026

Title Reading and Visualizing Output from 'Daisy' Simulations

Version 1.0.1

Description Read, transform, and visualize log files produced by 'Daisy' simulations. Supports importing dlf output files, reshaping depth-dependent and time-series data, and creating static and interactive plots for exploratory analysis and comparison of simulation results.

License GPL (>= 3)

Depends R (>= 4.1.0)

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

Imports cowplot, ggplot2, methods, rlang, stats, utils, plotly, data.table, tidyr, tidyselect

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL <https://daisy-model.github.io/daisy-r-tools/>, <https://daisy.ku.dk/>

NeedsCompilation no

Author Silas Ørting [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-3081-1547>>)

Maintainer Silas Ørting <silas@di.ku.dk>

Repository CRAN

Date/Publication 2026-10-04 08:40:02 UTC

Contents

animate_dlf	2
daisy_time_to_timestamp	3
depth_wide_to_long	4

dir_names_to_columns	5
Dlf-class	6
drop_dir_from_names	6
filter_dlfs	7
mass_balance	7
mass_balance_summary	8
merge_dlfs	9
plot_dlf	10
plot_dlf_depth	11
plot_mass_balance	12
read_dlf	13
read_dlf_dir	14
read_dlf_file	15
read_dlf_spawn	16
strip_common_prefix_from_names	16
subset_dlf	17
write_dlf	18
[,Dlf-method	18
[<-,Dlf-method	19
[[,Dlf-method	20
[[<-,Dlf-method	20
\$.Dlf-method	21
\$<-,Dlf-method	21

Index 22

animate_dlf	<i>Animate a depth dependent time series from a Dlf file</i>
-------------	--

Description

Animate a depth dependent time series from a Dlf file

Usage

```
animate_dlf(dlf, x_var = NULL, y_var = "z", time_var = "time")
```

Arguments

dlf	An S4 object of class Dlf
x_var	Name of variable on x axis. If NULL assume it is the column that is not y_var or time_var
y_var	Name of variable on y axis
time_var	Name of time variable

Value

A plotly figure

Examples

```
data_dir <- system.file("extdata", package="daisytools")
path <- file.path(data_dir, "daily/DailyP/DailyP-Daily-WaterFlux.dlf")
dlf <- read_dlf(path)
animate_dlf(dlf, "q")
```

daisy_time_to_timestamp

Transform daisy time columns to a single timestamp

Description

Transform daisy time columns to a single timestamp

Usage

```
daisy_time_to_timestamp(
  dlf,
  time_col_name = "time",
  year_col = NULL,
  month_col = NULL,
  day_col = NULL,
  hour_col = NULL,
  drop_daisy_time_cols = FALSE
)
```

Arguments

dlf	An S4 object of class Dlf or a list of Dlf objects
time_col_name	The name of the new timestamp column
year_col	Name of year column, set to NULL if no year column
month_col	Name of month column, set to NULL if no month column
day_col	Name of day column, set to NULL if no day column
hour_col	Name of hour column, set to NULL if no hour column
drop_daisy_time_cols	If TRUE drop the year, month, day, hour columns

Value

An S4 object of class Dlf

If all of year_col, month_col, day_col and hour_col are NULL, then an attempt is made to guess the relevant names.

The returned object is identical to the dlf, except that the Columns 'year', 'month', 'mday' and 'hour' are dropped and replaced with a column containing the corresponding POSIXct timestamp

Examples

```
data_dir <- system.file("extdata", package="daisytools")
path <- file.path(data_dir, 'daily/DailyP/DailyP-Daily-WaterFlux.dlf')
dlf <- read_dlf(path)
dlf@data[1,]
dlf <- daisy_time_to_timestamp(dlf)
dlf@data[1,]
```

depth_wide_to_long	<i>Transform wide format time series to a long format time series It is assumed that dlf@data only contains time_name columns and depth columns with the format <var_name>_<depth-below-surface> e.g. "q_100"</i>
--------------------	---

Description

Transform wide format time series to a long format time series It is assumed that dlf@data only contains time_name columns and depth columns with the format <var_name>_<depth-below-surface> e.g. "q_100"

Usage

```
depth_wide_to_long(
  dlf,
  var_name,
  time_name = "time",
  depth_name = "z",
  depth_unit = "unknown"
)
```

Arguments

dlf	An S4 object of class Dlf or a list of Dlf objects
var_name	Name of variable in columns
time_name	Name of time column
depth_name	Name to use for new depth column
depth_unit	Unit of depth

Value

An S4 object of class Dlf

The data of the returned Dlf object contains one row for each time/depth combination.

Examples

```
data_dir <- system.file("extdata", package="daisytools")
path <- file.path(data_dir, "daily/DailyP/DailyP-Daily-WaterFlux.dlf")
dlf <- read_dlf(path, convert_depth=FALSE)
dlf@data[1,]
dlf <- depth_wide_to_long(dlf, "q")
head(dlf@data)
```

dir_names_to_columns	<i>Remove directory part of dlf name and store it in the data slot</i>
----------------------	--

Description

Remove directory part of dlf name and store it in the data slot

Usage

```
dir_names_to_columns(dlfs, column_name_prefix = NULL)
```

Arguments

dlfs	List of dlfs
column_name_prefix	Prefix to use for storing directory structure

Value

A list of S4 objects of class Dlf. Each dlf is named with the basename of its original name

Examples

```
data_dir <- system.file("extdata", package="daisytools")
dlfs <- read_dlf_dir(file.path(data_dir, "annual"))
names(dlfs)
dlfs <- dir_names_to_columns(dlfs)
lapply(dlfs, function(dlf) {
  dlf@data[1, "dir"]
})
```

Dlf-class	<i>Class encapsulating Daisy log file (dlf) data</i>
-----------	--

Description

Class encapsulating Daisy log file (dlf) data

Slots

header A list with named components containing meta information

units A data.frame with the units of the values in data

data A data.frame with the values from the dlf file

drop_dir_from_names	<i>Remove directory part from dlf names</i>
---------------------	---

Description

Remove directory part from dlf names

Usage

```
drop_dir_from_names(dlfs)
```

Arguments

dlfs List of Dlf objects

Value

List of S4 objects of class Dlf with dir part stripped from names

filter_dlfs	<i>Filter a list of Dlfs</i>
-------------	------------------------------

Description

Filter a list of Dlfs

Usage

```
filter_dlfs(dlfs, FUN)
```

Arguments

dlfs	List of Dlfs
FUN	Function mapping dlf to a boolean of length 1

Value

A list of Dlfs where FUN is TRUE

Examples

```
data_dir <- system.file("extdata", package="daisytools")
dlfs <- read_dlf_dir(file.path(data_dir, "annual"))
names(dlfs)
## We only want the dlfs that have a "Crop" column
dlfs <- filter_dlfs(dlfs, function(dlf) {
  "Crop" %in% colnames(dlf@data)
})
names(dlfs)
```

mass_balance	<i>Calculate mass balance</i>
--------------	-------------------------------

Description

Calculate mass balance

Usage

```
mass_balance(
  dlfs,
  input,
  output,
  content,
  use_initial_content_as_reference = TRUE
)
```

Arguments

dlfs	Either a list of Dlf or a single Dlf
input	Name(s) of variable(s) containing mass input
output	Name(s) of variable(s) containing mass output
content	Name(s) of variable(s) containing mass content
use_initial_content_as_reference	If TRUE subtract the initial content from the content sum before calculating balance.

Value

A list of Dlf or a single Dlf. Four variables are added to each Dlf, input_sum, output_sum, content_sum, and balance, which hold the sum and balance of the input/output/content variables calculated for each time point.

Examples

```
data_dir <- system.file("extdata", package="daisytools")
path <- file.path(data_dir, "hourly/P2D-Daily-Soil_Chemical_110cm.dlf")
dlf <- read_dlf(path)
input <- c("In_Matrix", "In_Biopores", "External", "Transform", "Tillage")
output <- c("Decompose", "Leak_Matrix", "Leak_Biopores", "Drain_Soil",
           "Drain_Biopores", "Uptake")
content <- c("Content", "Biopores")
dlf <- mass_balance(dlf, input, output, content)
```

mass_balance_summary *Mass balance summary for a set of specified variables*

Description

Mass balance summary for a set of specified variables

Usage

```
mass_balance_summary(dlfs, input, output, content)
```

Arguments

dlfs	Either a list of Dlf or a single Dlf
input	Name(s) of variable(s) containing mass input
output	Name(s) of variable(s) containing mass output
content	Name(s) of variable(s) containing mass content

Value

Either a single list or a list of lists, with the nested lists having five elements: Inputs, Outputs, InitialContent, FinalContent, and Balance.

Examples

```
data_dir <- system.file("extdata", package="daisytools")
path <- file.path(data_dir, "hourly/P2D-Daily-Soil_Chemical_110cm.dlf")
dlf <- read_dlf(path)
input <- c("In_Matrix", "In_Biopores", "External", "Transform", "Tillage")
output <- c("Decompose", "Leak_Matrix", "Leak_Biopores", "Drain_Soil",
           "Drain_Biopores", "Uptake")
content <- c("Content", "Biopores")
mass_balance_summary(dlf, input, output, content)
```

merge_dlfs

*Merge a list of Dlfs***Description**

Merge a list of Dlfs

Usage

```
merge_dlfs(dlfs, dlf_name_column = "name")
```

Arguments

dlfs List of Dlfs

dlf_name_column

Name of column to store name of dlf each row came from

Value

A Dlf object with the data of the dlfs merged into a single dlf.

Examples

```
data_dir <- system.file("extdata", package="daisytools")
dlfs <- read_dlf_dir(file.path(data_dir, "annual"))
dlfs <- dlfs[startsWith(names(dlfs), "Annual-FN")]
dlfs <- dir_names_to_columns(dlfs)
dlf <- merge_dlfs(dlfs)
head(dlf@data)
```

plot_dlf	<i>Plot one or more variables from one or more Dlfs. Each variable is plotted in a separate subplot.</i>
----------	--

Description

Plot one or more variables from one or more Dlfs. Each variable is plotted in a separate subplot.

Usage

```
plot_dlf(
  dlfs,
  x_var,
  y_vars,
  type = "points",
  mode = "auto",
  title_suffix = "",
  group_col = "sim",
  x_label = NULL,
  legend_label = NULL
)
```

Arguments

dlfs	Either a list of Dlf or a single Dlf. If a list with of components, the names are used for the legend.
x_var	Name of variable for x axis
y_vars	Either a list of variables to plot, or a single variable
type	Either a string defining the type of plot to produce OR a single parameter function returning a ggplot2 object. If a string, the the following types are supported * "bar" * "points" * "lines" Types cannot be abbreviated, but they can be combined as in type="barpoints" The types can be separated by a non-type string for readability as in type="bar + lines + points" If a function, then it is passed a ggplot2 object and it should add a geom to the object and return it. For example, type=function(gg) { gg + geom_point() }
mode	One of 'auto', 'single', 'grouped', 'list'. Default is 'auto'. If 'auto' try to guess the mode. If 'single' assume dlfs is a single dlf without a grouping variable If 'grouped' assume dlfs is a single dlf with a grouping variable. The grouping variable is the value of group_col If 'list' assume dlfs is a named list of dlfs without grouping variables. The names are used for grouping.
title_suffix	A string that is appended to the title of all subplots
group_col	Name of column to use for grouping. Ignored if mode is not 'grouped' or 'auto'##
x_label	Label to use for x axis
legend_label	Label to use for legend

Value

A ggplot2 object. If y_vars is a single variable then the plot can be themed and further data added. If y_vars is a list of more than one variable, it is not possible to add further data to it.

Examples

```
data_dir <- system.file("extdata", package="daisytools")
path <- file.path(data_dir, "annual/Annual-FN")
dlfs <- read_dlf(path)

# Plot using a string for type
plot_dlf(dlfs[[1]], "year", "Crop", "bar")

# Same plot with a function for type
geom <- function(gg) { gg + ggplot2::geom_col(position="dodge") }
plot_dlf(dlfs[[1]], "year", "Crop", geom)

# A plot with four variables
dlfs <- strip_common_prefix_from_names(dlfs)
vars <- c("Matrix_Leaching", "Crop_Uptake", "Soil_Drain", "Surface_Loss")
plot_dlf(dlfs, "year", vars, "bar", title_suffix=" - Annual Field Nitrogen")
```

plot_dlf_depth	<i>Plot one or more timepoints from one or more dlfs containing depth dependent time series</i>
----------------	---

Description

Plot one or more timepoints from one or more dlfs containing depth dependent time series

Usage

```
plot_dlf_depth(
  dlfs,
  x_var = NULL,
  time_points = NULL,
  y_var = "z",
  time_var = "time",
  x_label = "",
  y_label = "",
  legend_name = "dlf",
  title = "",
  num_samples = 4
)
```

Arguments

dlfs	An S4 object of class Dlf or a list of Dlf objects
x_var	Name of variable on x axis. If NULL assume it is the column that is not y_var or time_var
time_points	List or vector of time points to plot. All dlfs must have data for all time points. If NULL then sample random time points.
y_var	Name of variable on y axis
time_var	Name of time variable
x_label	Label for x axis (unit will be appended)
y_label	Label for y axis (unit will be appended)
legend_name	Name to use for legend
title	Plot title (time point will be appended)
num_samples	How many time points to sample when time_points is NULL

Value

ggplot2 object

plot_mass_balance	<i>Plot Dlf object with mass balance calculation in a control chart</i>
-------------------	---

Description

Plot Dlf object with mass balance calculation in a control chart

Usage

```
plot_mass_balance(dlfs, x_var, title_suffix = "")
```

Arguments

dlfs	Either a list of Dlf or a single Dlf. If a list of Dlfs each Dlf is plotted in a separate subplot.
x_var	Name of variable for x axis
title_suffix	A string that is appended to the title of all subplots

Value

A ggplot2 object.

Examples

```
data_dir <- system.file("extdata", package="daisytools")
path <- file.path(data_dir, "hourly/P2D-Daily-Soil_Chemical_110cm.dlf")
dlf <- read_dlf(path)
input <- c("In_Matrix", "In_Biopores", "External", "Transform", "Tillage")
output <- c("Decompose", "Leak_Matrix", "Leak_Biopores", "Drain_Soil",
           "Drain_Biopores", "Uptake")
content <- c("Content", "Biopores")
mb <- mass_balance(dlf, input, output, content)
mb <- daisy_time_to_timestamp(mb)
plot_mass_balance(mb, "time", " - Soil chemical @ 110cm")
```

read_dlf	<i>Read a dlf file (Daisy log file) or a directory of dlf files</i>
----------	---

Description

Read a dlf file (Daisy log file) or a directory of dlf files

Usage

```
read_dlf(
  path,
  mode = "auto",
  pattern = ".*\\.dlf",
  col_name = "sim",
  convert_time = TRUE,
  convert_depth = TRUE
)
```

Arguments

path	Path to dlf file or directory containing dlf files.
mode	One of 'auto', 'file', 'spawn', 'dir'. Default is 'auto'. If 'auto' try to guess the mode. If 'file' assume path is a single dlf file. If 'spawn' assume path is a directory containing log directories generated by the daisy spawn program. If 'dir' assume path is a directory tree containing dlf files
pattern	Regex pattern of files to include. Only relevant for modes 'auto', 'spawn', and 'dir'
col_name	Name of column to store directory name in. Only relevant for modes 'auto' and 'spawn'.
convert_time	If TRUE convert daisy time to a timestamp stored under column 'time' in the data slot of each dlf
convert_depth	If TRUE and the file contains depth data, convert it to a nicer depth format with three columns (depth, time, variable). It implies convert_time=TRUE

Value

For mode 'file': An S4 object of class Dlf with three slots: header, units, data. header is list with everything in the dlf file before the data. key: value pairs from the header are stored as named components of the list. units is a data.frame containing the units of the values in data. data is a data.frame containing the logged values

For mode 'spawn': A named list of Dlf objects, where names are names of log types For mode 'dir': A named list of Dlf objects, where names are paths to each file.

See Also

read_dlf_file, read_dlf_dir, read_dlf_spawn

Examples

```
data_dir <- system.file("extdata", package="daisytools")
path <- file.path(data_dir, "annual/Annual-FN/HourlyP-Annual-FN-2-2b.dlf")
dlf <- read_dlf(path)
slotNames(dlf)
names(dlf@header)
dlf@units$Crop
dlf$Crop # equivalent to dlf@data$Crop
dlf[["Crop"]]
```

read_dlf_dir

Walk a directory tree and read all dlf files in it.

Description

Walk a directory tree and read all dlf files in it.

Usage

```
read_dlf_dir(directory, pattern = ".*\\.dlf")
```

Arguments

directory	Path to directory
pattern	Regex pattern of files to include

Value

A list of S4 objects of class Dlf. Each dlf is named with the relative path to it

Examples

```
data_dir <- system.file("extdata", package="daisytools")
dlfs <- read_dlf_dir(file.path(data_dir, "annual"))
print(names(dlfs))

dlfs <- drop_dir_from_names(dlfs)
print(names(dlfs))

dlfs <- strip_common_prefix_from_names(dlfs)
print(names(dlfs))
```

read_dlf_file	<i>Read a dlf file (Daisy log file)</i>
---------------	---

Description

Read a dlf file (Daisy log file)

Usage

```
read_dlf_file(path)
```

Arguments

path	Path to dlf file
------	------------------

Value

An S4 object of class Dlf with three slots: header, units, data

header is list with everything in the dlf file before the data. key: value pairs from the header are stored as named components of the list. units is a data.frame containing the units of the values in data. data is a data.frame containing the logged values

Examples

```
data_dir <- system.file("extdata", package="daisytools")
path <- file.path(data_dir, "annual/Annual-FN/HourlyP-Annual-FN-2-2b.dlf")
dlf <- read_dlf_file(path)
slotNames(dlf)
names(dlf@header)
dlf@units$Crop
dlf$Crop # equivalent to dlf@data$Crop
dlf[["Crop"]]
```

read_dlf_spawn	<i>Walk a directory tree generated by daisy spawn, read all dlf files in it, and merge similar log files into one dlf.</i>
----------------	--

Description

Walk a directory tree generated by daisy spawn, read all dlf files in it, and merge similar log files into one dlf.

Usage

```
read_dlf_spawn(directory, pattern = ".*\\.dlf", col_name = "sim")
```

Arguments

directory	Path to directory
pattern	Regex pattern of files to include
col_name	Name of column to store directory name in

Value

A list of S4 objects of class Dlf. Each dlf is named with the log type and the directory is stored in the column col_name

Examples

```
data_dir <- file.path(system.file("extdata", package="daisytools",
                                   "daisy-spawn-like"))
print(list.files(data_dir, recursive=TRUE))
dlfs <- read_dlf_spawn(data_dir)
dlf_names <- names(dlfs)
print(dlf_names)

print(unique(dlfs[[dlf_names[1]]]$sim))
```

strip_common_prefix_from_names	<i>Strip common prefix from dlf names</i>
--------------------------------	---

Description

Strip common prefix from dlf names

Usage

```
strip_common_prefix_from_names(dlfs)
```

Arguments

dlfs List of Dlf objects

Value

List of Dlf objects with common prefix stripped from names

subset_dlf	<i>Select the subset of measurements in a specific time range.</i>
------------	--

Description

Select the subset of measurements in a specific time range.

Usage

```
subset_dlf(  
  dlfs,  
  date_from = "",  
  date_to = "",  
  time_from = "00:00:00",  
  time_to = "23:59:59",  
  time_zone = "UTC",  
  time_col = "time"  
)
```

Arguments

dlfs	Either a list of Dlf or a single Dlf
date_from	Start date (included). ISO 8601 format (yyyy-mm-dd), e.g 1999-01-31. Can be the empty string "" in which case only dates after date_to are excluded
date_to	End date (included). ISO 8601 format (yyyy-mm-dd), e.g 1999-01-31. Can be the empty string "" in which case only dates before date_from are excluded
time_from	Start time (included). ISO 8601 format (hh:mm:ss), e.g. 07:54:20
time_to	End time (included). ISO 8601 format (hh:mm:ss), e.g. 07:54:20
time_zone	Name of time zone.
time_col	Name of time column in dlfs. Must be in POSIXct format. See daisy_time_to_timestamp for converting daisy time columns to POSIXct

Value

Dlfs with data slot containing the selected rows

Examples

```
# Load dlf data with read_dlf
data_dir <- file.path(system.file("extdata", package="daisytools"),
                        "annual/Annual-FN")
dlfs <- read_dlf_dir(data_dir)
dlfs <- daisy_time_to_timestamp(dlfs)
data.frame(lapply(dlfs, function(dlf) { dlf$time}))
dlfs <- subset_dlf(dlfs, "1995-04-01", "1999-04-01", time_zone="CEST")
data.frame(lapply(dlfs, function(dlf) { dlf$time}))
```

write_dlf	Write a dlf object to a file
-----------	------------------------------

Description

Write a dlf object to a file

Usage

```
write_dlf(dlf, path, sep = "\t", include_dlf_header = FALSE)
```

Arguments

dlf	Dlf object to write
path	Output path to write to
sep	Separator to use in csv file
include_dlf_header	If TRUE write the Dlf header to the beginning of the csv file

Value

Invisibly returns NULL. Called for its side effect of writing the Dlf contents to path.

[,Dlf-method	[indexing on the data part of the dlf
--------------	--

Description

[indexing on the data part of the dlf

Usage

```
## S4 method for signature 'Dlf'
x[i, j]
```

Arguments

x	The dlf object
i	Name or index of row OR if j is missing, name or index of column
j	Name or index of column

Value

If j is missing, a `data.frame` containing the selected rows from the data slot. Otherwise, the result of subsetting the data slot by rows and columns, typically a vector for a single selected column or a `data.frame` for multiple columns.

[<-,Dlf-method	<i>[assignment on the data part of the dlf</i>
----------------	---

Description

[assignment on the data part of the dlf

Usage

```
## S4 replacement method for signature 'Dlf'
x[i, j] <- value
```

Arguments

x	The dlf object
i	Name or index of row OR if j is missing, name or index of column
j	Name or index of column
value	New value to assign

Value

An updated Dlf object with the selected rows, columns, or cells replaced in the data slot.

[[,Dlf-method	<i>[[indexing on the data part of the dlf</i>
---------------	--

Description

[[indexing on the data part of the dlf

Usage

```
## S4 method for signature 'Dlf'
x[[i, j]]
```

Arguments

x	The dlf object
i	Name or index of row OR if j is missing, name or index of column
j	Name or index of column

Value

If j is missing, the selected column from the data slot as the underlying column vector. Otherwise, the single value stored at row i and column j.

[[<-,Dlf-method	<i>[[assignment on the data part of the dlf</i>
-----------------	--

Description

[[assignment on the data part of the dlf

Usage

```
## S4 replacement method for signature 'Dlf'
x[[i, j]] <- value
```

Arguments

x	The dlf object
i	Name or index of row OR if j is missing, name or index of column
j	Name or index of column
value	New value to assign

Value

An updated Dlf object with the selected column or single cell replaced in the data slot.

\$.Dlf-method	<i>\$ indexing on the data part of the dlf</i>
---------------	--

Description

\$ indexing on the data part of the dlf

Usage

```
## S4 method for signature 'Dlf'
x$name
```

Arguments

x	The dlf object
name	Name of column in data part

Value

The selected column from the data slot as the underlying column vector.

\$.<-,Dlf-method	<i>\$ assignment on the data part of the dlf</i>
------------------	--

Description

\$ assignment on the data part of the dlf

Usage

```
## S4 replacement method for signature 'Dlf'
x$name <- value
```

Arguments

x	The dlf object
name	Name of column in data part
value	New value to assign

Value

An updated Dlf object with the selected column replaced in the data slot.

Index

`[,Dlf-method`, [18](#)
`[<-,Dlf-method`, [19](#)
`[[,Dlf-method`, [20](#)
`[[<-,Dlf-method`, [20](#)
`$,Dlf-method`, [21](#)
`$<-,Dlf-method`, [21](#)

`animate_dlf`, [2](#)

`daisy_time_to_timestamp`, [3](#)
`depth_wide_to_long`, [4](#)
`dir_names_to_columns`, [5](#)
`Dlf (Dlf-class)`, [6](#)
`Dlf-class`, [6](#)
`drop_dir_from_names`, [6](#)

`filter_dlfs`, [7](#)

`mass_balance`, [7](#)
`mass_balance_summary`, [8](#)
`merge_dlfs`, [9](#)

`plot_dlf`, [10](#)
`plot_dlf_depth`, [11](#)
`plot_mass_balance`, [12](#)

`read_dlf`, [13](#)
`read_dlf_dir`, [14](#)
`read_dlf_file`, [15](#)
`read_dlf_spawn`, [16](#)

`strip_common_prefix_from_names`, [16](#)
`subset_dlf`, [17](#)

`write_dlf`, [18](#)